Intraoperative Localization and Scene Reconstruction using Differentiable Rendering and Graph-based Landmark Registration with Application to Cystoscopy

Von der Fakultät

Konstruktions-, Produktions- und Fahrzeugtechnik

der Universität Stuttgart

zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte Abhandlung

vorgelegt von

Johannes Matthias Schüle

geboren in Göppingen

Hauptberichter:	Prof. Dr.-Ing. Dr. h. c. Oliver Sawodny
Mitberichter:	Prof. Dr. rer. nat. Alois Herkommer
	Prof. Dr.med. Dr. h. c. Arnulf Stenzl
Tag der mündlichen Prüfung:	12.06.2023

Institut für Systemdynamik der Universität Stuttgart

2023

Vorwort

Die vorliegende Arbeit entstand während meiner wissenschaftlichen Tätigkeit am Institut für Systemdynamik (ISYS) der Universität Stuttgart in der Zeit von April 2020 bis April 2023. Mein Projekt zur Problemstellung der intraoperativen Lokalisierung war Teil des Graduiertenkollegs 2543 »Intraoperative multisensorische Gewebedifferenzierung in der Onkologie«. Dieses wird gemeinsam mit weiteren Instituten der Universität Stuttgart, der Eberhard-Karls-Universität Tübingen und dem Universitätsklinikum Tübingen durchgeführt und von der Deutschen Forschungsgemeinschaft finanziell unterstützt.

Die Zeit meiner Promotion war für mich eine herausragende Erfahrung, die mir stets große Freude bereitet hat und zugleich eine Vertiefung meiner fachlichen Interessen ermöglichte. An dieser Stelle möchte ich mich bei den zahlreichen Personen bedanken, die mich während dieser Zeit auf vielfältige Weise unterstützt haben.

Mein besonderer Dank gilt meinem Doktorvater Prof. Dr.-Ing. Dr. h.c. Oliver Sawodny, der am ISYS die Rahmenbedingungen für eine tolle Forschungsatmosphäre für meine Kollegen und mich ermöglicht hat. Ich bin ihm nicht nur für seine fachliche Anleitung und seine kritisch konstruktiven Anmerkungen dankbar, sondern auch für das Vertrauen, das er in mich gesetzt hat, und die Freiheit und Verantwortung, die er mir während der Forschung gewährt hat. Prof. Dr.-Ing. André Casal Kulzer vom Institut für Fahrzeugtechnik Stuttgart danke ich für die Übernahme des Prüfungsvorsitzes. Des Weiteren möchte ich Prof. Dr. rer. nat. Alois Herkommer vom Institut für Technische Optik und Prof. Dr. med. Dr. h. c. Arnulf Stenzl von der Klinik für Urologie am Universitätsklinikum in Tübingen für ihr Interesse an meiner Arbeit, ihre Unterstützung und die Zusammenarbeit während des gesamten Projektverlaufs sowie die Übernahme des Mitberichts danken. Das Institut für Technische Optik führte wesentliche experimentelle Versuche durch, unterstützt durch Valese Aslani. Die wesentliche klinische Expertise und die notwendigen Videodaten wurden von der Klinik für Urologie in Tübingen bereitgestellt. Mein besonderer Dank gilt Dr. med. Niklas Harland und Dr. med. Simon Walz, die im klinischen Alltag die Aufnahmen von zystoskopischen Videodaten integriert und so die für das Projekt erforderlichen intraoperativen Daten akquiriert haben.

Ich schätze meine Zeit am ISYS sehr und bin dankbar für die vielen wertvollen Freundschaften, die ich während dieser Zeit aufbauen konnte. Mein aufrichtiger Dank gilt allen Mitarbeiter:innen des ISYS, die immer bereit waren, mit Rat und Tat zur Seite zu stehen und ein familiäres Arbeitsklima am Institut schafften. Marion Fleischer, Corina Hommel und Gerlind Preisenhammer danke ich für die Übernahme aller

administrativen Aufgaben, die es mir ermöglichten, mich auf die wissenschaftlichen Problemstellungen zu konzentrieren. Zudem möchte ich Dr.-Ing. Eckhard Arnold und Dr.-Ing. Michael Böhm danken, die mir als Ansprechpartner und Mentoren zu jeglichen Themen zur Seite standen, sei es in der Lehre oder in Diskussionsrunden zur Optimierung und vielen weiteren Themen.

Außerdem möchte ich meinen Projekt- und Bürokollegen Peter Somers und Carina Veil für eine Promotionszeit danken die geprägt war von intensiver Zusammenarbeit und Freundschaft. Ebenfalls möchte ich mich bei Mark Burkhardt für unsere Wohngemeinschaft bedanken, die durch eine Vielzahl von nachfeierabendlichen Nachbesprechungen des Arbeitstages geprägt war. Es freut mich, dass Franziska Krauß als meine Nachfolgerin das Forschungsprojekt weiterführt. Ich möchte mich bei Ihr für die Unterstützung vor der mündlichen Promotionsprüfung danken.

Ein weiterer Dank geht an alle fleißigen Korrekturleser des Instituts: Melanie Gschweng, Kathrin Hoffmann, Anja Lauer, Bernd Müller, Christos Parlapanis, Jonas Stiefelmaier und Frank Wolff. Weiter möchte ich meinen lieben Freunden Salehah AR, Carmen Bannert, Lena Braun und Johannes Dunke danken. Sie haben meine Arbeit nicht nur in kürzester Zeit korrekturgelesen, sondern waren auch eine konstante mentale Unterstützung während der gesamten dreijährigen Promotionszeit.

Nicht zuletzt gilt mein ausdrücklicher Dank meiner gesamten Familie für die bedingungslose Unterstützung und den großen Rückhalt. Meine Faszination für Technik und naturwissenschaftliche Fragestellungen wurde immer mit geduldiger und ausführlicher Beantwortung bohrender Kinderfragen gefördert. Schließlich wurde ich während meiner gesamten Ausbildungs- und Promotionszeit stets unterstützt.

Büchenbronn, Juli 2023 Johannes Schüle

Abstract

Minimally invasive procedures offer many advantages to patients and surgeons for diagnosis and therapy, as they significantly reduce the surgical trauma. An endoscope tailored to the specific anatomy is inserted into the body through targeted incisions or natural orifices. However, the limited access and restricted field of view in the surgical site can pose significant challenges for navigation and visual orientation. As a result, intraoperative navigation and scene reconstruction have become essential areas of focus in computer-assisted surgery and are receiving increasing attention in the literature. Nevertheless, traditional geometry-based localization and mapping techniques are inadequate for this purpose. In general, there is a lack of holistic approaches in the literature that describe a comprehensive solution for dealing with deformable intraoperative environments.

This work addresses intraoperative localization and scene reconstruction, particularly for deformable environments. Motivated by the specific challenges of bladder endoscopy - cystoscopy -, a holistic approach to intraoperative localization and scene reconstruction that meets the requirements for robustness and accuracy in intraoperative conditions is proposed. The proposed reconstruction concept relies on a monocular camera image, but can be flexibly extended to include additional sensor data. Furthermore, the presented concept enables the reconstruction of camera position, geometry, and texture based on gradient-based optimization formulations. The fundamental reconstruction strategy employed in this work follows the question: How does the model representation and camera perspective need to be adjusted such that the rendering of the model matches the current observation?

The objective of the reconstruction is to solve several optimization problems, which ultimately require fully differentiable rendering mappings. Therefore a differentiable rendering process is employed, and the inverse rendering method is proposed, in which 2D image data is projected from the camera viewpoint onto a 3D mesh model. This projection results in new mesh-bound data in a differentiable form: the intersection point with the mesh itself, the normal direction of this point, and the surface texture. By including the normal information in addition to the intersection points of the given image patterns, the associated pose of the current observation can be reconstructed. In order to determine orientation, specific landmark information must be acquired and registered accordingly. Therefore, visible vascular structures are extracted into a graphical representation and used as intraoperative landmarks that allow for a deformation-invariant description of the landmark information. A robust graph-matching method based on deformation-tolerant descriptor descriptions is presented to determine related structures. Additionally, a new structure-based

and deformation-invariant outlier classification is proposed to check given matches based on anatomical features. A match is identified as an outlier, if it requires anatomically invalid - self-intersecting - vascular structures to match. The validated matching structures are stored in a global model representation after an appropriate gradient-based pose reconstruction so that in case a new observation is made, the observed structures can be matched with all previously observed structures in a single step.

Any discrepancies between related patterns that remain unresolved after pose reconstruction are compensated for by adjusting the model geometry. However, such geometry reconstruction often results in an over-determined optimization problem. To address this, new regularization costs are proposed that take the intraoperative requirements into account. Once the current camera perspective and geometry ratios are reconstructed, the currently observed vessel structure in the model texture can be updated. Recurrent structures are also identified to establish recognition reliabilities for each individual tissue structure. This helps to limit the complexity of the reconstruction to reliably identifiable structures, which benefits the entire reconstruction process by relying on unique patterns. In conclusion, graph-based landmark registration is used to enhance the overall robustness of the reconstruction pipeline, while gradient-based reconstruction accounts for the overall complexity of geometry and texture mapping.

Kurzfassung

In der Diagnose und Therapie bieten minimalinvasive Verfahren viele Vorteile für den Patienten und den Chirurgen, da sie das Verletzungstrauma drastisch reduzieren. Ein auf die spezifische Anatomie zugeschnittenes Endoskop wird durch gezielte Schnitte oder natürliche Körperöffnungen in den Körper eingeführt. Durch den minimierten Zugangsweg sind der Bewegungsradius und das Sichtfeld des Chirurgen allerdings sehr eingeschränkt. Eine umfassende Rundumsicht ist im Allgemeinen nicht gewährleistet, sodass die Orientierung im Körperinneren für die Operierenden eine große Herausforderung darstellt. Intraoperative Navigation und Szenenrekonstruktion sind folglich zentrale Fragestellungen für die Weiterentwicklung der computergestützten Chirurgie. Dabei stellt vor allem die Erfassung deformierbarer intraoperativer Umgebungen eine große Herausforderung dar; herkömmliche Lokalisierungs- und Kartographietechniken werden der intraoperativen Komplexität im Allgemeinen nicht gerecht. Dementsprechend fehlt es in der Literatur an Ansätzen, die eine ganzheitliche Lösung für deformierbare intraoperative Umgebungen beschreiben.

Diese Arbeit thematisiert intraoperative Lokalisierung und Szenenrekonstruktion, insbesondere für deformierbare Umgebungen. Motiviert durch die spezifischen Herausforderungen einer Blasenspiegelung - einer Zystoskopie - wird ein ganzheitlicher Ansatz zur intraoperativen Lokalisierung und Szenenrekonstruktion vorgestellt, der die Anforderungen an Robustheit und Genauigkeit der intraoperativen Bedingungen erfüllt. Das vorgeschlagene Rekonstruktionskonzept setzt in der allgemeinsten Formulierung ausschließlich ein monokulares Kamerabild voraus, kann aber flexibel um zusätzliche Sensordaten erweitert werden. Darüber hinaus ermöglicht das vorgestellte Rekonstruktionskonzept die Rekonstruktion von Kameraposition, Geometrie und Textur auf der Basis einer gradientenbasierten Optimierung. Die grundlegende Rekonstruktionsstrategie, die in dieser Arbeit angewendet wird, folgt der Frage: Wie müssen die Modellrepräsentation und die Kameraperspektive angepasst werden, damit das Rendering des Modells mit der aktuellen Beobachtung übereinstimmt? Das Rekonstruktionsziel wird durch verschiedene Optimierungsprobleme vorgegeben, die letztlich vollständig differenzierbare Rendering-Abbildung erfordern. Für den differenzierbaren Renderingprozess selbst wird in diesem Zusammenhang der umgekehrte Prozess das Inverse Rendering Verfahren vorgeschlagen, wodurch Bildinformationen direkt auf eine Mesh-Oberfläche übertragen werden, sodass der Schnittpunkt mit der Modelloberfläche, die Normalenrichtung und die Oberflächentextur in einer differenzierbaren Form bestimmt werden können. Unter Berücksichtigung der zugehörigen Punktinformationen und Normalenrichtungen, die durch die abgebildeten Muster bestimmt werden, kann schließlich die zugehörige Kamerapose der aktuellen Beobachtung rekonstruiert werden. Zur Orientierung werden spezifische Landmarken

erfasst und entsprechend registriert. Zu diesem Zweck werden die sichtbaren Gefäßstrukturen extrahiert und als Graphen dargestellt und dienen als intraoperative Landmarken, die eine deformationsinvariante Beschreibung ermöglichen. Es wird eine robuste Graphmusterregistrierung auf der Grundlage eines deformationstoleranten Deskriptordesigns vorgestellt, um zusammengehörige Strukturen zu identifizieren. Darüber hinaus wird eine neue strukturbasierte und deformationsinvariante Ausreißerklassifizierung vorgeschlagen, um gegebene Übereinstimmungen auf der Grundlage anatomischer Merkmale zu überprüfen. Eine Übereinstimmung wird dann als Ausreißer identifiziert, wenn die erforderliche Übereinstimmung anatomisch unzulässige - sich kreuzende - Gefäßstrukturen erfordert. Die gültigen übereinstimmenden Strukturen werden nach einer geeigneten gradientenbasierten Posenrekonstruktion in einer globalen Modellrepräsentation gespeichert, sodass bei einer neuen Beobachtung die beobachteten Strukturen mit allen zuvor beobachteten Strukturen in einem Zug abgeglichen werden können. Etwaige Diskrepanzen zwischen zusammengehörigen Mustern, die durch die Posenrekonstruktion nicht aufgelöst werden können, werden durch Anpassung der Modellgeometrie kompensiert. Die Geometrierekonstruktion durch entsprechende Anpassung der Geometrie führt allerdings im Allgemeinen zu einem überbestimmten Optimierungsproblem. Zu diesem Zweck werden neue Regularisierungskosten vorgeschlagen, die den intraoperativen Anforderungen gerecht werden. Nach Rekonstruktion der aktuellen Kameraperspektive und Geometrieverhältnissen kann die aktuell beobachtete Gefäßstruktur in der Modelltextur aktualisiert werden. Zusätzlich werden wiederkehrende Strukturen identifiziert, um Wiedererkennungszuverlässigkeiten für jede einzelne Gewebestruktur zu etablieren. Dadurch kann die Komplexität im Abgleich auf die zuverlässig identifizierbaren Strukturen reduziert werden, sodass das gesamte Rekonstruktionskonzept von eindeutigen Mustern profitiert. Die vorgestellte Gesamtrekonstruktionspipeline erzielt Robustheit durch eine graphenbasierte Landmarkenregistrierung, während die gradientenbasierte Rekonstruktion die Gesamtkomplexität der Geometrie- und Texturabbildung Rechnung trägt.

Contents

1 Introduction

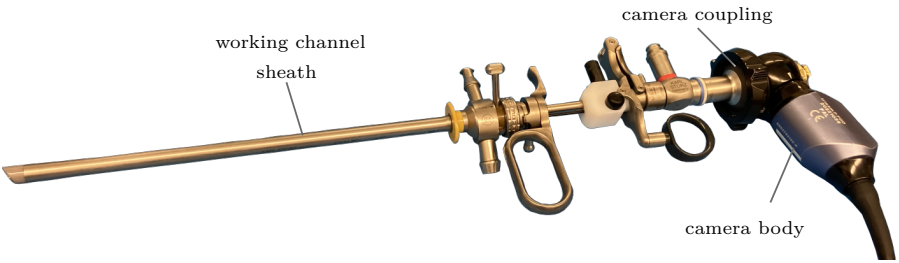
Minimally invasive surgery (MiS) is a surgical approach that aims to minimize the surgical trauma of the procedure by reducing potential damage to living tissue during the surgery. Reducing surgical trauma is a key focus in modern medicine, as the goal is to minimize the collateral damage caused by the surgical procedure while still achieving the main objectives of the respective intervention. MiS involves using a camera and instrument system called an endoscope to access the target area through small incisions or natural openings in the body, such as the mouth, esophagus, trachea, ear, nose, urethra, rectum, vagina, or tear ducts. This is in contrast to a traditional open surgery, which involves making a larger incision to access the target area.

1.1 Endoscopy

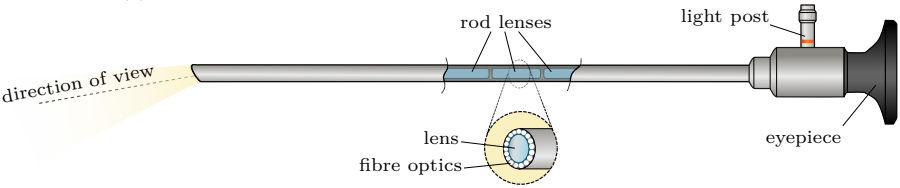
The term 'endoscope' is a neologism from ancient Greek, origin from 'éndon' and 'skopein' and means 'to observe from the inside'. A modern endoscope consists of optical lenses, a light source, and a camera sensor that can be located inside or outside the body, depending on the implementation of the optical setup.

The use of endoscopic procedures has expanded from diagnosis to complete surgical procedures, making it a current state-of-the-art procedure. For the surgical intervention beyond the diagnostic imaging, the endoscope is equipped with a working channel through which instruments can be inserted and actuated. Figure 1.1 shows a rigid endoscope with a working channel for instruments.

Endoscopy has become part of the daily routine in many medical disciplines, e.g. gastroenterology, urology, gynecology, somatic-abdominal surgery, and pneumology. There are highly specialized endoscope variants for almost every field of application – from thin metal tubes to flexible rubber tubes tailored to the respective anatomical conditions. The endoscope adapted to the anatomical constraints is usually referred to by the Latin name of the organ and the suffix scope, e.g. gastroscopy, laparoscopy, cystoscopy, hysteroscopy, arthroscopy, laryngoscopy, bronchoscopy, to take up a few selected examples from the above-mentioned medical fields, whereby the respective terminology does not describe an independent medical discipline itself, but instead refers to the respective medical technology [58].



(a) Assembly of a rigid cystoscope with an attached video camera.



(b) The components of the telescope include an array of rod lenses surrounded by fiber optics, which make up the endoscope's optical system. The cross section of the telescope reveals this arrangement.

Figure 1.1: Components of a rigid endoscope. The endoscope consists of an optical system and is connected to a video camera, allowing the surgeon to inspect the bladder during a minimally invasive procedure.

1.1.1 Advantages of Endoscopy

Compared to traditional open surgeries, minimally invasive procedures offer several benefits, including less tissue damage, smaller incisions, and reduced scarring [127]. These advantages encompass diminished blood loss, which consequently decreases the probability of complications and minimizes the requirement for blood transfusions [58]. Furthermore, patients undergoing minimally invasive surgery experience less pain after the procedure, resulting in an improved recovery experience and reduced need for pain medication [127]. These procedures are also associated with faster wound healing, as smaller incisions cause less trauma to the surrounding tissue, resulting in quicker healing and a reduced risk of infection [58, 127]. Shorter hospital stays are another benefit of minimally invasive surgery, as patients typically become mobile more quickly and can leave the hospital sooner, resulting in reduced costs for both the patient and the healthcare system [58]. For the sake of completeness, it is important to note that minimally invasive surgery may not be suitable for all patients or all types of procedures. The decision to undergo minimally invasive

surgery should be made on a case-by-case basis.

1.1.2 Limitations of Endoscopy

Despite the enormous advantages of endoscopy, various challenges, and difficulties are addressed briefly. Due to the narrow working area, the few incisions that are made must be planned precisely in order to reach the target structures. The movements of the instruments used in this process are highly dependent on the chosen anatomical approach, which can result in very limited movement. This limited movement of the endoscope means that only a small area of the surgical field can be covered, making it challenging to obtain a clear view of the surgical area. Additionally, the camera optics may not always be optimally aligned, further adding to the difficulty in achieving a clear view. Furthermore, the type of access and endoscope used can cause a leverage effect around the pivot point constrained by the anatomical structure. This leverage effect can significantly alter the force ratios acting on the tissue. Accordingly, due to the non-uniform positioning of the instruments around the anatomical pivot point, a reversal of motion is caused, where the instrument handle and tip move in opposite directions. Furthermore, various factors can compromise image quality during the intervention, such as distortions, uneven illumination, and disturbances like smoke caused by the electrical resection loop. In the worst-case scenario, bleeding or pieces of tissue can obstruct the endoscope, leading to a complete loss of vision during the procedure.

Furthermore, due to the lever-like transmission and friction of the contact surfaces, the tactile sensation, and the mechanical palpation of the surgeon are taken. In open surgery, the surgeon can obtain essential information about non-visible tissue layers by palpation; for example, indications of tumor margins as tumors typically exhibit in increased mechanical stiffness. Accordingly, due to the limited field of view and the lack of haptics, it is challenging to detect risk structures such as blood or nerve vessels solely from the image observations. In addition, the surgeon is often standing in an ergonomically unfavorable position, sometimes for several hours, bending over the patient while constantly keeping an eye on the camera image. This can lead to physical and mental exhaustion, which may have a notable impact on the accuracy and speed of the procedure.

Any unforeseen event during the operation, such as severe bleeding, is more difficult to handle due to the range of challenges. In some cases, the remaining operation may even need to be completed as open surgery. It is therefore important to work on overcoming these challenges in endoscopy and to continually improve the procedure.

1.2 Problem Description and Focus of the Work

Active computer-assisted surgery is a highly challenging and still a wide open field. This work addresses intraoperative navigation for challenges posed by deformable

environments. Localization is a prerequisite for any computer assisted sensor fusion algorithm that correlates information and provides the surgeon with the necessary information. In particular, this work addresses the open field of localization and scene reconstruction problems for deformable environments.

To meet the needs of surgeons and minimize invasiveness during interventions, it is essential to understand several concepts in medical technology which are related to the challenges at hand. Therefore, this review examines medical interventions from an engineering perspective, assessing their potential to meet the requirements of surgeons during medical procedures. The primary objective of this review is to investigate the role of localization and scene reconstruction in current trends in medical engineering, with a case study on interventions around the urinary bladder. The urinary bladder is one of the most geometrically variable organs, yet it simultaneously provides distinct and clear vascular structures.

In addition, recent advances in medical technology have emphasized multisensory tissue differentiation in oncology. By correlating data from various sources, a higher level of data validity can be achieved, which in turn requires highly reliable localization of measurement data. This review goes beyond motivating the research and provides the critical groundwork for developing an intraoperative localization framework by exploring recent medical technology developments and the need for reliable localization and scene reconstruction methods. Furthermore, it provides a detailed description of the localization and scene reconstruction requirements necessary to satisfy both medical and engineering perspectives.

1.2.1 Clinical Applications of Cystoscopy

Cystoscopy is a medical procedure that allows a physician to examine the inside of the urinary bladder and urethra using a cystoscope, as shown schematically for the bladder inside end wall in Figure 1.2. The cystoscope is inserted through the urethra into the bladder, allowing the physician to screen the condition of the urethra and bladder wall visually. To improve the visibility of all surfaces and abnormalities, the bladder is typically inflated with saline solution to increase its volume.

Cystoscopy is crucial for diagnosing and treating urinary bladder and urethral conditions like diverticula, stones, tumors, strictures, incontinence, inflammation, sphincter and prostate gland changes in men. Photodynamic diagnostics (PDD) is an advanced diagnostic technique used in cystoscopy for improved tumor detection. This method involves the administration of a specialized fluorescent dye, such as Hexvix® (developed by GE Healthcare), which is injected into the bladder through a thin catheter. Clinical studies have demonstrated that PDD increases the detection rate of bladder tumors, thereby providing enhanced assistance to clinicians during diagnosis [49]. Hexvix® is the only commercially available product for this technique, and it has been proven to be a useful diagnostic tool for detecting bladder

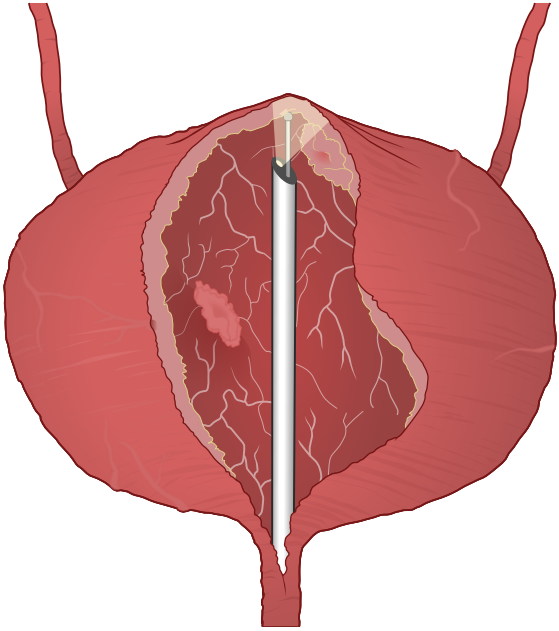
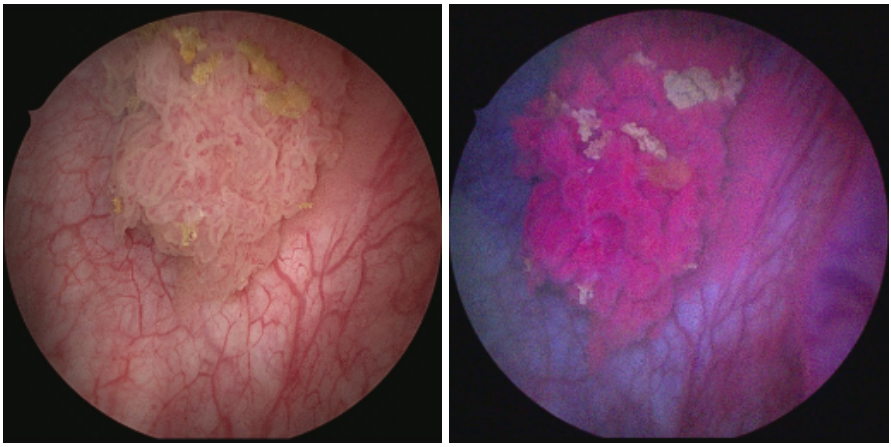


Figure 1.2: The endoscope, which is inserted through the urethra, provides the surgeon with an endoscopic camera image that enables examination of the bladder wall for abnormalities such as tumorous tissue. An additional working channel in the cystoscope gives the physician the ability to perform interventions with various tools or sensors. Furthermore, the vascular structures provide landmark information for orientation during the procedure.

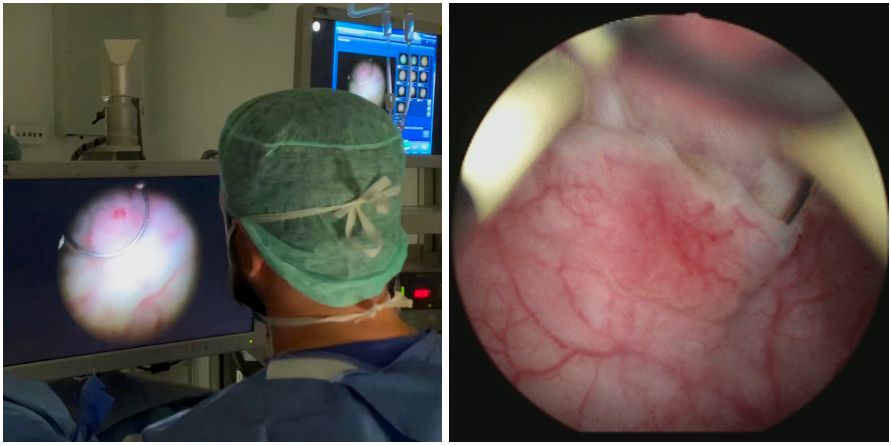
tumors. Upon irradiation, tumor cells absorb the fluorescent dye, causing them to fluoresce red when illuminated with blue light. This process significantly enhances tumor visualization and allows for more accurate diagnosis and treatment. The PDD technique is considered safe for patients since the dye is harmless and does not cause any adverse reactions. Figure 1.3a and Figure 1.3b demonstrate how PDD improves tissue visualization by showing tumor tissue under white light illumination and blue light fluorescence illumination, respectively.

In addition to visual inspection, a physician may also take a small tissue sample (a biopsy) from suspicious areas for further examination by a pathologist. To do this, the bladder volume is usually reduced to minimize the risk of damaging deeper tissue layers during the biopsy procedure. The necessary instruments are inserted through the working channel of the cystoscope to remove the tissue sample or to ablate superficial tumors using an electrical resection loop, as shown in Figure 1.4. An electrical cutting loop is a thin, wire-like device connected to an electrical current, which cuts the tissue through thermal development. This method has the advantage



(a) Recording under white light illumination. **(b)** Recording under blue light illumination.

Figure 1.3: Under white light illumination, tumor boundaries are only vaguely discernible; compared to photodynamic diagnostics, the tumorous tissue illuminates in its typical red fluorescence and can accordingly be identified.



(a) Cystoscopic surgery set-up. **(b)** Resection of suspicious tissue.

Figure 1.4: During a minimally invasive transurethral resection, the surgeon employs an electrode resection loop through the working channel of the cystoscope to remove abnormal tissue.

of simultaneously cauterising the tissue, which helps to stop bleeding by occluding affected vessels, as opposed to mechanical cutting with a scalpel.

The European Urological Society recommends documenting all abnormalities, tumor macroscopic features, and mucosal irregularities on a bladder map during cystoscopy, as illustrated in Figure 1.5 [135]. In clinical practice, this involves manually marking the positions of biopsies, abnormalities, and removed tumors on a paper map so that the biopsy results can be appropriately associated with the relevant locations in the event of subsequent pathological findings. However, this method is qualitative, prone to individual variability, and challenging to maintain in a busy clinical environment. A computer-assisted documentation framework has the valuable potential to enhance accuracy and efficiency in recording and tracking crucial cystoscopy information.

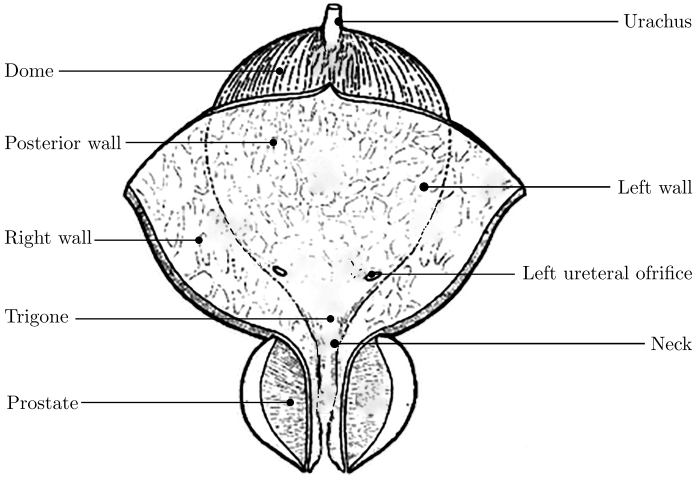


Figure 1.5: A bladder map used to document the qualitative location of abnormalities on the bladder wall. Confer with [102, 135].

The development of an automated procedure that accurately documents the locations of biopsies, abnormalities, and excised tumors could significantly enhance data correlation, leading to improved diagnostics based on an enlarged data basis. For the treatment of bladder carcinoma, a second transurethral resection of the bladder is generally required within two to six weeks of the initial resection. In this scenario, reconstructing the patient-specific surface of the bladder could furnish surgeons with vital information about areas of concern identified during the first procedure [110]. This approach would allow for a detailed evaluation of raw data in conjunction with current observations, facilitated by multi-modal data correlation. For example, information obtained from the initial tumor resection can be instrumental in guiding the diagnosis and treatment strategies during subsequent resections. This method-

ology is consistent with the principles of multi-modal data correlation, which involves assimilating data from various sensor types across different physical domains (e.g., imaging, impedance measurements, mechanical assessments, optical evaluations, etc.) to refine diagnostic accuracy. Presently, multi-modal data correlation is a burgeoning research field due to its potential to provide a more thorough and diverse range of information, thereby enhancing the final tissue classification [139].

1.2.2 Review on Multimodal Data Diagnostics

Novel multisensory methods for tissue differentiation may help to improve the accuracy and efficiency of tumor resections in the future. These methods involve using sensors to measure and analyze the properties of the intratumoral[1] and peritumoral[2] milieu, including changes in the mechanical, electrical, optical, and biochemical properties of tissue. Tumor tissue exhibits altered morphology and biochemical composition compared to healthy tissue, and these changes can be detected through sensors that measure, for example, changes in stiffness, capacitive properties, and water and salt content [9, 12, 111, 136]. By combining the information from multiple sensors, it may be possible to achieve a more reliable and accurate differentiation between tumor and healthy tissue, which helps to reduce the radicality, duration, and complication rate of the operation [115, 139].

Multi-modal approaches, including the evaluation of tissue stiffness through elastography, have been successfully applied in various medical contexts, such as the diagnosis and treatment of urinary incontinence, and the integration of preoperative and endoscopic imaging data [4, 17, 52, 139]. In the context of tumor resections, the use of multi-modal approaches may be beneficial for improving the accuracy of diagnosis and treatment planning, as well as for guiding the surgeon during the procedure. Furthermore, ongoing research explores the integration of various preoperative diagnostic techniques for breast cancer and the combination of preoperative and endoscopic imaging data [99].

In a broad and schematic representation, multi-modal sensor classification is depicted in Figure 1.6. However, several challenges must be addressed to effectively implement and utilize multi-modal approaches in surgery. To analyze and merge the intraoperatively acquired multimodal sensor data, raw sensor signals must be accompanied by coherent spatial information [125]. As a result, precise knowledge of local positions and orientations of tools and corresponding measurement points is essential and constitutes a critical aspect of any subsequent classification task. This requires the development of robust and reliable tracking and localization techniques, which can be challenging due to the complexity and variability of the surgical environment.

[1] Intratumoral refers to a location or process occurring within a tumor.
[2] Peritumoral refers to the area surrounding a tumor, which often includes healthy tissue and blood vessels.

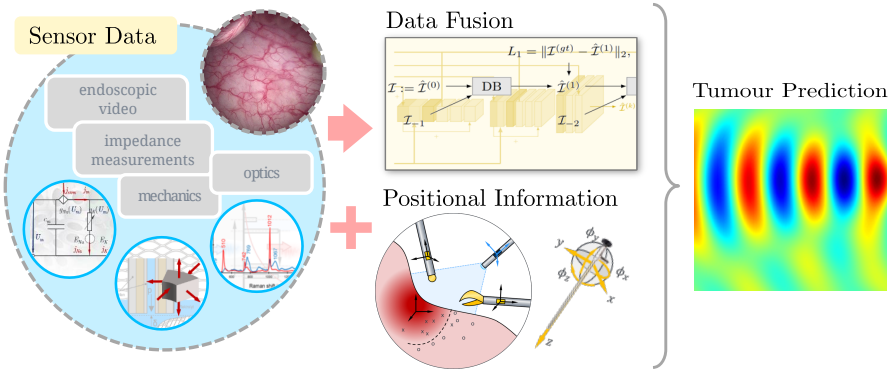


Figure 1.6: Multi-sensory determination of tissue properties across various physical domains holds great promise as a reliable technique for predicting tumor boundaries. However, for any type of multi-modal data fusion of spatially distributed data measurements, a corresponding position informant for the respective sensor measurements is required. Registration involves incorporating these measurements into a sensor fusion network of the tissue at the correct corresponding locations. The sensor localization task deals with registration and obtaining the correct surface positions with respect to the suspicious tissue and is, therefore, a vital part of surgical navigation.

1.2.3 Problem Statement

The challenge at hand is to achieve accurate intraoperative localization in deformable environments, encountered during endoscopic surgery, while fully utilizing the potential of multi-modal sensor information. Currently, it is difficult to identify the precise location of tissues, organs, and other structures within the surgical site, especially when the surrounding tissues are altered or distorted. This results in difficulties in avoiding damage to critical structures, precisely targeting specific areas for biopsy or removal, and orienting surgeons for re-screening suspicious tissue in subsequent interventions.

In addition, precise localization is essential to make a multi-modal sensor fusion of various sensor measurements, taken at different locations, feasible. Inaccurate localization can hinder proper alignment and combination of data from different sensors, potentially leading to errors or misinterpretations. To overcome these challenges, solutions adapted to the deformable environments are necessary. Intraoperative localization concepts are needed that enable the surgeon to locate structures within the surgical site more precisely and maximize the potential of multi-modal sensor information. Potential solutions must adapt to changes in tissue shape or position and provide guidance for the placement of surgical instruments and the orientation

of the surgeon. The primary focus is to provide a solution that enhances the accuracy of endoscopic surgeries and enables the development of multi-modal sensor fusion concepts, particularly for deformable environments.

1.3 Analyzing Scene Reconstruction Algorithms: A Review of the Literature and Future Directions.

Endoscopic vision has been the subject of extensive research and advancements by various laboratories and endoscope manufacturers. Currently, rigid mono-endoscopes are the most commonly used systems in clinical settings for laparoscopic interventions. This literature review aims to provide a comprehensive overview of the current state and future developments in advanced endoscopic vision, including navigation, tracking, depth estimation, and 3D surface reconstruction. With a focus on laparoscopic interventions, the surveys presented in [31, 105, 128] give a general overview of the current literature on intraoperative localization and scene reconstruction. In order to obtain a comprehensive understanding of the current state of endoscopic navigation, this literature review takes a broad approach by looking beyond laparoscopic interventions and providing an overview of navigation and localization algorithms. However, a comprehensive solution that addresses deformable environments remains elusive as the problem is the subject of ongoing research.

Physical and Geometric Reconstruction Techniques: The Simultaneous Localization and Mapping (SLAM) algorithm is a state-of-the-art solution for solving the problem of localization and mapping in unknown environments in robotics. It creates maps of the environment while simultaneously determining the system's location within the map. In the field of intraoperative environments, SLAM algorithms have been utilized in various works to reconstruct tissue surfaces using a monocular endoscope [10, 36, 43, 89, 97, 119, 146].

Visual SLAM (V-SLAM) algorithms are typically classified into two categories: direct and indirect methods. Direct methods use the entire image, relying on image-level changes such as optical flow for precise reconstruction of the 3D environment. However, they can be vulnerable to larger changes between images [1, 53, 130]. In contrast, indirect methods use intermediate representations such as sparse representations, instead of raw image data, and rely on recognizable image features to determine landmark positions, providing robust results in challenging environments. The so-called ORB landmark feature detection is recognized as one of the state-of-the-art feature detection algorithms for indirect methods in robotics [109]. ORB stands for Oriented FAST and Rotated BRIEF, two algorithms used for feature detection and description in Visual SLAM (V-SLAM) methods for robotics. FAST is an abbreviation for Features from Accelerated Segment Test [138], a corner detection algorithm that identifies key points or interest points in an image. BRIEF is short

for Binary Robust Independent Elementary Features [13], a feature descriptor that produces a binary string as output for a given image patch. Together, these algorithms make up the ORB landmark feature detection algorithm, which is considered a state-of-the-art feature detection method for a wide range of image registration tasks in robotics.

Based on this, ORB-SLAM, which has been presented in its latest versions [14, 82, 83], is a well-established VI-SLAM algorithm that leverages ORB features for landmark determination. It has demonstrated its versatility in both rigid robotic environments and intraoperative scenarios [14, 71, 97]. Despite the focus on rigid scenarios, ORB-SLAM has inspired further research in handling deformation for surgical scenes [61, 71, 95, 126]. In [18], ORB-SLAM [14] is used for scenarios with slow, quasi-rigid deformations between two images, which allows the algorithm to handle small deviations and iteratively adjust the pose and scene based on previous images. However, in this approach, the pose and map reconstruction may experience drift if deformations occur over an extended period. The method attempts to approximate deformations using single rigid registrations to address this. Some studies tackle SLAM in a known environment, where the mapping is given a prior with an anatomical geometry model [78, 89, 134]. In [134], a patient-specific anatomical MRI model was utilized as a solution to address the challenge of navigation in the endonasal skull base. The method simplified the SLAM problem by transforming it into a localization problem within the provided geometry map. The algorithm updates the features in the map but does not determine the spatial information itself. An evaluation was conducted on an endonasal skull surgery of a pig, which achieved a pose accuracy of less than 1mm with the aid of an optical tracking system. However, the authors note that the precision of the reconstruction is limited by the accuracy of the pre-operative geometry model.

In [80], a solution for handling deformations in non-rigid environments, particularly during lung endoscopy, is presented. This approach involves continuously updating the model surface based on a lung motion model, which effectively addresses the non-rigid environment and produces accurate results in the presented work. A real-time capable simultaneous finite element model (FEM) simulation based on [28] is employed in [114] to address deformations in intraoperative scenes. This method takes into account the underlying deformation and incorporates available model information, such as the organ's Young's modulus or the forces generated by the instruments, to predict the deformation in the scene. The SLAM algorithm then uses the updated map, reflecting current observations. The proposed framework can be applied to any deformable scene in an intraoperative environment, but requires extra sensor data and extensive patient-specific initialization information for any practical surgical application. Despite these challenges, the method offers the advantage of considering underlying deformations and incorporating available information for improved accuracy [30].

Similarly to V-SLAM algorithms, the Structure from Motion (SfM) algorithm uses a

batch of images to calculate and optimize the map all at once. Although this method can achieve higher accuracy and a denser map reconstruction compared to the online SLAM algorithm, it relies on the assumption that there were no deformations between the images in the batch [145]. Nevertheless, combining SfM with SLAM can enhance the accuracy and robustness of 3D reconstruction in scenarios with deformations. Non-Rigid Structure from Motion (NRSfM) [34, 46] is an extension of SfM, specifically designed for handling deformable environments. However, SfM's requirement for offline processing limits its feasibility for real-time applications. To overcome this challenge, the field of 3D reconstruction of deforming scenes has made significant progress through the integration of SfM and SLAM algorithms.

The fusion of these two algorithms can effectively address the limitations of traditional SLAM algorithms, such as drift errors [61]. This is because the SfM approach prioritizes the accuracy of reconstructed 3D structures. However, traditional SfM methods are not suitable for real-time applications as they rely on offline batch processing. This limitation is overcome by the DefSLAM library [61], a monocular SLAM solution designed specifically for deformable scenes. The DefSLAM algorithm integrates Sparse Feature from Tracking (SfT) and Non-Rigid Structure from Motion (NRSfM) through a parallel-threaded fusion of separate map reconstructions. The SfT operates in a tracking thread for faster performance, while NRSfM runs in parallel in an optimization thread to reconstruct the deformed map. Therein, the ORB-SLAM [14] serves as the backbone of the algorithm, providing online localization and feature registration. This approach offers a promising solution for real-time 3D reconstruction in deformable environments. However, the DefSLAM algorithm still faces challenges in maintaining stable reconstruction in the presence of viewing loss. To tackle this problem, statistical information is integrated into the localization procedure to improve the likelihood of finding similar landmarks when the view is regained [35]. Despite exhibiting promising results, the DefSLAM algorithm still depends on the clear retrievability of landmark features. To overcome this limitation, some works even rely on manually specified landmark correspondences to explore the deformation aspects from a more methodical perspective [73].

Another effective non-rigid SfM approach for handling cardiac surgeries is presented in [40]. This approach used only image segments of the same phase in the cardiac cycle for reconstruction, providing an elegant solution to the deformation of the heart muscle as the same heart deformation repeated over the cycle. Building on the proposed concept for cardiac surgeries, the authors extended their approach to the liver in [84].

Accurate and robust landmark information is crucial for both SLAM and SfM algorithms. However, identifying landmarks in intraoperative settings can be challenging due to factors such as blurred textures and underlying deformations. To address this challenge, ongoing research is focused on developing tailored landmark extraction methods that are more robust in such environments. One promising concept is leveraging the vascular structure of the retina, as suggested in [23]. For example,

based on that, the EyeSLAM approach [10] facilitates the vascular structure of the retina to enhance mapping and orientation during intraocular microsurgery.

When landmarks are absent, the Shape from Shading (SfS) concept offers a viable alternative method for determining relative depth information solely based on the variations in shading originating from the scene's reflectance properties and overall brightness. The approach assumes that the surface reflects light equally from all viewing angles. This means that when light hits the surface, it scatters in all directions, resulting in uniform brightness from all viewing angles. Thus, the SfS approach is particularly well-suited for homogeneous textures and uniform depth changes [128]. Despite this, the use of SfS in endoscopy continues to pose a complex challenge due to its lack of robustness in handling various endoscopic surface conditions. Previous works, such as those described in [64, 80], offer a comprehensive overview of various perspectives, including perspective-based methods [41, 140] that face robustness and accuracy issues with varying lighting conditions. The research presented in [62] explores the potential of combining SfS methods with feature-based or stereoscopic approaches to overcome the limitations of each. In [51], a novel combination of SfM and SfS was introduced, leveraging SfS for surface reconstruction at each time sample and SfM for the temporal registration of corresponding surfaces via feature detection. The proposed method was evaluated through the use of virtual colonoscopy images.

Data Driven Reconstruction Techniques: Recent advances in research have led to the development of data-driven approaches for scene reconstruction, offering the advantage of being specifically tailored to meet the needs of a particular scene. One of the pioneering works in this field is presented in [27], which proposed one of the first depth prediction networks based on Convolutional Neural Networks (CNNs).

Supervised Monocular Depth Estimation is a method that uses a single image as input and a ground truth depth map as a supervision signal to estimate the depth of each pixel in the image. It can be viewed as a regression problem, where the goal is to predict the continuous depth value for each pixel. Recent studies have demonstrated promising results in this area, with both [76] and [59] reporting strong performance using a fully convolutional residual network (FCRN) as the fundamental architecture. Recent studies have combined Generative Adversarial Networks (GANs) with common SLAM and SfM algorithms to provide ground truth data for training. The works in [145] and [16] use depth prediction to obtain depth information from monocular image data and register the image and depth observations using a well-known SLAM algorithm. However, the existing methods for supervised training have limitations. For example, they may be constrained by the limited availability of ground truth data. Furthermore, if SLAM and SfM algorithms are used to provide the ground truth data, the accuracy may be hindered by the same challenges faced by SLAM and SfM, including weak textures, reflections, and tissue deformations. As a result, beyond the architecture itself, data driven scene reconstruction concepts are

inherently limited by the quality of the available data.

To overcome the limitations of supervised depth estimation, unsupervised approaches have been proposed as an alternative solution. In these methods, instead of relying on a ground truth depth map, the network is trained to infer depth from a single image. Early works in this field trained their models on a per-patient basis, while later works such as [100, 123] employed a synthetic training set and employed domain adaptation through a Generative Adversarial Network (GAN) architecture. The GAN is trained to generalize the model's behavior to real-world observations, allowing the use of real-world data in an unsupervised manner. However, the training of a GAN can be challenging to balance, and the results may not be reliable.

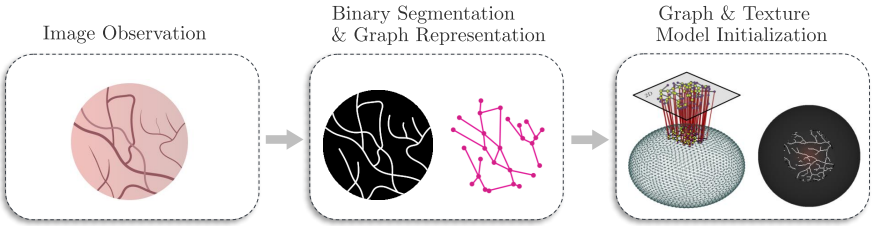
The limitations of the network's predictions are attributed to the adaptation of real data to synthetic data during the training process. This approach relies on finding the most equivalent representation in the synthetic environment, which fails to account for physical-based constraints. As a result, the predictions may be incorrect or misleading, particularly in scenarios that were not accounted for during training.

Recent studies [72, 77] have reviewed the current image-based localization methods for minimally invasive surgeries and identified significant limitations in their robustness and concerns regarding the evaluation methods used in the literature. Based on these findings, it is concluded that current surface reconstruction systems are not yet suitable for clinical use and require improvements in robustness before they can be integrated into clinical workflows. Non-rigid registration is a key challenge highlighted by both studies, posing difficulties for accurate orientation and significantly limiting the robustness of the available methods.

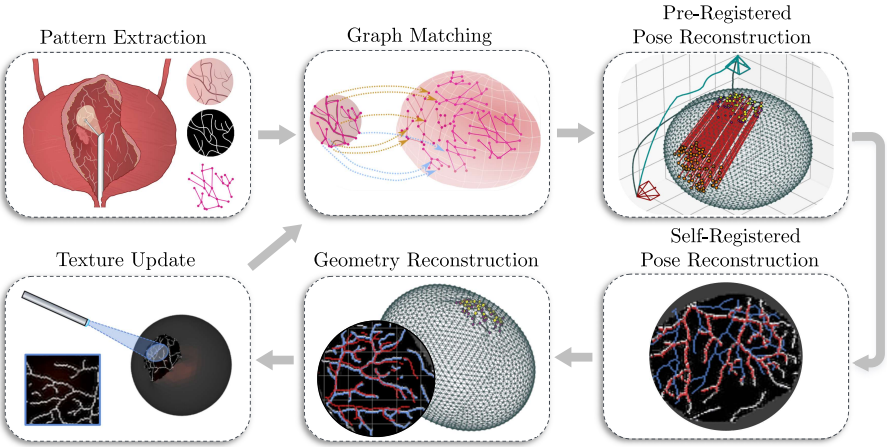
1.4 Reconstruction Concept Proposed in this Work

This work takes a new perspective on solving the problem of intraoperative reconstruction and localization. The main contribution of this work is a holistic approach to the intraoperative reconstruction problem, which aims to achieve both robustness and high accuracy in the reconstruction results simultaneously. In order to achieve the goal, a hybrid approach is proposed that combines a graph-based pattern mapping technique for robustness with a gradient-based reconstruction optimization to satisfy accuracy requirements.

To facilitate orientation and reconstruction, distinctive structures are extracted from an initial image observation. Vascular structure courses are represented using binary segmentation and graph representations. Figure 1.7a illustrates the schematic flowchart of model initialization, displaying and initializing corresponding patterns for an image observation. The simultaneous representation at the graph level and pattern progression at the pixel level provides a robust and lightweight representation through graph depiction, while the texture representation allows for precise high-resolution representation.



(a) Initialization of the model representation by the binary skeleton and graph pattern of the initial image observation.



(b) Reconstruction process for consecutive observations: The camera pose is sequentially reconstructed using the graph pattern, which provides a robust foundation for reconstruction relative to the initial camera position. Skeleton patterns are employed in the sequential reconstruction to refine the camera pose. Subsequently, any remaining discrepancies on the image plane of corresponding patterns are addressed by adjusting the model geometry according to the camera reconstruction. Upon reconstructing the pose and geometric relations, the model representation is updated with the newly observed texture information.

Figure 1.7: Overall reconstruction pipeline, including initialization and reconstruction scheme for consecutive observations.

Once an initial model representation is established, any consecutive observation can be oriented relative to the given model representation. The reconstruction process for successive observations is shown in Figure 1.7b. For each new observation, the respective graph pattern is extracted and registered with the global graph representation. This registration enables an initial pose reconstruction based on the deviation of the registered graph patterns from the global graph. Due to potential inaccuracies in graph extraction and matching, a finer pose reconstruction is performed by pattern-matching the entire vascular structures at the pixel level with the model rep-

resentation. The pose reconstruction facilitated by the graph allows for assignment independent of the initial pose, ensuring reliable assignment in the observed area regardless of the initial pose. The subsequent direct pattern matching provides the necessary accuracy but is highly sensitive to initial conditions. The remaining deviations of the associated patterns, which could not be resolved by adjusting the camera pose, are attributed to the underlying deformation of the scene between the previous reconstruction and the current observation. The remaining discrepancy between the model and current observations is resolved by adjusting the geometry. Finally, the global graph model and texture are updated with the latest observations after adjusting the camera pose and geometry, ensuring that the model representation reflects current conditions and is ready for new observations.

1.5 Delimitation of the Work & Main Contributions

The main reconstruction concept in this work involves a parameterized model representation of the organ. This is adjusted such that the model rendering accurately reflects the real-world image observation. Traditional rendering processes often involve various discrete, non-differentiable sub-operations. To overcome this, the study leverages recent advancements in differentiable rendering processes. This work builds on the most recent developments in the field and significantly extends the work of [65] and [101].

The main contributions of this work include the application of differential rendering to intraoperative navigation and the integration of computer vision techniques that differ from the current state-of-the-art. These individual contributions build up the overall reconstruction pipeline, which is illustrated in Figure 1.8 and described in detail as follows:

Endoscopic Projection Model: One of the contributions of this work is the mathematical model that represents the endoscopic imaging process, encompassing a general organ model, the kinematics of the endoscope, and the digital camera imaging. The modeling of the endoscopic kinematics and optics is especially important, as the kinematics-based projection of the image coordinates does not align with the actually observed image. The result of this comprehensive mathematical representation of the entire imaging process is a differentiable formulation that can be used in a supervised, gradient-based reconstruction of the intraoperative scene from given image observations.

Differential Rendering for Intraoperative Scene Reconstruction: A reliable geometry adaptation is crucial because any enumerated orientation and evaluation process becomes obsolete if the most fundamental size relations are not correctly reflected.

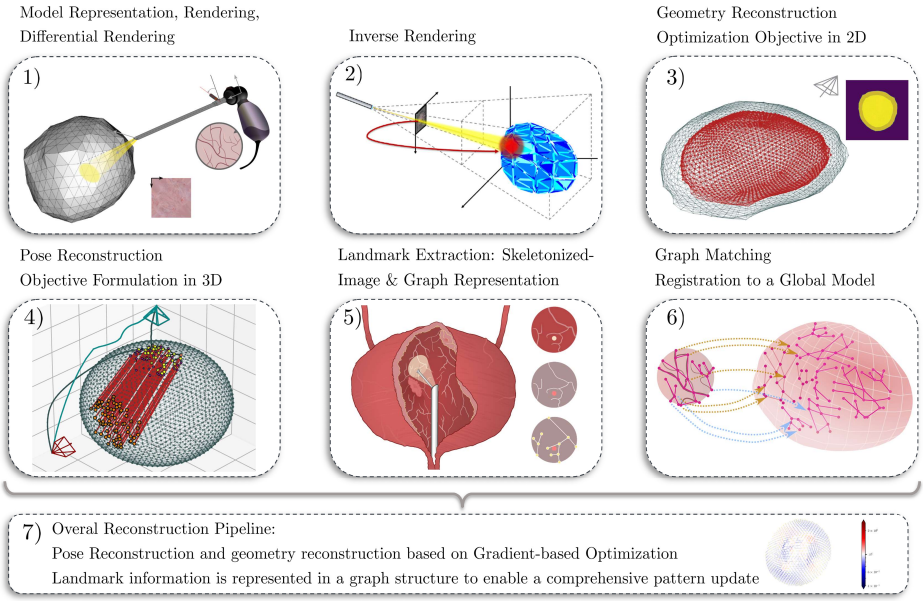


Figure 1.8: The main contributions of this work are: 1) a parameterized model representation of the organ and endoscopic projection, 2) an inverse rendering process for determining spatial information from pixel data, 3&4) a differentiable scene representation and tailored geometry regularization losses facilitate the use of a single silhouette observation for geometry and pose reconstruction, 5) a vascular structure extraction algorithm for robust landmark determination, 6) a graph representation for landmark registration, and embedding graph observations into a global representation enables comparison and updates with previously seen patterns in a single calculation, and 7) a comprehensive reconstruction pipeline that integrates all these concepts to achieve robust pose, geometry, and texture reconstruction from a single monocular image observation.

Therefore, and to resolve the over-determinacy given at the geometry reconstruction, new regularization losses are presented that are adapted for intraoperative conditions. For instance, a scale-invariant regularization error is proposed to ensure that the observed scale on the image plane is accurately propagated to the entire 3D model reconstruction, including regions not visible in the current image. This approach addresses the specific needs of cystoscopy, where the bladder volume must be continually adjusted during surgery. This includes the potential exploitation of silhouetted image data, which for example is highly advantageous in laparoscopic surgeries where landmark data is limited due to visual constraints.

Inverse Differential Rendering: A significant contribution of this work is the development of an inverse rendering process, which is a differentiable design of the inverse mapping of the traditional rendering process. While conventional rendering maps model information to the image plane, the proposed inverse rendering allows for the transformation of image information directly to the model surface. This means that image pixels can be registered directly on the model, representing position information on the image plane. For each surface position, the corresponding normal directions and feature values can be determined. The ability to directly supervise related features in 3D space using surface position and normal directions opens up new opportunities for formulating optimization objectives. This results in a more robust pose reconstruction using 3D information rather than 2D information. This is a methodological contribution to the field of computer vision that improves the accuracy and efficiency of the pose reconstruction considered in this work.

Vascular Graph Extraction for Landmark Information: The visible vascular structures are used as features for orientation, so-called landmarks, which correspond to the intraoperative complexity and thus help to provide robust orientation in this work. The practice of representing vascular structures as graphs has its roots in research that focuses on retinal analysis, where retinal structures are represented as graphs to provide essential underlying and diagnostic information. While previous research dealt with retinal structures in a static environment, the graph information in this study is specifically formulated to address the needs of deformable environments. Therefore the structural information are embedded in node positions. In addition, the extraction of edge information, including edge lengths and curvature, improves the representation's ability to retrieve associated patterns. In the context of medical imaging, the extraction of edges is a crucial step in the analysis of vascular structures and is a well-known challenge. However, traditional edge extraction methods can produce unreliable results in the presence of lighting variations and image artifacts, and they depend heavily on the preprocessing pipeline. To address these limitations, a novel data-driven approach utilizing a deep neural network architecture is proposed in this work. The approach accounts for image uncertainties and enables reliable identification of connected structures. This contribution provides real-time edge extraction capabilities, advancing the field towards more robust and reliable landmark identification.

Graph Matching, Outlier Removal, Global Graph: Based on extracted graph features in the image plane, a novel approach is proposed, which includes graph matching, deformation-invariant outlier elimination for vascular structures, and the construction of a global graph representation. These steps are designed to comprehensively describe all observed patterns in a global model representation. This approach provides the following individual contributions to intraoperative landmark-based localization.

(i) **Descriptor-based Node Matching:** Based on extracted information from the graph, robust descriptors are defined to express the structural properties of a location within the graph. These descriptors enable the comparison of patterns from different observations and determine their similarity. The descriptor-based matching follows conventional procedures found in literature, with the key difference being the design of the descriptors, which incorporates graph information. This approach provides robustness against deformation, as the structural integration of graph information is fully deformation invariant. Additional descriptors are designed to be scaling invariant and can be updated if a geometry update is available.

(ii) **Structure-based Outlier Removal:** In the descriptor-based matching method, erroneous assignments of landmark information inevitably occur due to ambiguous descriptor similarities. This issue is common among many prevalent matching methods in the literature. The usual approach is to verify that matches agree with the camera model consensus, and outliers are then eliminated. However, this concept is limited to rigid observation environments and is invalid for deformable environments. This work introduces a new deformation invariant structure-based outlier classification that employs the anatomical vascular structures for outlier classification. The proposed deformation invariant outlier elimination is developed for 2D to 2D graph matching and generalizable for 3D graph comparison as needed in the remainder of the work.

(iii) **Global Graph Representation:** For robust pattern matching, a 3D global graph model is created to represent all previously seen patterns, providing a patient-specific map of vascular structures. Unlike other localization methods in the literature, the current observation does not need to be matched to previous image observations on the image plane. Instead, it can be matched with the global graph representation containing all detected patterns at once. This allows all descriptor information to be updated simultaneously with the observed geometry deformation, even for areas not captured in the current view. It also allows for the matching to proceed smoothly when the view is restored after a temporal perturbation.

The global graph representation is created using gradient-based reconstruction and inverse differential rendering. Inverse rendering transfers 2D image information to the model surface, while gradient-based geometry reconstruction updates the model geometry to match the spatially dependent landmark descriptors to the current observations. To ensure that the global graph representation contains all necessary structures and is not overloaded with outdated or unreliable structures, patterns are individually checked to determine which are new in the current observation and not yet present in the global graph, as well as which patterns are present in the global structure but are no longer observable in the current graph extraction. In addition, a reliability metric is designed, which is determined for each landmark struc-

ture individually to ensure that only the most reliable structures are used for localization.

Holistic Intraoperative Scene and Pose Reconstruction: The presented approach involves a holistic reconstruction method for cystoscopy's intraoperative needs and challenges, based on the inverse rendering method, graph extraction, and matching. The method combines the advantages of the gradient-based and graph-based matching concepts to ensure the reconstruction's robustness and accuracy. The graph-based matching method uses vascular landmark structures to provide a robust mapping from any view to previously seen patterns, which can still be effective even if there is a temporary loss of view. Although the extracted landmarks and established matches can be disrupted, the graphs still yield a robust orientation. Meanwhile, using inverse differential rendering, the gradient-based reconstruction method allows for a refined pose reconstruction with increased accuracy by leveraging the entire vessel structures. As a result of the pose reconstruction, any remaining deviation of the model observation from the current image observation is resolved by adjusting the model geometry. The pose and geometry reconstruction are sequentially executed, ultimately resulting in matching patterns. Previously unmapped patterns can be updated through appropriate pattern matching in the texture model.

1.6 Structure of the Work

The remainder of the work is structured as follows: This work is structured around the proposed holistic reconstruction pipeline, which outlines the methods and techniques that are all deployed in the proposed intraoperative reconstruction pipeline.

In Chapter 2, the geometric relations between image projection and the 3D environment are established. In this context, the physical ray projection from the 3D environment to the image plane is introduced first, followed by the synthetic imaging process that goes from model representation to the digital image plane. As a result, the specific kinematics and the camera of the endoscope are taken into consideration. For the rendering process, a triangular mesh model is introduced to parameterize the human bladder. The rendering process, specific to the model representation, is discussed for differentiability. Building upon that, a probability-based formulation is presented to resolve discontinuities presented by state-of-the-art rendering pipelines. Therefore, image rendering is discussed for both texture and silhouette information.

Building on this foundation, the proposed gradient-based reconstruction is explicitly formulated in Chapter 3. Geometry and texture reconstruction are discussed for intraoperative scenarios. To this end, adapted regularization losses for geometry and texture reconstruction are presented.

In Chapter 4, the technique of inverse differential rendering is proposed as a way to enhance the accuracy and reliability of the reconstruction process. This method

transforms image information into corresponding 3D point information and normal direction on the model surface in a differentiable form. Thus, more precise information for supervision is established such that surface information can be included in the optimization objective rather than relying solely on error formulations on the image plane.

In Chapter 5, a pipeline is introduced for extracting visible vascular structures to provide reliable landmark information. This includes highlighting the vascular structures in the respective image, as well as presenting the graph extraction process, which encompasses the edge extraction network.

The corresponding graph matching procedure is presented in Chapter 6, where a global graph representation is used to propose a deformation-tolerant matching and outlier removal process. This chapter also covers the process of updating the graph for newly seen patterns and a geometry update to ensure reliable spatial ratios.

Finally, in Chapter 7, the holistic reconstruction pipeline is presented, where a proof of concept validation for pose and geometry reconstruction is provided. Additionally, the flexible solution of the reconstruction formulation is demonstrated for the concept of reconstruction of in-plane deformation.

2 Rendering Pipeline following the State-of-the-Art

A gradient-based optimization formulation is proposed to address the endoscopic scene reconstruction problem. State-of-the-art computer vision techniques are employed to tackle image reconstruction from a top-down perspective, optimizing model parameters so that the synthetic model accurately represents the recorded real-world data. Consequently, the reconstruction problem is formulated as a gradient-based optimization problem with the objective of minimizing the difference between synthetic and observed data.

To accomplish this, it is necessary to establish an appropriate linkage between the endoscopic real-world observations and the synthetic model observations. Therefore, the digital image rendering process must be designed in accordance with the physical endoscope set-up to facilitate the comparison between real-world observations and model adaptation during scene reconstruction. In the realm of computer graphics, rendering entails generating a digital image of a synthetic scene representation by projecting the model's information onto an image plane while considering the camera's perspective and lighting conditions. Furthermore, the model representation, including its geometry and texture, is combined with the extrinsic scene parameters and the camera to create a realistic image. Figure 2.1 illustrates the synthetic scene, encompassing the model representation, endoscope kinematics, and lighting conditions.

In contrast to a rigid camera system, modeling an endoscopic image recording requires accounting for kinematic and optical degrees of freedom. This is due to the connection between scene illumination and rotational movements with the endoscope's kinematics. For instance, in a rigid-fiber optics endoscope system, the optical relay system may yield unexpected image results. In such cases, the image-based camera reconstruction is not situated on the physical endoscope body but at the opposite end of the endoscope, referred to as the tool center point (TCP) in this context.

Additionally, the inherent rotational degrees in a fiber optic endoscope system can cause the observed camera image sequence to diverge from the anticipated physical rotation of the TCP. This misalignment results in a discrepancy between the TCP pose and the image-based pose reconstruction.

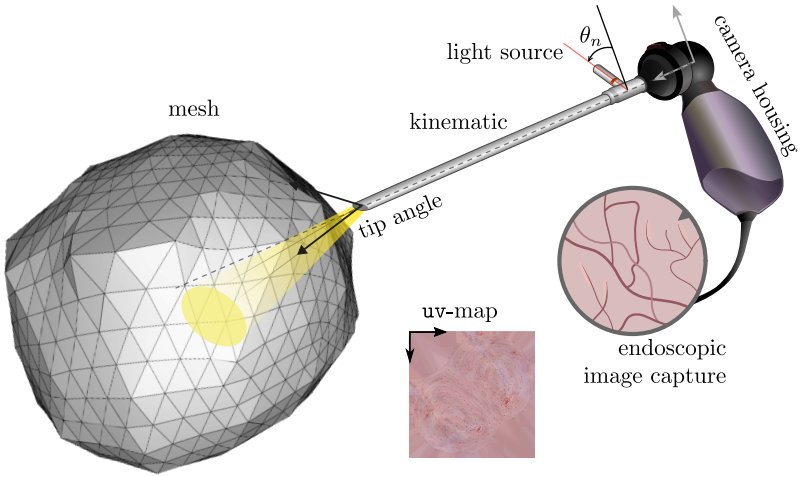


Figure 2.1: Overview of the rendering process. The synthetic model representation consists of a mesh model which describes the geometry of the organ and corresponding texture for modeling the tissue surface. The computation of a synthetic image of the model scene depends on lighting and camera optics, as well as endoscope kinematics. All dependencies must be considered to calculate a synthetic image that can be used to supervise the model adaptation.

To comprehensively model endoscopic imaging, it is essential to incorporate relevant kinematics into the camera projection model while adhering to established rendering procedures. This strategy also enables the linkage between the endoscope's pose and the image rendering outcome. This chapter utilizes methods from the domains of ray optics, computational photography, computer vision, and imaging science to achieve the following research objectives:

- Developing a mathematical model that integrates the kinematics and ray optics of an endoscopic camera system, providing a representation of real-world endoscope imaging.
- Constructing a synthetic model representation capable of depicting the deformable surgical site encountered during a cystoscopy.
- Implementing an efficient synthetic imaging process that renders the synthetic model representation on a digital image plane, based on the camera projection model, lighting model, and scene representation.
- Establishing differentiability for the image rendering, enabling the evaluation of the sensitivity of the synthetic image rendering with respect to the model parameters.

The chapter is structured as follows: initially, an analytical camera model for the

endoscope is presented, followed by the introduction of the digital model representation and imaging process. In Section 2.1, the analytical camera model is devised to precisely depict the kinematic intricacies of a rigid cystoscope, as well as its distortion effects. Section 2.2 introduces a suitable synthetic model representation for the surgical site and expounds on the rendering process, which is founded on the analytical camera model. This rendering process generates a photorealistic digital image that takes spatial occlusion into account. In Section 2.3, the non-differentiable operations present in a state-of-the-art rendering process are identified and reformulated for differentiability.

2.1 Analytical Camera Model

The analytical camera model presented in this study delineates the projection of a 3D scene onto a 2D image plane and the kinematic behavior of the camera during this process. This representation is crucial for rendering 3D scenes onto a 2D image plane, imitating the image capture of a real-world camera observation.

This research presents a general approach for reconstructing the camera pose solely based on intraoperative image observations. Therefore, the camera projection model is introduced in a general form that accommodates the camera perspective determined by the system's rotation and translation. However, a specific endoscope kinematic is indispensable when external pose measurements are incorporated for validation or integration into the reconstruction framework. As such, this study takes into account the technical implementation of a rigid cystoscope without restricting the generality intended for all methods presented in this work. Additionally, the proposed endoscope model integrates constraints on both the endoscope kinematics and the projection to respect the given degree. The procedure can be applied to any rigid fiber optics camera system. It is important to note that while the specific camera kinematic is crucial for sensor fusion and validation purposes, the purely image-based reconstruction does not rely on the particular camera kinematics.

2.1.1 Pinhole Camera Model

The pinhole camera model is one of the most elementary and fundamental camera projection models. It models the projection of a three-dimensional point onto a two-dimensional image plane through a pinhole, as illustrated in Figure 2.2. Within the pinhole camera model, the camera's optical projection center is located at the origin of the world coordinate system, while the z-axis signifies the camera's viewing direction. The retinal plane $\mathcal{R}$ is defined as the shifted xy-plane at $z = 1$. Additionally, the intersection of the retinal plane and the optical axis specifies the focal length f, which is given herein as $f = 1$.

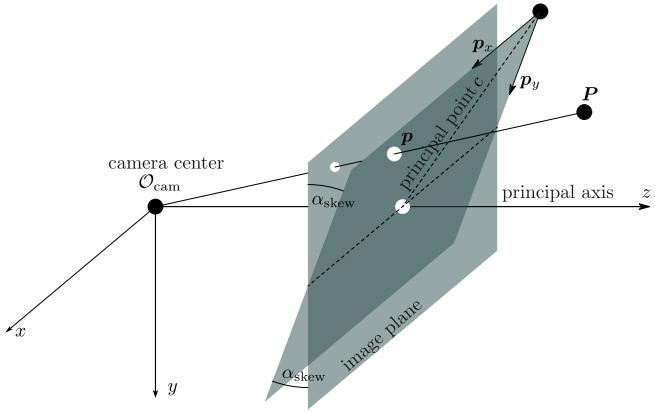


Figure 2.2: Point projection on the image space following the pinhole camera model.

However, in reality, the optical center is located between the spatial point and the image plane, leading to a reflected image. To simplify the model and eliminate reflection effects, the image plane is virtually placed in front of the optical center. In accordance with the ray theorem, the projection process of a 3D point $\boldsymbol{P} = \left[P_x, P_y, P_z\right]^{\mathrm{T}} \in \mathbb{R}^3$ onto the image plane is modeled by

$$x_{\mathcal{R}} = \frac{P_x}{P_z} \qquad y_{\mathcal{R}} = \frac{P_y}{P_z} \ , \tag{2.1a}$$

where $x_{\mathcal{R}}$ and $y_{\mathcal{R}}$ are the coordinates of the retinal imaging plane. The nonlinear mapping prescription can be transformed into a homogeneous representation

$$\begin{bmatrix} x_{\mathcal{R}} \\ y_{\mathcal{R}} \\ 1 \end{bmatrix} \propto \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \tag{2.1b}$$

that allows for linearly proportional model relations.

2.1.2 Intrinsic Camera Projection

The pinhole camera model, as represented by (2.1b), requires modification to describe a more general camera projection. In an real world camera system, the distance between the projection center and the retinal plane, known as the focal length f, is typically not equal to one. As a result, the image coordinates in (2.1b) must be scaled according to the focal length f. Additionally, the projection coordinates

on the retinal plane do not directly correspond to the final image coordinates captured by a digital camera image. The captured image depends on camera-specific parameters, including the pixel size Δp, skew angle α_{skew}, and position of the sensor relative to the optical axis, specified by the principal point c. Accounting for the camera-specific parameters, the transformation from retinal coordinates to image coordinates results in

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \propto \begin{bmatrix} \frac{f}{\Delta p_x} & \frac{f}{\Delta p_y}\tan(\alpha_{\text{skew}}) & c_x \\ 0 & \frac{f}{\Delta p_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\mathcal{R}} \\ y_{\mathcal{R}} \\ 1 \end{bmatrix} \tag{2.2a}$$

with x and y that specify the projection point $\boldsymbol{p} = [x\ y]^{\mathrm{T}}$ of $\boldsymbol{P}$ in the respective continuous image coordinate system. However, the parameterization is sufficiently specified in the projection model

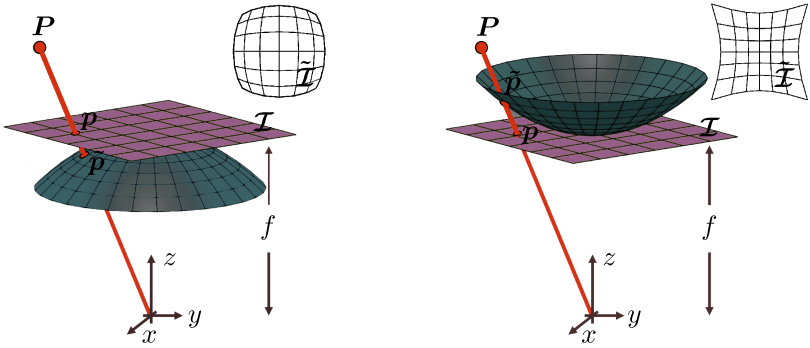
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \propto \underbrace{\begin{bmatrix} f_x & s_{\text{skew}} & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{K}} \begin{bmatrix} x_{\mathcal{R}} \\ y_{\mathcal{R}} \\ 1 \end{bmatrix} . \tag{2.2b}$$

The projection model described in this study uses a simplified notation for the calibration matrix $\boldsymbol{K}$. This matrix includes the measured focal lengths f_x and f_y, which are determined separately for the x and y directions, respectively, along with a skew factor s_{skew} that accounts for non-rectangular pixel sizes. By combining these camera parameters into a single calibration matrix $\boldsymbol{K}$ a more compact representation of the camera projection is provided. Cameras with fixed optics are consistent across all images captured using the same optics. However, modern endoscopes with automatic zoom and focus control may require ongoing adjustments to the calibration matrix's parameterization to accommodate changes in focal length.

2.1.3 Image Distortion

The optical lens typically introduces image distortions that increase from the optical center outward, which are particularly noticeable at the edges of the optical lens, as illustrated in Figure 2.3. This ultimately produces misleading depth information in the 3D reconstruction. The shorter the focal length and the wider the lens angle, the greater the distortion. In endoscopes, wide-angle lenses are typically deployed to enlarge the field of view. As a result, the distortion caused in endoscopic images is particularly pronounced and must be corrected accordingly [105, 149].

A model is necessary to compensate for the distortion effects. By exploiting an analytical model that takes into account the distortion parameters, it becomes possible to transform a distorted image into its undistorted equivalent. If not rectified, lens distortion can disrupt the pattern alignment across different observation perspectives. This can result in the misidentification of depth information as image



(a) Barrel distortion. **(b)** Pincushion distortion.

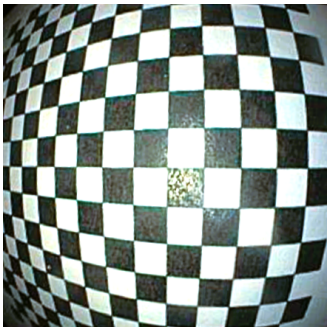
Figure 2.3: Around the principal point, the image surface behaves like a pinhole (violet plane). However, due to optical aberrations and imperfections in the camera lens, the captured image can become distorted. This distorted image is represented by the surface $\tilde{\boldsymbol{I}}$, while the equivalent undistorted image is represented by the surface $\boldsymbol{I}$. Confer with [48].

distortion could potentially be misclassified as deformation effects after completing the pattern matching.

More specifically, lens distortion can generally be classified into two categories: radial and tangential distortion. Radial distortion arises when light rays deflect from the image center, becoming more prominent as the rays refract further from the lens center. Tangential distortion occurs when the lens and the image plane are not parallel, causing rectangular geometries to appear as trapezoids in the image plane.

Radial Distortion: Radial distortion is a form of lens distortion that occurs when the pixels in an image are radially displaced from their accurate positions. This is caused by the fact that the image is formed on a curved surface, rather than a plane, due to the refraction of light as it passes through the lens. The degree of refraction depends on the lens material's index of refraction and the lens shape. Smaller lenses, characterized by shorter focal lengths and stronger curvatures, typically generate more radial distortion. This is because the light that passes through these lenses bends more sharply, leading to a greater degree of image curvature and distortion that must be corrected when projecting the image onto a flat surface.

There are multiple approaches to model radial distortion. A common empirical



(a) Observed endoscopic image with distortion.

(b) Undistorted equivalent of (a) after calibration.

Figure 2.4: Images showing the effect of distortion correction on an endoscopic image. The image in (a) shows radial distortion, which is corrected in (b) through the use of camera calibration.

model is given by

$$\begin{bmatrix}\tilde{x}\\ \tilde{y}\end{bmatrix} = \begin{bmatrix}x(1+k_1r^2+k_2r^4+k_3r^6)\\ y(1+k_1r^2+k_2r^4+k_3r^6)\end{bmatrix} \tag{2.3a}$$

where $r^2 = x^2+y^2$ relates the distorted image coordinates $\tilde{\boldsymbol{p}} = [\tilde{x},\tilde{y}]^{\mathrm{T}}$ to the undistored image coordinates $\boldsymbol{p} = [x,y]^{\mathrm{T}}$ with the distortion parameters $k_{\{1,2,3\}}$ [70, 144]. For lenses producing less complex distortions, the third parameter (k_3) may be omitted.

Tangential Distortion : Tangential distortion occurs when the lens and the image plane are not perfectly aligned parallel to each other. This can cause rectangles to appear as trapezoids in the captured image. The tangential distortion is typically modeled through the empirical model formulation

$$\begin{bmatrix}\tilde{x}\\ \tilde{y}\end{bmatrix} = \begin{bmatrix}x+2t_1xy+t_2\left(r^2+2x^2\right)\\ y+t_1\left(r^2+2y^2\right)+2t_2xy\end{bmatrix} \tag{2.3b}$$

holding the tangential model parameters $t_{1,2}$ to relate the distorted pixel coordinates $\tilde{x}$ and $\tilde{y}$ to the respective distortion free coordinates x and y [70, 144].

2.1.4 Image Undistortion

The distortion model presented in (2.3) can be used to recalculate an undistorted image based on a given parameterization. The nonlinear models for radial distortion (2.3a) and tangential distortion (2.3b) are combined in the general notation

f_{dist}, with the corresponding inverse mapping denoted by f_{undist} for ease of notation. Figure 2.4a displays an endoscopic image capture of a checkerboard in its original distorted state, while Figure 2.4b shows the corresponding undistorted image in its 'unwarped' form.

The camera parameters are commonly identified by utilizing predefined patterns such as regular checkerboard patterns. The calibration process involves capturing images of the pattern from different perspectives. To accurately model the camera's free movement, the extrinsic camera parameters such as the rotation and translation must be integrated into the overall projection model [94].

2.1.5 Integrating Intrinsic and Extrinsic Parameters for a General Camera Projection Model

In the projection model (2.2b), the camera perspective is fixed to a pre-defined coordinate system. A transformation is necessary, to accommodate arbitrary camera perspectives. This transformation is expressed as

$$^{W}\boldsymbol{P} = {}^{\mathrm{W}}\boldsymbol{R}_{\mathrm{C}}{}^{C}\boldsymbol{P} + {}^{W}\boldsymbol{T} \,, \tag{2.4}$$

where $^{W}\boldsymbol{P} \in \mathbb{R}^3$ is the projection point in the world coordinate system $\{W\}$. This point is transformed to its representation in the camera coordinate system $\{C\}$. The transformation involves an orthogonal rotation matrix $^{\mathrm{W}}\boldsymbol{R}_{\mathrm{C}} \in \mathbb{R}^{3\times3}$ (simplified as $\boldsymbol{R}$), which transforms the direction of the vector without affecting its length, and a corresponding translation vector $\boldsymbol{T} \in \mathbb{R}^3$. These components„ termed the extrinsic camera parameters, facilitate the modeling of any camera perspective. Alterations in the camera's perspective can be perceived as the corresponding reverse displacement of the scene in the camera coordinate system $\{C\}$.

The inverse mapping of (2.4), with $^{\mathrm{C}}\boldsymbol{R}_{\mathrm{W}} = {}^{\mathrm{W}}\boldsymbol{R}_{\mathrm{C}}^{\mathrm{T}}$, captures the image flow as the camera transitions from world to camera coordinates. The collection of all sub-models (2.1b),(2.2b), and(2.4) facilitates synthesis of both intrinsic and extrinsic parameters, forming a comprehensive analytical camera projection model

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \propto \underbrace{\begin{bmatrix} f_x & s_{\text{skew}} & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}^{\mathrm{T}} & -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{T} \\ \boldsymbol{0}_3^{\mathrm{T}} & 1 \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \boldsymbol{P}_x \\ \boldsymbol{P}_y \\ \boldsymbol{P}_z \\ 1 \end{bmatrix}, \tag{2.5}$$

where the resulting matrix $\mathbf{M} \in \mathbb{R}^{3\times4}$ is referred to as the *camera projection matrix* [94]. The projection matrix (2.5) provides the general description for projecting model information onto the image plane depending on the camera parameters. For the scope of this work, a parameterization of the external camera parameters consisting of rotation and translation is needed.

2.1.6 Parameterization of the Extrinsic Motion Parameters

The rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ has nine degrees of freedom which are not independent and must be condensed into a unique parameterization that is suitable for the formulation of a well-conditioned optimization problem. A standard parameterization is given by the Euler angles, which represent the orientation of the camera as three separate rotational angles around the x, y, and z axes. However, Euler angles have limitations, such as the potential for singularities, which can result in discontinuities in the camera's orientation [57, 142]. An alternative approach is to employ quaternions coordinates, a mathematical representation of rotations in 4D space that are more compact than rotation matrices but may be more challenging to work with mathematically and are less intuitive to interpret than Euler angles [20, 57, 74, 75, 142].

In this work, the rotation matrix $\boldsymbol{R}$ is parameterized by the so-called Rodrigues vectors as discussed in [21, 74]. The Rodrigues vector is a representation of 3D rotations defined by

$$\mathbf{r} = \frac{\vartheta}{2}\mathbf{i} \tag{2.6a}$$

where $\mathbf{r}$ is the Rodrigues vector, $\mathbf{i}$ is the unit vector representing the axis of rotation, and ϑ is the rotation angle. The angle of rotation ϑ is given by the norm $\vartheta = |\mathbf{r}|$. The Rodrigues vector is related to the axis of rotation and the rotation angle and can be used to parametrize a rotation matrix $\boldsymbol{R}$. The rotation matrix is calculated by

$$\boldsymbol{R} = \boldsymbol{E} + \frac{\sin|\mathbf{r}|}{|\mathbf{r}|}[\mathbf{r}]_\times + \frac{1-\cos|\mathbf{r}|}{|\mathbf{r}|^2}\mathbf{r}\mathbf{r}^\intercal, \tag{2.6b}$$

where $\boldsymbol{E}$ is the identity matrix, and $[\mathbf{r}]_\times$ denotes the skew-symmetric matrix corresponding to the cross product with $\mathbf{r}$.

Compared to Euler angles, Rodrigues vectors are less susceptible to singularities and error accumulation, and are easier to interpret compared to quaternion coordinates [38]. The Rodrigues parameterization is used in this work because it allows for the continuous optimization of rotations; small changes in the Rodrigues vector correspond to small changes in the rotation, making it useful in optimization problems where incremental changes to the rotation are necessary. Additionally, Rodrigues vectors are more numerically stable than Euler angles because the quantified deviation between Rodrigues vectors is more uniformly related to the corresponding change of pose. This property is essential for any parameter update of the form (A.1), where a uniformly distributed parameter update over the entire vector supports a more stable convergence behavior during optimization [42]. Finally, to provide a comprehensive notation for the threedimensional pose representation that includes both the translation vector $\boldsymbol{T}$ and the rotational degrees expressed by the Rodrigues vector $\mathbf{r}$, this work defines the pose representation $\boldsymbol{\phi} = [\boldsymbol{T}, \mathbf{r}]$.

2.1.7 Calibration of the Camera Model

Calibration is typically accomplished using a calibration object with predefined pattern, such as a checkerboard pattern, captured from multiple viewpoints. The corresponding point correspondences are used to determine the camera parameters. The camera model projects spatial points onto the image plane according to (2.5). However, since this projection is not invertible, the inverse mapping is ambiguous. To unambiguously determine a point in space, it is necessary to capture the same point from at least two intersecting perspectives based on the triangulation principle, which is illustrated in Figure 2.2. The triangulation principle exploits corresponding point matches $\mathbf{m}_{\tilde{\boldsymbol{p}}_A \leftrightarrow \tilde{\boldsymbol{p}}_B}$ in two or more images, and then intersecting the rays passing through those points in the respective camera spaces A and B. The intersection of these rays results in the location of the corresponding 3D point $\boldsymbol{P}$.

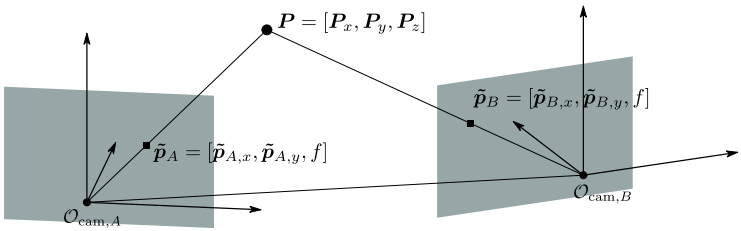


Figure 2.5: Triangulation principle; based on the known projection matrices, a 3D point $\boldsymbol{P}$ can be reconstructed from two matching image points $\tilde{\boldsymbol{p}}_A$ and $\tilde{\boldsymbol{p}}_B$. Confer with [94].

Thus, based on the triangulation principle, the intrinsic and extrinsic parameters are determined by minimizing the discrepancy

$$ {}^{W}\boldsymbol{R}_A^{-1}\boldsymbol{K}^{-1} f_{\mathrm{undist}}\left(\tilde{\boldsymbol{p}}_A\right) - {}^{W}\boldsymbol{R}_A^{-1}\boldsymbol{T}_{AW} = {}^{W}\boldsymbol{R}_B^{-1}\boldsymbol{K}^{-1} f_{\mathrm{undist}}\left(\tilde{\boldsymbol{p}}_B\right) - {}^{W}\boldsymbol{R}_B^{-1}\boldsymbol{T}_{BW} \ , \tag{2.7} $$

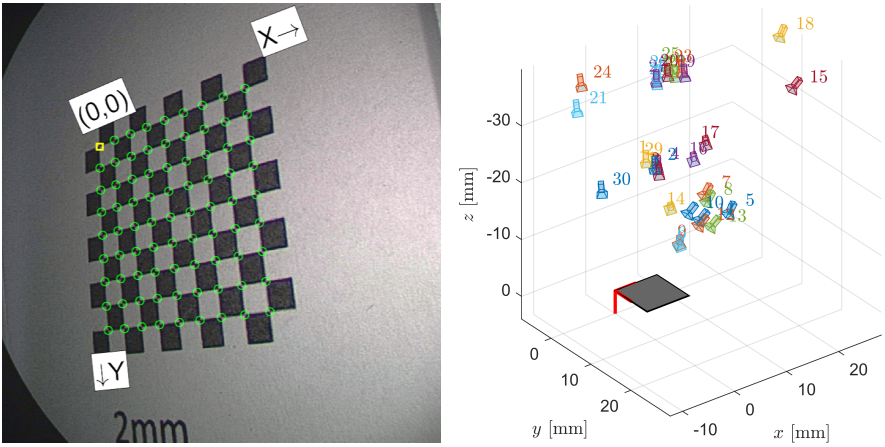
where the distortion is included in the back transformation, allowing for the unique determination of the intrinsic parameters. The extrinsic parameters can be reconstructed up to an arbitrary scaling factor. A sufficient number of matching node pairs $\tilde{\boldsymbol{p}}_A \leftrightarrow \tilde{\boldsymbol{p}}_B$ must be established through redundancy in the formulation of (2.7) to ensure accuracy and robustness.

To enhance the identification of point correspondences, a calibration object, like the checkerboard pattern presented in Figure 2.6a, is employed. The geometric arrangement of the checkerboard pattern and the utilization of basic edge detection techniques enable the accurate detection of landmarks with high precision down to the sub-pixel level. Moreover, the predetermined dimensions of the checkerboard provide the scale ratio without any ambiguity.

To determine the extrinsic parameters for a batch of images, pixel matches are used, while the intrinsic parameters may be determined using the entire dataset to increase accuracy and account for outliers. As an illustration, the calibrated intrinsic camera parameters of the Storz endoscope [131] (shown in Figure 1.1b) are presented below:

- Focal length in mm : $f_x = 18.24\,\text{mm}$, $f_y = 18.26\,\text{mm}$
- Principal point in pixels: $c_x = 639.5$ pixels, $c_y = 359.5$ pixels
- Skew coefficient (unitless): $s_{\text{skew}} = 0.01$
- Distortion coefficients (unitless): $k_1 = -0.4$, $k_2 = 0.21$, $k_3 = 0.00$, $t_1 = 0.0$, $t_2 = 0.00$

The alignment of checkerboard pairs is commonly used to solve the regression problem for calibrating and determining camera parameters, as depicted in the general formulation (2.7). Although various studies, such as [29, 79, 94, 149] have presented alternative self-calibration techniques and more comprehensive reconstruction procedures, they are outside the scope of this work and will not be further discussed.



(a) Checkerboard with detected edges. **(b)** Reconstructed poses.

Figure 2.6: For camera calibration, 30 independent checkerboard images were captured from distinct perspectives. The reconstructed poses obtained from the calibration process are shown in (b).

2.1.8 Kinematic Model of a Rigid Cystoscope

The image-based perspective reconstruction, as specified in (2.7), offers a versatile and distinct solution without compromising generality or uniqueness. To accurately tie the reconstructed camera perspective to a specific position on the endoscope,

it is imperative to take into account the endoscope's physical kinematics, like its angled tip and given degrees of freedom. Relying solely on a simplified virtual camera position within the observed image can be insufficient. This consideration becomes especially critical in this work when validating the reconstruction with external measurement markers on the endoscope. In this section, a kinematic model is derived for a rigid cystoscope to demonstrate the modeling of rigid endoscopes and manipulator systems. Therefore, the technical aspects of a rigid cystoscope are briefly reviewed first.

2.1.8.1 Technical Description of the Cystoscope

In medical endoscopy, the technical implementation of image transmission relies on either fiber-optic or purely optical-electronic concepts, depending on the clinical needs and anatomical conditions. For procedures that require minimal access trauma, small and flexible endoscopes are typically used. However, larger endoscope diameters may be required for wider lenses and working channels to ensure optimal image quality and surgical intervention space for certain applications.

In cystoscopy procedures, both rigid and flexible cystoscopes are available; however, the rigid form is more commonly utilized in clinical practice [105]. The rigid endoscope consists of a telescope that determines the area of invasion and depth of penetration, while an external light source illuminates the surrounding tissue. Reflected light from the tissue is focused at the tip of the endoscope and transmitted through a series of rod lenses to the eyepiece, offering a clear view of the internal tissue. Additionally, a connected camera digitizes the image seen through the eyepiece, allowing the surgeon to operate with the appropriate resolution on a screen.

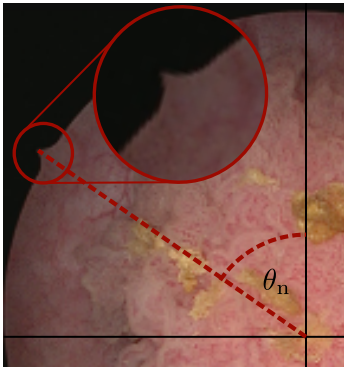
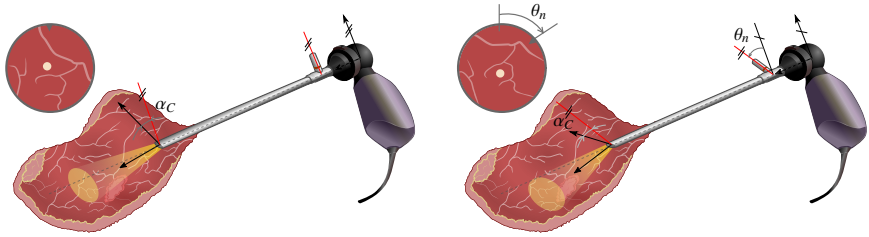


Figure 2.7: The notch at the distal end of the endoscope is visible as a small triangle in the captured image. The angle of the notch can be extracted from the imaged notch relative to the corresponding image coordinate system.

A schematic cross-section of a rigid cystoscope is shown in Figure 1.1b. To increase the field of view, rigid cystoscopes often have observation lenses at various angles, some of which can be adjusted or fixed in angle, as shown in Figure 1.1a. This provides the surgeon with an additional field of view beyond the range of motion limited by anatomy. For example, a 120° tip may even allow the surgeon to see in reverse directions. Accordingly, the cystoscope's field of view is controlled by the rotation of the telescope.

To facilitate orientation, the camera sensor typically has a degree of rotational freedom with the eyepiece. This allows for adjusting the field of view by rotating the

telescope without rotating the camera image itself. The visual outcome for a change in the telescope rotation of the cystoscope kinematics is illustrated in Figure 2.8. Moreover, a small notch is included in the cystoscope image output to monitor the relative rotation of the camera and cystoscope, as highlighted in Figure 2.7 and visible in all preceding endoscopic image captures, which were shown in Figure 1.3a, Figure 1.3b, and Figure 1.4b. The notch indicates the angle between the camera and cystoscope and enables the surgeon to determine the orientation of the tip, helping to avoid collisions with the surrounding anatomy.



(a) Cystoscope has no relative rotation to the camera, $\theta_n = 0$.

(b) Cystoscope is rotated $\theta_n > 0$ relative to the camera.

Figure 2.8: Relative rotation between the cystoscope shaft, characterized by tip angle α_C, and the physical camera. The camera's resulting image is displayed on the top left, where a notch indicates the relative rotation θ_n between the shaft and the camera. This notch corresponds to the angular displacement of the light post from the vertical axis of the attached camera, causing the object in view to appear increasingly off-center due to the tip angle.

2.1.8.2 Kinematic Model of a Rigid Cystoscope

The Denavit-Hartenberg convention [22] is a prevalent method of modeling kinematic chains in robotics, which also applies to the given endoscope kinematics. This model description is dedicated to the kinematics of a rigid cystoscope, while also considering the rotational degree of the fiber optic system in the projection model. Although the model outlined in this description is tailored to the cystoscope depicted in Figure 1.1, the model approach is generally applicable to any other fiber optic endoscope and telemanipulator system. To introduce the required body-fixed coordinate systems, the rotational coupling of the camera to the eyepiece is referred to as the frame $\{P\}$ at the center of the pivot, accounting for the relative telescope rotation. The distal end of the cystoscope, referred to as the center of view (TCV), is referenced in the coordinate system $\{E\}$. As shown in Figure 2.9, the frame $\{E\}$ has an angular offset

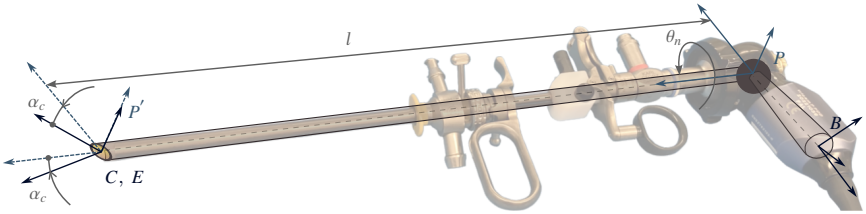


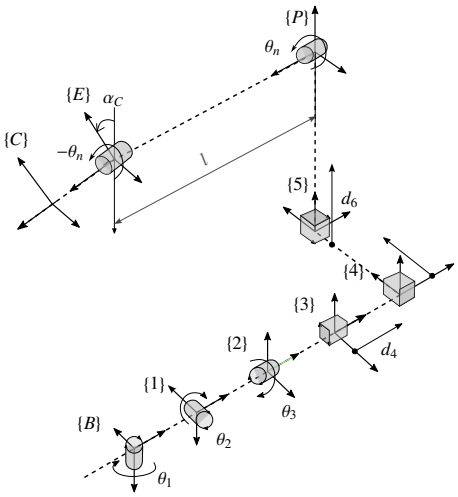
Figure 2.9: Kinematic model of the endoscope.

of α_c relative to the frame $\{P\}$, which corresponds to the angled direction of view of the cystoscope. Initially, frames $\{E\}$ and $\{C\}$ share a common origin, with O_E equivalent to O_C. Since frame $\{E\}$ is attached to the physical end of the cystoscope, the x and y axes of $\{E\}$ rotate around the long axis of the cystoscope according to the rotation of the cystoscope shaft. This rotation is specified by the notch angle θ_n and corresponds to the angular displacement of the notch seen in the respective video image. The resulting overall transformation from the observed virtual camera perspective $\{V\}$ to the reference point of the endoscope, located in the handle of the camera body without loss of generality, follows a serial transformation of joints and connections according to the DH convention [22]. This yields a transformation ${}^{\mathrm{C}}\boldsymbol{T}_{\mathrm{B}}(\boldsymbol{\chi})$ from $\{B\}$ to $\{C\}$ concerning the joint degrees of freedom $\boldsymbol{\chi}$ of $\{V\}$, where frame $\{B\}$ characterizes the cystoscope's hand-piece. The full kinematic diagram and corresponding DH parameters of the joints

$$\boldsymbol{\chi} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & d_4 & d_5 & d_6 & \theta_n \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^7 \tag{2.8}$$

are listed and shown in Figure 2.10 following the DH convention. In this kinematic model, the camera's end position in $\{B\}$ maintains a distinct position and orientation relative to the pivot point in $\{P\}$. Since the image origin $\{E\}$ coincides with the camera position, O_E is tantamount to O_C. When the cystoscope undergoes θ_n rotation, the camera's optical axis $\{C\}$ inevitably aligns with the cystoscope's viewing direction, thus establishing $\{E\} \equiv \{C\}$.

However, due to the rotating mechanism of the telescope, there is a disconnect between the image-based pose reconstruction and the actual physical camera, with the handpiece serving as the target reference point. Since the physical camera is located opposite $\{P\}$ and $\{C\}$ represents the projection of the physical camera's orientation, the orientation of the frame $\{C\}$ remains unchanged during shaft rotation θ_n. This implies that $x_E \neq x_C$ and $y_E \neq y_C$, leading to a mismatch between the image-reconstructed perspective and the physical TCP orientation at $\{E\}$. To account for this mismatch, the axes x_C and y_C are constrained using θ_n in combination with the image observation. Therefore, a compensation joint $E \rightarrow C$ is defined so that a rotation around $P \rightarrow E$ can be corrected. This ensures that the optical axis z_C remains collinear with z_E while accurately aligning the image-based camera axis with



${}^{C}\boldsymbol{T}_{B}(\boldsymbol{\chi})$

i	α	a	d	θ
1	90°	0	0	$\theta_1 + 90°$
2	−90°	0	0	$\theta_2 - 90°$
3	0°	0	0	θ_3
4	90°	0	d_4	0°
5	90°	0	d_5	90°
6 (P)	90°	0	d_6	−90°
E	α_c	0	l	θ_n
C	0°	0	0	$-\theta_n$

Figure 2.10: The kinematic diagram illustrates the Denavit-Hartenberg (DH) parameters of the sensor setup.

the physical shaft alignment. Except for the shaft rotation angle θ_n, all lengths and angles are fixed and can be determined using the system specifications of the cystoscope. The shaft rotation angle θ_n can be easily obtained from the notch in the transmitted image, which rotates with the telescope housing and indicates the relative rotation between the shaft and the camera, as outlined in Section 2.1.8.1 and seen in Figure 2.9.

2.2 Intraoperative Model Representation and Synthetic Image Projection

The rendering process generates a digital image representation of a synthetic 3D scene. This goes beyond the analytical image projection as it involves the interaction between object, light, and material properties in the scene to synthesize the final image data. Essentially, the rendering is a synthetic camera projection of a virtual model, capturing a digital image.

The process of rendering involves generating a digital representation of a synthetic 3D scene. Unlike the analytical image projection, rendering requires calculating the interactions among objects, lights, and materials within the scene to produce the final image data.

In the following, a synthetic model representation suitable for intraoperative use is introduced in Section 2.2.1. Building upon the synthetic model representation and the analytical projection model, the conventional rendering process is presented in Section 2.2.5. In this context, all operations within the rendering process are classified for differentiability.

2.2.1 Scene Representation

Rendering algorithms and model representation are tightly linked. The choice of a suitable model representation depends on the complexity of the application and the desired model resolution. Model representations frequently employed in rendering algorithms encompass point clouds, voxels, meshes, and implicit neural network representations. Furthermore, the rendering process's efficiency and precision are depended on the chosen model representation. As such, the model representation selection must meet the requirements of both the complexity of the application and the desired resolution of the model.

Point clouds depict the model geometry through an assembly of points in 3D space. Nonetheless, extracting pertinent information, such as surfaces and normals, can be computationally demanding and ambiguous, particularly in the case of sparse point clouds. In contrast, voxel models represent interconnected volumes using unit cubes. This makes them straightforward to handle, but they are constrained to small scenes with low resolution due to the considerable number of parameters needed. Furthermore, there are implicit model representations that rely on a neural network formulation to delineate an object's geometric constraints. Typically, the network models the distance from a point to the object's surface, implicitly characterizing the surface as the set of points $\boldsymbol{P}$ fulfilling the condition of the network prediction $\mathcal{N}(\boldsymbol{P}) = 0$. Although this method necessitates a constant, resolution-independent parameter space, it remains unfeasible for real-world applications, given the requirement for accurate ground truth data to train the network.[50]

Meshes represent surfaces using a limited number of parameters, thereby facilitating efficient optimization despite high geometric complexity. They prove especially advantageous when focusing on surface information exclusively and can be utilized with a high level of detail and a minimized parameter space. Owing to these attributes and the emphasis on cystoscopic interventions, where a spherical mesh can accurately represent the bladder surface, the mesh model is deemed the optimal choice for model representation in this work. The combination of efficient parameterization and high surface resolution in mesh models without neglecting intricate details, while the minimized parameter space reduces the computational demands.

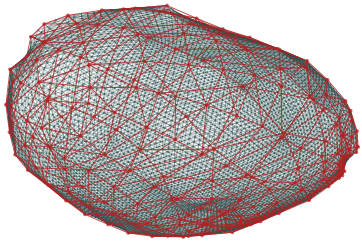


Figure 2.11: Triangular mesh at two different resolutions, shown in red with a number of $|\mathbf{V}| = 212$ vertices and in grey with a number of $|\mathbf{V}| = 10242$ vertices, used to parameterize the geometry of a human urinary bladder.

2.2.2 Digital Model Representation

Polygon meshes are one of the prevalent model representation in computer graphics for depicting 3D surface information. A polygon mesh comprises vertices, edges, and faces that delineate the shape of the model. Within this structure, a face can be a triangle, square, or any other n-dimensional convex polygon. Moreover, various operations can be performed on a mesh, including smoothing, subdivision, and logical operations. This work exclusively focuses on triangular meshes, which are particularly well-suited for the rendering process and constructing a differentiable framework due to their unique properties. Triangles are both planar and convex, enabling straightforward computational relationships when intersecting with lines or normals. The weighted centroid of adjacent vertices also allows for efficient and tractable interpolation of properties across the entire spanned space. Furthermore, as most modern graphics units and algorithms are optimized for triangle meshes, they facilitate accelerated rendering and more efficient mesh processing.

The architecture of a triangular mesh is composed of a set of vertices, denoted as $\mathbf{V}$, wherein each vertex $\mathbf{V}_j$ is expressed as a 3D coordinate $\mathbf{V}_j \in \mathbb{R}^3$. Additionally, a list of faces $\mathbf{F}_j$ is defined by their respective indices $j \in \mathbb{N}$, which correspond to the vertices present in the vertex list. An example of such a structure is illustrated in Figure 2.11, which depicts the surface geometry of a urinary bladder parameterized by a triangular mesh. The model representation's resolution is dictated by the number of mesh vertices.

In polygon meshes, the viewing direction is determined by organizing the vertices in a mathematically counterclockwise orientation. This specific orientation is responsible for establishing the front and back facets of a face, subsequently influencing the lighting and shading effects during the rendering process.

2.2.3 Parameterization of the Surface Geometry

The resolution and geometry of a polygon mesh depend on the vertices $\mathbf{V}$ and faces $\mathbf{F}$, which determine the connectivity between the vertices in the mesh. A face $\mathbf{F}_j$ is defined by its corresponding vertex coordinates $\left[\mathbf{V}_{j,\mathrm{a}}, \mathbf{V}_{j,\mathrm{b}}, \mathbf{V}_{j,\mathrm{c}}\right]$ and determines the surface plane of the mesh. The coordinates of a point $\boldsymbol{P}$ within the face $\mathbf{F}_j$ are determined by

$$\boldsymbol{P}_j = \mathbf{V}_{j,\mathrm{a}} + \mathtt{u}\left(\mathbf{V}_{j,\mathrm{b}} - \mathbf{V}_{j,\mathrm{a}}\right) + \mathtt{v}\left(\mathbf{V}_{j,\mathrm{c}} - \mathbf{V}_{j,\mathrm{a}}\right), \tag{2.9}$$

which can be interpreted as the weighted center of gravity of the respective face plane. The respective weighting of the vertices is given by the barycentric coordinates $\mathtt{u} \in \mathbb{R}$ and $\mathtt{v} \in \mathbb{R}$. Constraining the coordinate space according to

$$0 \leq \mathtt{u}, \mathtt{v} \leq 1, \quad \text{and} \quad \mathtt{u} + \mathtt{v} = 1, \tag{2.10}$$

reduces the spanned plane (2.9) to the set of coordinates within the area covered by the face $\mathbf{F}_j$.

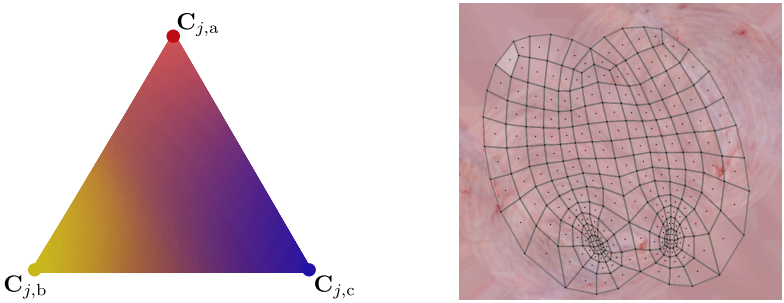
2.2.4 Texture Model

The faces and barycentric weighting of face coordinates in a mesh model define all points on the surface. In addition to geometric shape, surface appearance is crucial to model representation, as it involves color and light interaction. These characteristics, such as roughness, metallicity, specularity, opacity, and other surface effects, are represented in a feature space and must be assigned to the mesh surface using an appropriate texture representation. Three common texture representations exist; vertex texture mapping, `uv`-texture mapping, and atlas texture mapping. The following gives a brief overview of each method, along with its advantages and disadvantages in the context of this work.

Vertex Texture: The most elementary texture representation is provided by vertex texture representation. In this approach, each vertex in the mesh is assigned a feature vector space $\mathbf{C}$ containing texture properties such as surface color and other material properties. Given a point $\boldsymbol{P}_j$ on a face $\mathbf{F}_j$, the corresponding surface feature $\mathbf{C}_j$ is determined by continuous interpolation

$$\mathbf{C}_j = \mathbf{C}_{j,\mathrm{a}} + \mathtt{u}\left(\mathbf{C}_{j,\mathrm{b}} - \mathbf{C}_{j,\mathrm{a}}\right) + \mathtt{v}\left(\mathbf{C}_{j,\mathrm{c}} - \mathbf{C}_{j,\mathrm{a}}\right) \tag{2.11}$$

based on the `uv`-value parameterization. This interpolation relies on the respective feature vectors assigned to the vertices of the face, as illustrated in Figure 2.12a. While intuitive and straightforward to implement, vertex texture is not well-suited for intricate and complex textures on meshes with large face areas.



(a) Vertex texture. **(b)** Texture uv-map.

Figure 2.12: Panel (a) illustrates the color distribution on a triangular mesh face based on it vertex features, while panel (b) showcases the texture uv-map in conjunction with the mesh skeleton.

UV Texture: The uv-texture map is a widely used texture model for rendering. This mapping technique involves projecting a two-dimensional map onto a three-dimensional mesh surface, or vice versa, which entails mapping mesh information onto a two-dimensional feature map. Instead of embedding the texture in the three-dimensional geometry space, this approach maps the feature space in a separate image space. By interpolating between discrete pixel values texture mapping provides continuous feature allocation and surface coloring based on the triangular query and corresponding coordinates.

Figure 2.12b illustrates an example of a uv-texture map applied to synthetic bladder geometry. Assigning fixed two-dimensional coordinates to each vertex in the mesh is a crucial step in the mapping process. Various methods exist for initializing these coordinate assignments. It's important to note that the texture map itself is just a memory location, and its interpretation is not always straightforward. The initial assignment of coordinates to the feature map is not unique, and neighboring faces may not have consecutive references.

Atlas Texture Mapping: The atlas texture model is similar to the uv-texture model and relies on reference-based feature mapping. However, in atlas texture mapping, each face of the mesh has its own two-dimensional texture atlas, providing unambiguous initial reference alignment of the texture attributes. Furthermore, this flexible representation allows for different resolutions in the mesh and even to easily manipulate the resolutions and texture at any time without re-initializing the whole texture map. Intraoperative data analysis benefits particularly from the dynamic increase of resolution in the examined areas. This approach allows for a high level of resolution in the examined areas without adding unnecessary complexity to the overall model.

Both the uv-mapping and atlas texture mapping models enable arbitrarily high resolution and detailed coloring at any geometry resolution in the mesh, making them state-of-the-art in computer rendering processes of technical or architectural models.

2.2.5 Synthetic Image Projection Following the State-of-the-Art

The specific steps involved in rendering, such as modeling, texturing, lighting, animation, and post-processing, can vary depending on the software, hardware, and requirements of the application. In this study, image rendering is limited to the rendering process specific to a mesh as a synthetic scene representation. The established camera projection model serves as the analytical foundation for subsequent rendering. Furthermore, the rendering process systematically examines the model representation to uniformly cover the entire discrete image space. This involves identifying visible surface information and determining which aspects are obscured by other mesh faces from the given camera perspective.

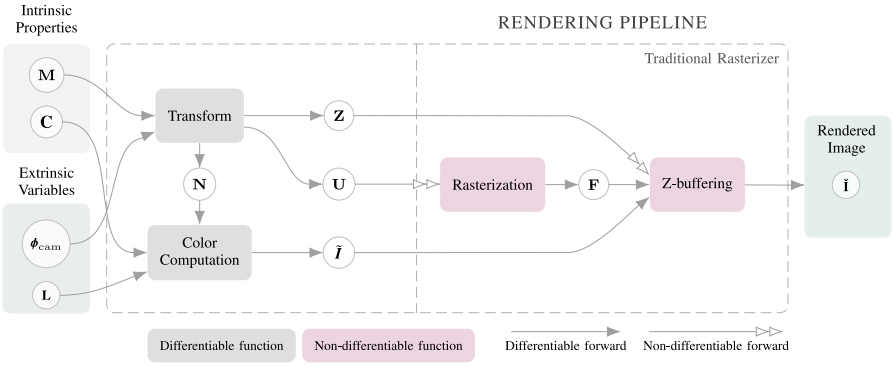


Figure 2.13: Schematic illustration of primary sub-operations in state-of-the-art rendering pipeline.

Figure 2.13 provides a schematic overview of the rendering process examined in this study, which aligns with widely-used real-time rendering pipelines such as OpenGL [96]. The process involves a transformation scheme that relates the camera and scene coordinate systems, enabling the analytical computation of normals $\mathbf{N}$, depth information $\mathbf{z}$, and barycentric coordinates $\mathbf{U}$ of the mesh $\mathbf{M}$ associated with the camera perspective $\boldsymbol{\phi}_{\text{cam}}$. Subsequently, the color information for the image plane $\tilde{\boldsymbol{I}}$ is determined based on the texture model $\mathbf{C}$ and the lighting conditions $\mathbf{L}$. For non-transparent objects, it is essential to map the closest surface information of the model representation from the camera perspective onto the image plane and to

assess the opacity and visibility of all included model points. Finally, these individual operations are interconnected to produce a discretized image information in the rasterized image matrix $\check{\mathbf{I}}$.

Decomposed, the rendering process follows a series of steps and preparatory procedures presented in the respective subsections, including:

- Defining the digital image space in Section 2.2.5.1 to set the requirements for the rendering pipeline and tailor it to the desired output data.
- Performing the analytical projection in Section 2.2.5.2 to generate image information for each pixel in the digital image based on the camera perspective and the model. This involves calculating the position and properties of objects in the scene as they would appear from the perspective of the camera.
- Addressing the occlusion problem in Section 2.2.5.3 to ensure that the appropriate surface information is determined for each pixel in the digital image.
- Implementing the appropriate lighting model in Section 2.2.6 based on the camera perspective to add realism and depth to the scene. The lighting model calculates the intensity of the surface color based on the lighting conditions.
- Presenting the combination of the texture and lighting information in Section 2.2.7 using the method of color interpolation known as Phong shading.

2.2.5.1 Digital Image Space

A digital image is composed of a grid of pixels, which are individual points arranged in a rectangular matrix $\mathbf{I}$ to represent the image intensities. Grayscale pixels have intensity values ranging from black to white in the integer interval [0, 255], while color pixels require three channels to encode the corresponding color information through the additive mixture of three primary colors. It is important to note that color is subjective and the mixture of colors does not adhere to the physical laws of optics. Instead, it mirrors the physiological response of the human eye to light [122]. Various color systems employ distinct primary colors as the foundation for their respective color spaces, which can be transformed into one another via coordinate representations. The most prevalent color system relies on the primary colors red, green, and blue (RGB), wherein any color can be expressed as a three-dimensional vector encoding the weights of the respective primary colors. In computer graphics, discrete pixel intensities are generally scaled to a continuous range between zero and one, facilitating continuous image processing in either color or grayscale. In this study, the continuous representation of pixel intensities and the RGB color space are utilized. The integer indices $\mathtt{h} \in \mathbb{H} \subset \mathbb{N}$ and $\mathtt{w} \in \mathbb{W} \subset \mathbb{N}$ specify the discrete location of pixels within the image matrix. The corresponding Euclidean coordinate in the image plane is denoted by $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$.

2.2.5.2 Analytical Ray Tracing

The rendering of the digital image requires the computation of pixel intensity by aggregating the relevant surface information for each pixel, based on the camera's perspective and light conditions. This process, known as ray tracing, involves passing a ray from the camera's perspective through each pixel of the image plane and checking for the closest intersection with the scene model [91]. This technique can be extended by tracing the rays further past the first hit with the respective surface. Ray tracing is also known as 'backward tracing' because the rays are traced from the camera's viewpoint instead of starting from the light sources. However, this approach enables the simulation of physical phenomena such as absorption and reflection, resulting in highly realistic and detailed shading effects.

Following the ray tracing principle, the intersection of a chosen pixel $(\mathtt{h}, \mathtt{w})$ with an arbitrarily selected face $\mathbf{F}_j$ is determined by passing a ray from the projection center $\boldsymbol{O}_{\mathrm{cam}}$ of the camera through the Euclidean pixel position $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$ to the respective face $\mathbf{F}_j$. It is essential to note that $(\mathtt{h}, \mathtt{w})$ works as an index of the image matrix and does not represent any Euclidean information. In contrast, $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$ describes the respective Euclidean distances according to the given origin on the image plane according to (2.5). Including the focal length f, the principal point c, and the camera orientation in the world coordinate system, the pixel position in the three-dimensional space is given by

$$\mathbf{P}^{\mathtt{h},\mathtt{w}} = \begin{bmatrix} c_x + \boldsymbol{p}_x^{\mathtt{h},\mathtt{w}} \\ c_y + \boldsymbol{p}_y^{\mathtt{h},\mathtt{w}} \\ f \end{bmatrix} . \tag{2.12a}$$

Thus, the corresponding ray passing through the camera origin and the given pixel can be expressed by

$$\mathbf{R}^{\mathtt{h},\mathtt{w}} = \boldsymbol{O}_{\mathrm{cam}} + \mu_{\mathrm{ray}} \left(\boldsymbol{p}^{\mathtt{h},\mathtt{w}} - \boldsymbol{O}_{\mathrm{cam}} \right) , \tag{2.12b}$$

where $\mu_{\mathrm{ray}} \in \mathbb{R}$ is a scalar parameter, which parameterizes a point $\mathbf{P}^{\mathtt{h},\mathtt{w}}$ on the ray $\mathbf{R}^{\mathtt{h},\mathtt{w}}$. The intersection point $\boldsymbol{P}_j^{\mathtt{h},\mathtt{w}}$ of the ray (2.12b) with the given face plane (2.9) of $\mathbf{F}_j$ is found by equating (2.12b) with spanned face plane (2.9), and solving for the intersection point. The face plane is parameterized by μ_{ray} and the $\mathtt{uv}$ coefficients of the respective face $\mathbf{F}_j$. If the parameterization satisfies the barycentric constraint (2.10), the intersection point lies within the face boundaries. If not, the intersection point lies outside the face. In a brute-force approach to aggregating pixel values, the intersection point for each given face must be calculated and evaluated to determine whether it lies inside or outside the given face boundaries. This results in a point cloud $\boldsymbol{P}_{j\in\|\mathbf{F}\|}^{\mathtt{h},\mathtt{w}}$ of $\|\mathbf{F}\|$ individual number of intersections for the pixel $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$.

However, for the final aggregation of pixel values, at least for opaque bodies, only the closest surface in the field of view is relevant, which requires additional depth information to be queried. If an admissible intersection point $\mathbf{P}^{\mathtt{h},\mathtt{w}}$ is determined

within the face triangle under consideration, the pixel intensity $\mathbf{I}^{\mathtt{h},\mathtt{w}}$ that corresponds to the surface intersection $\mathbf{P}^{\mathtt{h},\mathtt{w}}$, is temporarily assigned to the pixel's feature space. The corresponding texture $\mathbf{I}^{\mathtt{h},\mathtt{w}}$ is used after the occlusion evaluation to determine the pixel intensity in conjunction with the lighting conditions.

2.2.5.3 Occlusion Check

Several occlusion detection algorithms exist in the literature, which can be classified into two broad categories: methods that analytically determine the visibility in the object space and methods that exploit the image space and sparsely perform for each pixel an opacity check. In modern graphics systems, either the Z-buffer algorithm [132], which offers high computational efficiency, or the ray tracing algorithm [32], which generates photorealistic, high-resolution renderings, are commonly employed.

Like ray tracing, Z-buffering checks for each pixel which elements of a scene need to be drawn or hidden from the perspective of the viewer. In the Z-buffering process, both the face IDs and the distances of intersections are recorded for all pixels. Therefore, the Z-buffer entries are initialized with an infinite value to represent the background, and then iterated for all pixels over all faces. The entries are then updated whenever smaller distances along the z-axis are observed [132]. Figure 2.14 illustrates the Z-buffer algorithm, and the matrix update can be formally defined by

$$Z_{\text{buffer}}[\mathtt{h},\mathtt{w}] = \begin{cases} Z_{\text{buffer}}[\mathtt{h},\mathtt{w}], & \text{if } Z_{\text{buffer}}[\mathtt{h},\mathtt{w}] \leq \left[\boldsymbol{P}_j^{\mathtt{h},\mathtt{w}}\right]_{C,z} \\ \left[\boldsymbol{P}_j^{\mathtt{h},\mathtt{w}}\right]_{C,z}, & \text{if } Z_{\text{buffer}}[\mathtt{h},\mathtt{w}] > \left[\boldsymbol{P}_j^{\mathtt{h},\mathtt{w}}\right]_{C,z} \end{cases} . \tag{2.13}$$

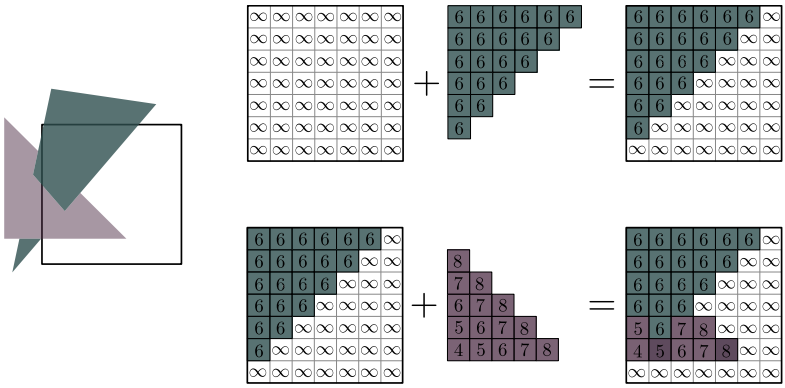


Figure 2.14: Illustration of the Z-buffer principle, where the distances to individual faces are checked for each pixel to determine the closest face information in view for the corresponding pixel. Confer with [147].

The determined depth value for pixel $(\mathtt{h}, \mathtt{w})$ is stored in the Z-buffer entry $Z_{\text{buffer}}[\mathtt{h}, \mathtt{w}]$, while the specific depth values $\left[\boldsymbol{P}_j^{\mathtt{h},\mathtt{w}}\right]_{C,z}$ is determined by the z coordinate of the intersection point $\boldsymbol{P}_{j,C}^{\mathtt{h},\mathtt{w}}$ defined in the camera coordinate system $\{C\}$. Moreover, $Z_{\text{buffer}}[\mathtt{h}, \mathtt{w}] \mapsto \left[\mathbf{P}_{\text{cview}}^{\mathtt{h},\mathtt{w}}, \mathbf{I}_{\text{cview}}^{\mathtt{h},\mathtt{w}}\right]$ encodes the visibility of the respective intersection point $\mathbf{P}_{\text{cview}}^{\mathtt{h},\mathtt{w}}$ and feature information $\mathbf{I}_{\text{cview}}^{\mathtt{h},\mathtt{w}}$ based on the calculated face intersections as defined in 2.12. By parallelizing the distance calculations on modern graphic processing units (GPU), the Z-buffering process becomes computationally highly efficient.

2.2.6 Lighting and Shading Model

The lighting conditions must be considered when determining the final pixel intensity and color rendering. An illumination model in computer graphics simulates the light behavior during the rendering process, ultimately determining the brightness and pixel color based on viewing direction, angle of light incidence, material properties, and light source. In this work, the Phong illumination model [93] is employed due to its analytical approach and inherent ability to support differentiation and real-time calculations. The Phong lighting model is a local illumination model that provides an analytical description of lighting effects and is suitable for smooth surfaces with specular lighting effects. It is based purely on empirical evidence and does not have a direct physical foundation. However, it is computationally efficient and can produce photo-realistic results. Other local illumination models, such as the Schlick [112] or Kook Torrance [19] model, are physically based and adhere to the energy conservation law, meaning that they do not reflect more photons than they irradiate. These models, however, require more computational resources.

The Phong reflection model consists of ambient, ideal diffuse, and ideal specular reflection. Each of these components contributes to the overall reflection intensity of a surface under illumination from a light source. Each of the sub-models is empirically formulated as follows:

Ambient: The ambient reflection

$$I_{\text{amb}} = \kappa_{\text{amb}} I_{\text{int}} \tag{2.14a}$$

is exclusively dependent on the intensity of the ambient light I_{int} and the empiric material-dependent reflection coefficient κ_{amb}. Thus, the ambient reflection is independent of the camera perspective and the angle of incidence of the light source.

Diffuse: Lambert's law describes the relation between the intensity of diffuse reflection from a surface and the angle between the incident light beam and the surface normal. Specifically, the intensity of diffuse reflection is inversely proportional to

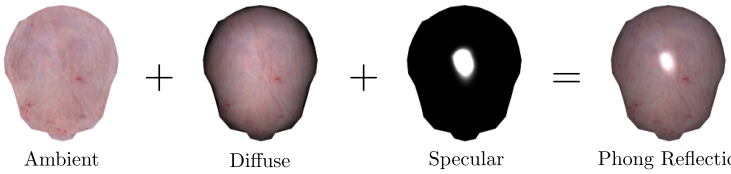


Figure 2.15: Image rendering using Phong illumination submodels, featuring ambient, diffuse, and specular components and the resulting superimposed image.

the angle between the incident light beam and the surface normal. When the incident light beam is perpendicular to the surface, the diffuse reflection is at its maximum intensity. As the angle between the incident light beam and the surface normal increases, the intensity of the diffuse reflection decreases. This relationship is mathematically represented by

$$I_{\text{diff}} = \kappa_{\text{diff}} I_{\text{int}} \cos(\alpha_{\text{dirt}}) \quad = \kappa_{\text{diff}} I_{\text{int}} \left(\boldsymbol{\phi}_{\text{light}} \mathbf{N}_j \right) , \tag{2.14b}$$

where κ_{diff} is the material-dependent reflection coefficient, I_{int} is the intensity of the incoming light source, and α_{dirt} is the angle between the orientation $\boldsymbol{\phi}_{\text{light}}$ of the incident light beam and the surface normal $\mathbf{N}_j$. It is worth noting that Lambert's law applies only to diffuse reflection, which is characterized by a diffuse, uniform scattering of light in all directions. It does not apply to specular reflection, which is characterized by a concentrated reflection in a single direction.

Specular: The specular reflection component involves the dependence on the camera perspective. It is defined by the angle α_{spec} between the observer's viewing direction $\mathbf{R}_{\text{view}}$ and the direction of an ideal reflection $\mathbf{R}_{\text{ref}}$. Thus, the ideal reflection is determined by

$$I_{\text{spec}} = \kappa_{\text{spec}} I_{\text{int}} \cos(\alpha_{\text{spec}})^{\kappa_{\text{ref,spec}}} \quad = \kappa_{\text{spec}} I_{\text{int}} \left(\mathbf{R}_{\text{view}} \mathbf{R}_{\text{ref}} \right)^{\kappa_{\text{ref,spec}}} . \tag{2.14c}$$

The material exponent $\kappa_{\text{ref,spec}}$ characterizes the surface condition. As $\kappa_{\text{ref,spec}}$ approaches zero, the surface becomes coarser, while as $\kappa_{\text{ref,spec}}$ approaches ∞, the model behaves like a perfect mirror. In addition, the reflection is influenced by the empirical, material-dependent reflection coefficient κ_{spec}.

2.2.7 Image Aggregation

Finally, the color intensity of each pixel $\boldsymbol{I}^{\text{hw}}$ can be determined based on the visible texture information and the light conditions established by the Phong model (2.14). The visual texture information is assigned to the feature intersection evaluated for visibility through Z-buffering.

The resulting color appearance of the pixel information is determined by the Phong shading interpolation model [93] through

$$\mathbf{I}^{\mathrm{h,w}} = (I^{\mathrm{h,w}}_{\mathrm{amb}} + I^{\mathrm{h,w}}_{\mathrm{diff}})\mathbf{I}^{\mathrm{h,w}}_{\mathrm{cview}} + I^{\mathrm{h,w}}_{\mathrm{spec}} \ . \tag{2.15}$$

Thus, the aggregated color information $\mathbf{I}^{\mathrm{h,w}}$ is calculated by the product of the respectively visible texture information $\mathbf{I}^{\mathrm{h,w}}_{\mathrm{cview}}$ with the ambient (2.14a) and diffuse reflectance (2.14b) in addition to the specular reflectance (2.14c). Figure 2.15 shows the resulting image aggregations of a synthetic bladder model, which is obtained by applying different light intensities, reflectance properties, and roughness values in the Phong shading process. It is important to note that the Phong shading model is distinct from the Phong illumination model; the former defines only the interpolation rule for using ambient, diffuse, and specular light properties, while the latter is a model for simulating light behavior in the rendering process.

2.3 Pixel Intensity-based Differentiable Rendering

Building upon the state-of-the-art rendering concept discussed in Chapter 2, the remaining challenge is to ensure differentiability of the rendering process such that a gradient-based scene reconstruction of the form given in (3.1) becomes feasible. The sub-operations involved in the rendering process are depicted in Figure 2.16, with each operation categorized as either differentiable or non-differentiable. Discontinuities stemming from image rasterization and Z-buffering are illustrated in Figure 2.17.

In more detail, the conventional Z-buffering procedure discretely determines the pixels in an image based on the closest face in view. Discontinuous shifts may occur in the aggregated pixel information when the camera or a captured mesh face moves in depth, as another surface approaches the camera image plane along the z-direction. Likewise, a shift in the xy-plane can also result in a discontinuity in pixel aggregation. These discontinuities are intrinsically linked to changes in aggregated pixel information. As a result, if facial information is not represented in the image plane, it will be absent in any image-based loss definition. This absence constrains the availability of information necessary for back-propagation in gradient-based reconstruction, making distant or occluded vertices problematic, as they cannot contribute to the proposed image-based reconstruction formulation. This issue can quickly cause the reconstruction to become trapped in a local minimum for those vertices. Although some progress has been made in the literature on formulating differentiable rendering processes, many of the proposed solutions still struggle with limited effectiveness [50].

The problem of establishing a differentiable rendering process raises the following specific challenges and research questions:

- How can a discrete occlusion check be transformed into a continuous mapping to ensure differentiability, especially when depth relations in the model change?

- In what way should pixel information be aggregated to prevent discontinuities

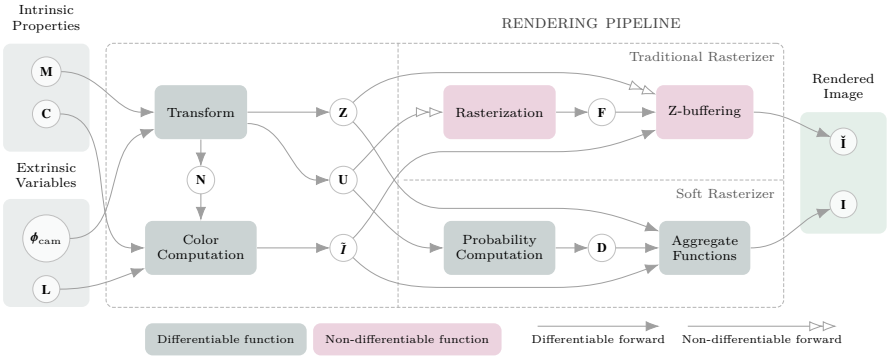


Figure 2.16: Structure and categorization of non-differentiable sub-operations within a modern rendering pipeline. The flow chart showcases the solutions presented in this work to organize these concepts in their respective locations throughout the overall pipeline process. Confer with [65].

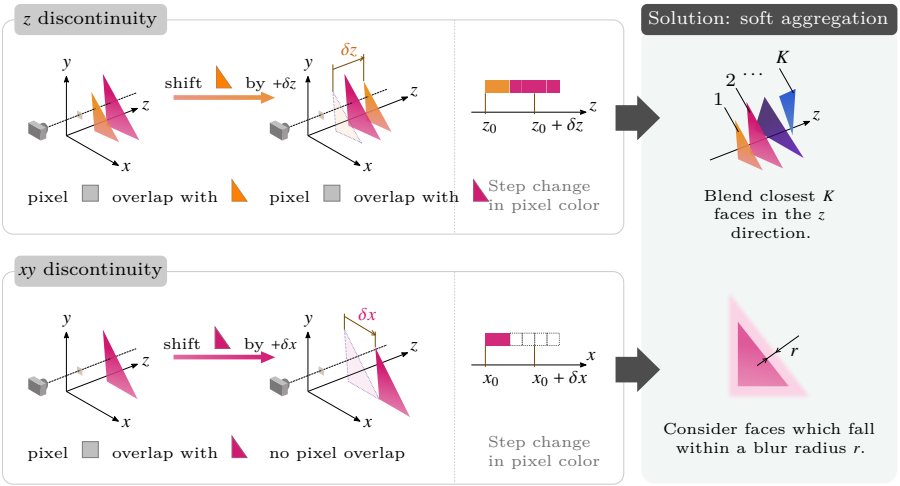


Figure 2.17: Discontinuities in a conventional pipeline hinder differentiation. Soft aggregation, based on the weighted average of multiple intermediate rendering information, provides a continuous distribution and enables differentiation. Confer with [101].

in the mapped information when there is a shift in the camera or the model relative to the *xy* camera coordinates?
- How can rendering mappings be fine-tuned and tailored for gradient-based reconstruction and intraoperative use, leveraging silhouette data to provide more reliable information when texture features are indistinct?

To overcome these challenges and attain true differentiability in the rendering process, a sensitivity-based approach is adopted. This is depicted in Figure 2.16 through the design of soft rasterization, which makes the differentiation of the rendered image matrix feasible. This novel formulation originally presented in [69] and further elaborated in [65, 101] represents a significant advancement in the field.

In the following Section 2.3.1, the gradient required for model reconstruction is analytically decomposed to constrain and classify the differentiability for each suboperation. A sensitivity distribution is then designed in Section 2.3.2 to map information to the corresponding pixels reliably. In Section 2.3.3.1, image aggregation is used to present solutions to both the *xy* and *z* discontinuities through weighted averaging based on the sensitivity distribution. Further, in Section 2.3.3.2, silhouette rendering is proposed to provide robust aggregation information by representing the confidence if object or non-object information is aggregated in the respective pixel intensity. This approach enables the supervision of size relations without relying on texture information, making it particularly useful for intraoperative reconstructions.

2.3.1 Specifying the Discontinuity Given in the Rendering Pipeline

The analytical decomposition of the state-of-the-art rendering process, as discussed in Section 2.2, provides a systematic approach for the identification and resolution of discontinuities in the rendering process. The gradients with respect to the mesh geometry, given by its vertex positions $\mathbf{V}$, is formulated through

$$\frac{\partial \mathbf{I}}{\partial \mathbf{V}} = \frac{\partial \mathbf{I}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \mathbf{V}} + \frac{\partial \mathbf{I}}{\partial z} \frac{\partial z}{\partial \mathbf{V}} + \frac{\partial \mathbf{I}}{\partial \mathbf{N}} \frac{\partial \mathbf{N}}{\partial \mathbf{V}} \ . \tag{2.16a}$$

Similarly, the gradient with respect to the camera pose $\boldsymbol{\phi}_{\text{cam}}$ is expressed by

$$\frac{\partial \mathbf{I}}{\partial \boldsymbol{\phi}_{\text{cam}}} = \frac{\partial \mathbf{I}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\phi}_{\text{cam}}} + \frac{\partial \mathbf{I}}{\partial z} \frac{\partial z}{\partial \boldsymbol{\phi}_{\text{cam}}} + \frac{\partial \mathbf{I}}{\partial \mathbf{N}} \frac{\partial \mathbf{N}}{\partial \boldsymbol{\phi}_{\text{cam}}} \ . \tag{2.16b}$$

Given the partial derivatives in (2.16a) and (2.16b), the respective gradients

$$\frac{\partial \boldsymbol{p}}{\partial \mathbf{V}}, \ \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\phi}_{\text{cam}}}, \ \frac{\partial z}{\partial \mathbf{V}}, \ \frac{\partial z}{\partial \boldsymbol{\phi}_{\text{cam}}}, \ \frac{\partial \mathbf{I}}{\partial \mathbf{N}}, \ \frac{\partial \mathbf{N}}{\partial \mathbf{V}}, \ \frac{\partial \mathbf{N}}{\partial \boldsymbol{\phi}_{\text{cam}}} \tag{2.16c}$$

are analytically determinable by differentiating the projection matrix (2.5), and the illumination model (2.15). However, as previously mentioned, the gradients $\frac{\partial \mathbf{I}}{\partial \boldsymbol{p}}$ and $\frac{\partial \mathbf{I}}{\partial z}$ cannot be obtained in a conventional rendering pipeline due to the discontinuities

caused by the discretization of the image plane and the occlusion check, as given in this work through the Z-buffering process.

The necessity of establishing a differentiable rendering process raises the following specific challenges: How can the rasterization and depth-related visibility check be reformulated to make the aggregated image differentiable with respect to the model parameters required in (2.16a) and (2.16b). Therefore, two main modifications from [65] are adopted. First, a sensitivity distribution $\mathcal{D}$ is included, which makes the rendered pixel intensities $\mathbf{I}$ differentiable with respect to the considered model parameters. Thereby, the influence of a particular face $\mathbf{F}_j$ on a pixel $(\mathtt{h}, \mathtt{w})$ is represented by a sensitivity distribution $\mathcal{D}_j^{\mathtt{h},\mathtt{w}}$, which ultimately factorizes the gradient $\frac{\partial \mathbf{I}}{\partial \boldsymbol{p}} = \frac{\partial \mathbf{I}}{\partial \mathcal{D}} \frac{\partial \mathcal{D}}{\partial \boldsymbol{p}}$ and enables the differentiation of $\frac{\partial \mathbf{I}}{\partial \boldsymbol{p}}$.

Second, the depth-dependent discontinuity $\frac{\partial \mathbf{I}}{\partial z}$ is addressed by continuously merging the available intersection points $\boldsymbol{P}_j^{\mathtt{h},\mathtt{w}}$ for a given pixel by involving a depth-dependent weighting of neighboring information. In this process, the Z-buffering visibility check is replaced. Thus, the final pixel color is aggregated based on the color information C of the corresponding face intersections using the respective sensitivity distribution $\mathcal{D}$ and depth-dependent weighting design $\mathcal{W}$. This process ensures that all spatial information is incorporated into the aggregated pixel, enabling the respective gradient to reflect sensitivities about the scene parameters. To address the challenges of rendering, the sensitivity distribution is designed in the following section.

2.3.2 Sensitivity Distribution for Resolving Discontinuity

The distribution design proposed in [65] aims to assess the confidence of the projected face information $\mathcal{F}_j$ of $\mathbf{F}_j$ in predicting the vulnerability of pixel $(\mathtt{h}, \mathtt{w})$ to discontinuities. Therefore, the considered distribution $\mathcal{D}$ relies on the distance $d_j^{\mathtt{h},\mathtt{w}}$ of the pixel $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$ to its closest boundary to the projected face information $\mathcal{F}_j$ of the arbitrarily chosen face $\mathbf{F}_j$ on the image plane as illustrated in Figure 2.18. Based on that, the sensitivity distribution $\mathcal{D}_j^{\mathtt{h},\mathtt{w}}$ is defined as

$$\mathcal{D}_j^{\mathtt{h},\mathtt{w}} = \operatorname{sigmoid}\left(\delta_j^{\mathtt{h},\mathtt{w}} \frac{\left(d_j^{\mathtt{h},\mathtt{w}}\right)^2}{\sigma_{\text{diff}}}\right) \quad j \in [1, |\mathbf{F}|] \ . \tag{2.17}$$

The design parameter σ_{diff} controls the slope of the distribution, with steeper slopes indicating a stronger influence of the face $\mathbf{F}_j$ on the pixel $(\mathtt{h}, \mathtt{w})$. Moreover, regarding the distance measurement, it is fundamental to the design of the distribution (2.17) to specify whether the distance of a pixel location $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$ to the projected face $\mathcal{F}_j$ falls inside or outside the the respective boundaries $\mathcal{F}_j^{\blacktriangle}$, which is determined by the sign indicator

$$\delta_j^{\mathtt{h},\mathtt{w}} = \left\{+1, \text{ if } \boldsymbol{p}^{\mathtt{h},\mathtt{w}} \in \mathcal{F}_j^{\blacktriangle} \;\middle|\; -1, \text{ otherwise}\right\}, \tag{2.18}$$

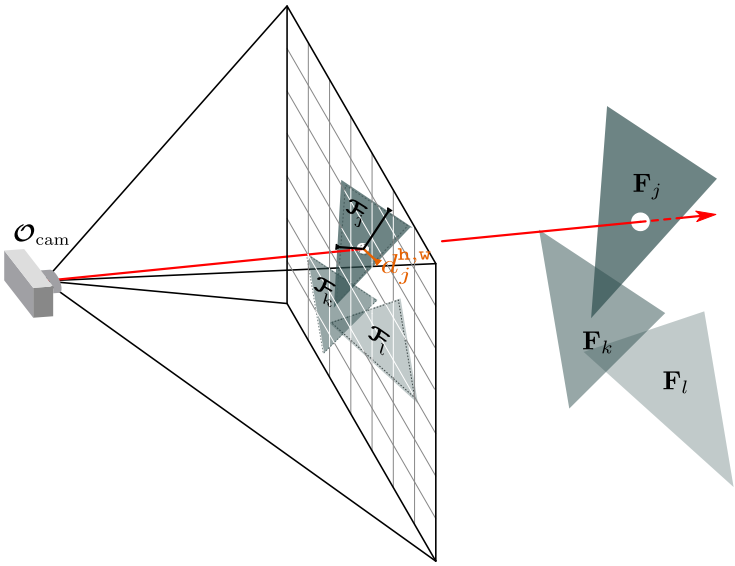


Figure 2.18: The face projection $\mathcal{F}_{j,k,l}$ represents the projection of the faces $\mathbf{F}_{j,k,l}$ onto the image plane. For a given pixel $(\mathtt{h}, \mathtt{w})$, the closest distance to the projected face boundary $\mathcal{F}_j$ is indicated by $d_j^{\mathtt{h},\mathtt{w}}$.

that takes on the value of +1 if the pixel position $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$ is within the projection boundaries $\mathcal{F}_j^{\blacktriangle}$, and −1 otherwise. The designed sensitivity distribution provides a mathematical formulation for predicting the discontinuity vulnerability as the distance $d_j^{\mathtt{h},\mathtt{w}}$ determines the contribution of face $\mathbf{F}_j$ to the pixel aggregation of pixel $(\mathtt{h}, \mathtt{w})$.

The sigmoid function in 2.17 limits the effect of a face on a pixel to a smooth range between zero and one. For instance, pixels located far within the projected face plane $\mathcal{F}_j$ are strongly affected, while pixels located outside the projected area are only slightly affected. However, by incorporating boundary regions in a continuous manner as designed in 2.17, the information about the face is maintained in the resulting pixel and can be traced in the gradient representation. The distribution $\mathcal{D}_j^{\mathtt{h},\mathtt{w}}$ is aimed to quantify the xy-discontinuity problem and to control the impact of the face information on pixel $(\mathtt{h}, \mathtt{w})$ in the aggregation process. The design of $\mathcal{D}_j^{\mathtt{h},\mathtt{w}}$ is not intended to have statistical significance but instead serves as a tool for predicting the discontinuity vulnerability of pixels based on their distance to the closest boundary of the projected face information. Thus, this allows for many variations on how $\mathcal{D}_j^{\mathtt{h},\mathtt{w}}$ is defined, as long as it is continuous with respect to changes of its distance to the projected face boundaries.

The spatial course of the distribution $\mathcal{D}$ is qualitatively depicted in Figure 2.19, in

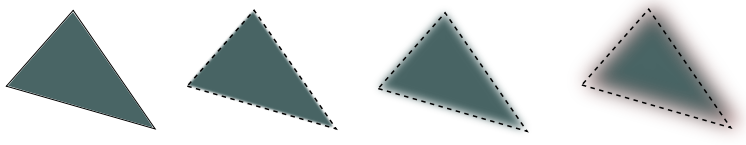


Figure 2.19: Visualization of the distribution design $\mathcal{D}_j^{\text{h,w}}$ for a face projection $\mathcal{F}_j$, where the color intensity at a specific location indicates the weight of the distribution for each pixel location.

dependence of the control parameter σ_{diff}. This parameter regulates the sharpness of the mapping $\mathcal{D}$, such that when $\sigma_{\text{diff}} \to 0$, the distribution closely follows the geometry of the face projection, resulting in a conventional rasterization on the image grid.

Finally, the distribution design (2.17) facilitates the differentiation of $\frac{\partial \mathcal{D}}{\partial \boldsymbol{p}}$. To demonstrate this, the matrix

$$\boldsymbol{U}_j = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}_{\mathcal{F}_j} . \tag{2.19}$$

is used to represent the Euclidean positions of the projected vertices of $\mathbf{V}$ of face $\mathbf{F}_j$ on the image projection $\mathcal{F}_j$ plane. Thus, any point specified by the barycentric coordinates $\boldsymbol{t}_j^{\text{h,w}} \in \mathbb{R}^3$ can be related to its corresponding coordinates $\boldsymbol{p}$ in Euclidean space following the mapping

$$\boldsymbol{p}^{\text{h,w}} = \boldsymbol{U}_j \boldsymbol{t}_j^{\text{h,w}} . \tag{2.20}$$

The mapping enables a comprehensive examination of the gradient formation of $\frac{\partial \mathcal{D}}{\partial \boldsymbol{p}}$ by converting points specified in the barycentric coordinate space of the face projection $\mathcal{F}_j$ to the corresponding positions in the respective Euclidean image space. In addition, the barycentric coordinates $\boldsymbol{t}_j^{\text{h,w}}$ enable the representation of the point on the face boundary that is closest to a selected pixel $\boldsymbol{p}^{\text{h,w}}$. Thus, the corresponding distribution based on the signed Euclidean distance is specified as

$$\mathcal{D}_j^{\text{h,w}} = \text{sigmoid}\left(\frac{\delta_j^{\text{h,w}}}{\sigma_{\text{diff}}} \| \boldsymbol{U}_j \boldsymbol{t}_j^{\text{h,w}} - \boldsymbol{p}^{\text{h,w}} \|^2 \right) . \tag{2.21}$$

As the sigmoid function is differentiable, the gradient $\frac{\partial \mathcal{D}}{\partial \boldsymbol{p}}$ is given by

$$\frac{\partial \mathcal{D}}{\partial \boldsymbol{p}} = 2 \left(\frac{\delta_j^{\text{h,w}}}{\sigma_{\text{diff}}} \| \boldsymbol{U}_j \boldsymbol{t}_j^{\text{h,w}} - \boldsymbol{p}^{\text{h,w}} \| \right) \left(\boldsymbol{t}_j^{\text{h,w}} \right)^{\text{T}} , \tag{2.22}$$

based on the reformulation of (2.17) through (2.21).

2.3.3 Pixel Aggregation

Based on the designed sensitivity distribution (2.17), the remaining task is to address the z-discontinuity and to aggregate the pixel intensity $\mathbf{I}^{\mathrm{h,w}}$. Therefore, two distinct aggregation functions are presented in the following. The first is used to render image information with realistic color features, while in the second aggregation formulation an adapted silhouette rendering function is presented that serves to differentiate between object and background information.

2.3.3.1 Image Aggregation

As demonstrated in (2.16), the visibility of the closest faces in view primarily determines the final pixel aggregation in the image space. However, the discrete occupation check, as implemented by the Z-buffering approach, can cause a discontinuity issue, as illustrated in Figure 2.17 in the top row. To address this issue, individual image features are weighted for the image feature aggregation to reduce the effect of discontinuity. As such, the relative depths $z_j^{\mathrm{h,w}}$ of $\boldsymbol{p}^{\mathrm{h,w}}$ to the corresponding face intersection $\mathbf{P}^{\mathrm{h,w}}$ are employed for the weight design

$$\boldsymbol{w}_j^{\mathrm{h,w}} = \frac{\mathcal{D}_j^{\mathrm{h,w}} \exp\left(z_j^{\mathrm{h,w}}/\gamma_{\mathrm{diff}}\right)}{\sum_k \mathcal{D}_k^{\mathrm{h,w}} \exp\left(z_k^{\mathrm{h,w}}/\gamma_{\mathrm{diff}}\right) + \exp(\epsilon_{\mathrm{transp}}/\gamma_{\mathrm{diff}})} \ . \tag{2.23}$$

The depth-depend weighting (2.23) allows for a continuous aggregation of the pixel intensities $\mathbf{I}^{\mathrm{hw}}$ based on the weighted average

$$\mathbf{I}^{\mathrm{hw}} = \sum_{j \in \|\mathbf{F}\|} \boldsymbol{w}_j^{\mathrm{h,w}} \mathbf{C}_j^{\mathrm{h,w}} + \boldsymbol{w}_{\mathrm{back}}^{\mathrm{h,w}} \boldsymbol{C}_{\mathrm{back}} \ , \tag{2.24}$$

where the weight $\boldsymbol{w}_{\mathrm{back}}^{\mathrm{h,w}}$ for the background color $\boldsymbol{C}_{\mathrm{back}}$ is inherently determined by the required normalization

$$\sum_{j \in \|\mathbf{F}\|} \boldsymbol{w}_j^{\mathrm{h,w}} + \boldsymbol{w}_{\mathrm{back}}^{\mathrm{h,w}} = 1 \ . \tag{2.25}$$

This formulation is consistent with the image aggregation as proposed in [65]. The parameter $\epsilon_{\mathrm{transp}}$ accounts for the transparency of the object. As $\epsilon_{\mathrm{transp}}$ increases, the effect of the background color $\boldsymbol{C}_{\mathrm{back}}$ becomes stronger for all faces that are considered to be background faces determined by the respective face indices back. Additionally, the sharpness of the aggregation is controlled by γ_{diff}. As the control parameter γ_{diff} increases, the weighting of more distant faces is increasingly attenuated compared to that of closer faces. This is due to the normalization of the weights, which assigns more weight to closer faces and correspondingly less weight to more distant faces. As a result, the aggregation function reproduces the traditional z-buffering process as $\gamma_{\mathrm{diff}} \mapsto 0$ approaches zero.

To summarize, the sensitivity distribution $\mathbf{C}_j^{\mathtt{h},\mathtt{w}}$ enables the weighting $\boldsymbol{w}_j^{\mathtt{h},\mathtt{w}}$ to be differentiable in shifts in the xy direction, while the depth-dependent weighting (2.23) additionally ensures continuity in the z direction. As such, the aggregated image pixel $\mathbf{I}^{\mathtt{h}\mathtt{w}}$ reflects not only the neighboring and hidden face projections, but also preserves their sensitivities in the respective gradient.

For a general notation, the differentiable rendering process can be represented in

$$\mathbf{I} = \mathscr{R}_{\boldsymbol{\phi}_{\mathrm{cam}}}(\mathbf{M}), \tag{2.26}$$

where the synthetic image $\mathbf{I}$ is obtained by rendering the mesh model $\mathbf{M}$ from the camera perspective $\boldsymbol{\phi}_{\mathrm{cam}}$. The differentiability is reflected in the analytical formulations for $\frac{\partial \mathbf{I}}{\partial \mathcal{D}}$ and $\frac{\partial \mathbf{I}}{\partial z}$, which are described in

$$\frac{\partial \mathbf{I}^{\mathtt{h},\mathtt{w}}}{\partial \mathcal{D}_j^{\mathtt{h},\mathtt{w}}} = \frac{\boldsymbol{w}_j^{\mathtt{h},\mathtt{w}}}{\mathcal{D}_j^{\mathtt{h},\mathtt{w}}}\left(\mathbf{C}_j^{\mathtt{h},\mathtt{w}} - \mathbf{I}^{\mathtt{h},\mathtt{w}}\right) \tag{2.27a}$$

$$\frac{\partial \mathbf{I}^{\mathtt{h},\mathtt{w}}}{\partial \boldsymbol{z}_j^{\mathtt{h},\mathtt{w}}} = \frac{\boldsymbol{w}_j^{\mathtt{h},\mathtt{w}}}{\gamma_{\mathrm{diff}}}\left(\mathbf{C}_j^{\mathtt{h},\mathtt{w}} - \mathbf{I}^{\mathtt{h},\mathtt{w}}\right). \tag{2.27b}$$

The computation of the partial derivatives relies on the aggregation and weighting design specified in (2.23) and (2.24). It should be noted that the given image pixel aggregation (2.24) is not unique and can be tailored to suit the needs of the application. The work in [66] also investigates the use of a universal aggregation based on a neural network. Initial results indicate that this approach may slightly improve accuracy for synthetic scenes, albeit at increased computational and efficiency costs.

2.3.3.2 Silhouette Aggregation

The aggregation (2.24) is designed for rendering color intensities while incorporating transparency information. This concept can be similarly extended for aggregating the proposed silhouette information by utilizing the transparency design given by (2.24). However, the concept of a weighted average of various face information is not optimal for the purpose of silhouetting. In addition to the depth dependency reflected in (2.23), it is necessary to determine binary information specifying whether any object or background information maps to the respective pixel $(\mathtt{h},\mathtt{w})$. The probabilistic interpretation of $\mathcal{D}_j^{\mathtt{h},\mathtt{w}}$ suggests the following silhouette aggregation function

$$\mathbf{I}_{\mathrm{sil}}^{\mathtt{h}\mathtt{w}} = 1 - \prod_{j \in \|\mathbf{F}\|}\left(1 - \mathcal{D}_j^{\mathtt{h},\mathtt{w}}\right), \tag{2.28}$$

independent of the object's color and relative depths. The design of the aggregation function in (2.28) offers a probabilistic confidence interpretation of the silhouetted

image pixel $\mathbf{I}_{\text{sil}}^{\mathtt{hw}}$ that at least one face $\mathbf{F}_{j\in[1,|\mathbf{F}|]}$ is being projected onto the pixel $(\mathtt{h},\mathtt{w})$. Respectively, the gradient for the silhouette can be explicitly expressed through

$$\frac{\partial \mathbf{I}_{\text{sil}}^{\mathtt{h},\mathtt{w}}}{\partial \mathcal{D}_j^{\mathtt{h},\mathtt{w}}} = \frac{1-\mathbf{I}_{\text{sil}}^{\mathtt{h},\mathtt{w}}}{1-\mathcal{D}_j^{\mathtt{h},\mathtt{w}}} \ . \tag{2.29}$$

Notably, the silhouette aggregation is independent of the depth values z_j of the faces in the mesh, resulting in a vanishing gradient $\frac{\partial \mathbf{I}_{\text{sil}}^{\mathtt{h},\mathtt{w}}}{\partial z_j^{\mathtt{h},\mathtt{w}}} = 0$ with respect to the depth values. Overall, the silhouette rendering function $\mathscr{R}_{\text{sil}}(\mathbf{M},\boldsymbol{\phi}_{\text{cam}}) \mapsto \mathbf{I}_{\text{sil}}$ calculates the silhouetted image renderings $\mathbf{I}_{\text{sil}}$ of the mesh model $\mathbf{M}$, for the camera perspectives $\boldsymbol{\phi}_{\text{cam}}$, irrespective of the object's color or relative depths.

2.4 Summary & Conclusion

In summary, this chapter has presented the fundamental concepts required to determine a digital image from a synthetic model representation that aligns with state-of-the-art practices. The core aspects discussed include the analytical projection equations and the synthetic rendering process. Specifically, the analytical camera model provides the basis for understanding the relationship between 3D data and projected 2D image data. Additionally, by incorporating the kinematics of the endoscope, it is also feasible to relate image-based camera coordinates to the physical location of the endoscope.

A mesh model was chosen for the synthetic representation of the urinary bladder due to its efficiency and suitability for spherical organ problems. With the mesh representation and analytical camera model, the rendering process was introduced, facilitating the synthetic imaging of mesh model representations. This process establishes the connection between the model representation and a digital real-world image observation. Although most calculations in state-of-the-art real-time rendering pipelines are differentiable, the discrete occlusion check in the Z-buffering procedure and the discrete image rasterization impede gradient calculation for image observation concerning scene parameters.

To address this, the conventional rendering process was transformed into a fully differentiable rendering process. Following the concepts in the works of [65, 101], a probability-based formulation for image information was utilized to aggregate projection data through a weighted average formulation. This approach ensures that pixels accurately reflect changes in the model's geometry, texture, and perspective, allowing the generation of gradients needed for optimization of the proposed gradient-based scene reconstruction. Furthermore, a silhouette aggregation technique was introduced, providing valuable information about pixel confidence concerning object surfaces or backgrounds, thereby expanding loss definition options beyond conventional texture-based image comparisons.

3 Texture and Geometry Reconstruction

The proposed methodology for intraoperative scene reconstruction relies on a gradient-based optimization, enabling the generation of an accurate 3D reconstruction of the surgical site. For this objective, an unconstrained optimization problem

$$\mathbf{M}^{\star}, \hat{\boldsymbol{\phi}}^{\star} = \arg\min_{\hat{\boldsymbol{\phi}}, \mathbf{M}} \sum_{i=t-h}^{t} \mathcal{L}\left(\boldsymbol{\mathcal{I}}_i, \mathscr{R}\left(\mathbf{M}, \hat{\boldsymbol{\phi}}_i\right)\right), \tag{3.1}$$

is formulated, wherein each optimization iteration incorporates a batch of data accumulated from the last h observations up to the present observation at time step t. The goal of this optimization problem is to ascertain the optimal synthetic model representation $\mathbf{M}^{\star}$ and camera perspectives $\hat{\boldsymbol{\phi}}^{\star}$ by minimizing the discrepancy between a batch of observed mages $\boldsymbol{\mathcal{I}}$ and synthetic images produced from the 3D model. The rendering function

$$\mathscr{R}(\mathbf{M}, \boldsymbol{\phi}) = \mathbf{I} \tag{3.2}$$

calculates a batch of synthetic images $\mathbf{I}$ for the batch of camera perspectives $\boldsymbol{\phi}$ and the provided model representation $\mathbf{M}$. The disparity between the observations and the rendering is assessed by the loss function $\mathcal{L}$, which serves as the objective function in the context of the reconstruction formulation. Consequently, devising an effective loss function constitutes the pivotal component of the proposed optimization-based reconstruction approach.

The differentiable formulation of the rendering process enables a gradient-based reconstruction objective at the image level, adhering to the general formulation (3.1). Despite the established differentiability of the rendering process, finding the solution to the general differential rendering reconstruction optimization problem remains a formidable challenge for real-world applications. Furthermore, the intraoperative scenario presents particularly challenging aspects with respect to the deformation problem at hand. For instance, the intraoperative environment restricts the variety of available data, limits the perspectives employed due to the minimally invasive access path, and potentially provides an intraoperative image that lacks clear visual features for reference. Concerning geometry reconstruction, this means that geometry adaptation is ambiguous due to the underdetermined nature of the problem complexity. For example, some parts of the mesh may not be observed from the current

rendering perspective and could potentially shift arbitrarily without significantly impacting the information at the image level. This results in an ill-conditioned problem for geometry adaption where not all geometry parameters of the mesh model are included in the optimization objective, which then potentially causes the optimization process to get stuck in local minima or even become numerically unstable.

Despite there being even more aspects and arising challenges for real-world intraoperative scene reconstruction, the specific research questions formulated in the following are addressed with the objective of developing holistic and methodological evaluation for the reconstruction problem as given by (3.1):

- How can the optimization problem for geometry adaptation be regularized to prevent an ill-posed optimization problem?
- How to regularize the objective function so that scaling observed on the image plane is propagated holistically to the overall model, even for surface information that is not observable in the rendering from the current perspective?
- How can the required degrees of freedom be efficiently incorporated into the optimization formulation to allow for the simultaneous use of texture and geometry with limited complexity in the reconstruction, avoiding an overly complex or computationally expensive reconstruction problem?

The challenges of reconstructing the model's geometry and texture are separately addressed in Section 3.1 and Section 3.2 before their joint reconstruction is explored in Section 3.3. New mesh regularizations are introduced for geometry matching, including the reconstruction of geometry using silhouette information which does not require the explicit use of high-resolution texture information. The scaling of a silhouette to the entire geometry through regularization design is analyzed in this process. Section 3.2 delves into the use of texture information, introducing a mesh subdivision strategy that allows for attaining high texture resolution in the model without unnecessarily increasing mesh complexity. The techniques outlined in this chapter form the foundation for the work's aims and have the potential for practical use, especially with regard to silhouette-based geometry adaptation.

3.1 Geometry Reconstruction

The following geometry reconstruction aligns in its core design with the general reconstruction formulation of (3.1), where the degrees of freedom are limited to vertex positions in this process. The resulting geometry adaption may be hindered by various factors, such as the non-convexity of the optimized loss function and the overall poorly conditioned nature of the geometry-based optimization problems. In addition, the update process of the high-dimensional interdependent geometry parameter space poses various challenges. To stabilize the geometry adaptation process, a simple yet effective parameter transformation is introduced in Section 3.1.1. In Section 3.1.2, several mesh regularization losses are proposed that are tailored

to the specific problem conditions of this work to effectively address the ill-posed optimization problems that may arise during geometry reconstruction.

3.1.1 Geometry Parameter Transformation

To enhance numerical stability during the optimization process, the parameter space is transformed to the difference $\mathbf{\Delta V} = \mathbf{V} - \mathbf{V}_0$ between the initial $\mathbf{V}_0$ and the current vertex positions $\mathbf{V}$, rather than applying the current vertex positions directly as the adjustable parameter set. The core design of the geometry adaptation is then stated as

$$\mathbf{V}^\star = \mathbf{V}_0 + \arg\min_{\mathbf{\Delta V}} \sum_{i=t-h}^{t} \mathcal{L}\Big(\boldsymbol{I}_i, \mathscr{R}\big(\mathbf{M}, \hat{\boldsymbol{\phi}}_i\big)\Big) \,, \tag{3.3}$$

where $\mathbf{\Delta V}$ is the optimizable parameter set in the general image-based optimization objective as introduced in (3.1).

3.1.2 Mesh Regularization

The purely image-based geometry adaptation in the form of (3.3) leads to an underdetermined and ill-conditioned optimization problem. A simple yet illustrative scenario of this occurs when a mesh geometry needs adjustment from a single perspective. In this case, the gradient along the corresponding rendered image provides information exclusively for the vertices visible from that perspective. Consequently, all non-visible vertices do not influence the optimization objective, leading to an underdetermined reconstruction problem with multiple possible solutions, which may be partially noisy or discontinuous depending on the initial conditions. Thus, due to the underdetermined nature of the problem the reconstruction problem results in poor numerical conditions . To address the issues of underdetermination and numerical instability, the optimization problem must be reformulated to establish a unique solution and enhanced numerical stability. This can be accomplished by incorporating the entire geometry parameter space into the optimization objective through regularization. Regularization adds constraints to the solution, such as smoothness or sparsity, which limit the number of potential solutions and improve the numerical stability of the optimization procedure. Mesh regularization losses, as discussed in [81, 85, 143], aim to ensure uniformity in the size and shape of faces and reduce non-uniform and irregular deformations of individual faces. The principles discussed in previous literature are extended in this work to create scale-independent loss functions and designs that can be tailored to specific objectives. As a result, template meshes are introduced to provide prior shape information in a scale-invariant manner.

In the following sections, the individual mesh regularization techniques, which build upon existing designs from the literature as well as extended scale-invariant regularization concepts, are presented. While each design has its unique strengths,

the most effective results can be achieved by combining multiple regularization approaches. To control the sensitivity and contribution of each regularization term to the overall loss, a weighting parameter λ is assigned to each loss design.

Normal Loss: To attain a uniform and consistent surface and penalize strong variations in corner angles, the following normal loss can be employed. For a given a mesh $\mathbf{M}$, the normal similarity loss is calculated as the sum of the normal similarity measures for all adjacent faces $\mathbf{F}^{\blacktriangle} \in \mathcal{N}(\mathbf{F}_i)$ of the corresponding face $\mathbf{F}_i$. Thus, the normal similarity measure is expressed as

$$\mathcal{L}_{\text{nor}}(\mathbf{M}) = \lambda_{\text{nor}} \sum_{\mathbf{F}_i \in \|\mathbf{F}\|} \left(\sum_{\mathbf{F}_j^{\blacktriangle} \in \mathcal{N}(\mathbf{F}_j)} \left(\cos^2 \measuredangle(\mathbf{N}(\mathbf{F}), \mathbf{N}(\mathbf{F}^{\blacktriangle})) - 1 \right) \right), \tag{3.4}$$

where $\mathbf{N}(\mathbf{F})$ and $\mathbf{N}(\mathbf{F}^{\blacktriangle})$ denote the normal vectors for the respective faces, and $\measuredangle$ represents the intersection angle between them [143]. The design of (3.4) ensures that the loss is zero for faces with normal vectors oriented in the same direction, while the loss increases continuously as the angle between them increases. When applied to a uniform and closed mesh, the global minimum of this loss function would result in a perfect sphere.

Normal Loss with Template: To set the default geometry for the loss function, a template mesh $\mathbf{M}^{\diamond}$ is utilized as a stabilizing reference. The template mesh $\mathbf{M}^{\diamond}$ can be initialized using an magnetic resonance image (MRI) geometry reconstruction or a reconstruction from a previous time step. Then, instead of evaluating normal consistency between neighboring faces, a normal similarity measure is applied with respect to the template mesh. The normal similarity is quantified by the inclusion angle between the vertex normals $\mathbf{N}$ in the parameterized mesh $\mathbf{M}$ and the corresponding vertex normals $\mathbf{N}^{\diamond}$ in the template mesh. The overall normal similarity between the two meshes is computed as

$$\mathcal{L}_{\text{nor}}^{\diamond}(\mathbf{M}, \mathbf{M}^{\diamond}) = \lambda_{\text{nor}}^{\diamond} \sum_{\mathbf{N}_i, \mathbf{N}_i^{\diamond} \in \mathbf{N}, \mathbf{N}^{\diamond}} \left(\cos \measuredangle\left(\mathbf{N}_i, \mathbf{N}_i^{\diamond}\right) - 1 \right)^2, \tag{3.5}$$

where the equilibrium is given for $\mathbf{M} \propto \mathbf{M}^{\diamond}$ as a scaled version of the given template mesh. To ensure accurate computation of the normal similarity, it is necessary that the template $\mathbf{M}^{\diamond}$ mesh possesses the same topology as the parameterized mesh $\mathbf{M}$, such that each vertex normal $\mathbf{N}$ in $\mathbf{M}$ has a corresponding vertex normal $\mathbf{N}^{\diamond}$ in the template mesh $\mathbf{M}^{\diamond}$.

Edge loss: The scale invariant edge regularization

$$\mathcal{L}_{\text{edg}}(\mathbf{M}) = \lambda_{\text{edg}} \sum_{\mathbf{V}_j \in \mathbf{V}} \left(\sum_{\mathbf{V}^{\blacktriangle} \in \mathcal{N}(\mathbf{V}_j)} \left(\|\|\mathbf{V}_j - \mathbf{V}^{\blacktriangle}\| - \boldsymbol{l}_{\text{edg}}\|^2 \right) \right) \tag{3.6a}$$

is designed to penalize large variations in face sizes, promoting uniform faces and suppressing "flying vertices". In contrast to many approaches that regulate edge length uniformly at zero, this work requires a uniform distribution of edge length to maintain the overall shape of the model, particularly for vertices that are not supervised by the primary visual loss. To achieve this, the average edge length $\boldsymbol{l}_{\text{edg}}(\mathbf{M})$ in the given mesh is calculated as

$$\boldsymbol{l}_{\text{edg}}(\mathbf{M}) = \lambda_{\text{edg}}\texttt{Mean}(\|\mathbf{V} - \mathbf{V}^{\blacktriangle}\|) \ , \tag{3.6b}$$

where $\mathbf{V}^{\blacktriangle}$ specifies the neighboring vertices. This design promotes uniform edge lengths.

Laplacian loss: The Laplacian loss is introduced to prevent self-intersecting mesh surfaces by penalizing significant geometry changes, as described in [143]. The Laplacian coordinates, defined as

$$\Upsilon_i = \mathbf{V}_j - \sum_{\mathbf{V}^{\blacktriangle} \in \mathcal{N}(\mathbf{V}_j)} \frac{\mathbf{V}^{\blacktriangle}}{\|\mathcal{N}(\mathbf{V}_j)\|} \ , \tag{3.7a}$$

penalizes vertices that deviate from the average of the surrounding centroids. The total Laplacian loss is given by

$$\mathcal{L}_{\text{lap}}(\mathbf{M}) = \lambda_{\text{lap}} \sum_i \|\mathbf{V}_i - \Upsilon_i\| \ . \tag{3.7b}$$

The Laplacian loss promotes smooth and uniform mesh geometries by penalizing significant changes between iterations and encouraging neighboring vertices to move similarly. The respective regularization minimum for a fully connected mesh is a sphere, which is similar to the normal consistency loss stated in (3.4).

Laplacian Loss given Predetermined Minimum: To preset the energy minimum of the Laplacian loss to a predefined geometry, a Laplacian similarity measure is applied with respect to a given template mesh $\mathbf{M}^{\diamond}$. This is achieved by using the Laplacian coordinates $\Upsilon^{\diamond}$ of the template mesh to calculate the loss

$$\mathcal{L}^{\diamond}_{\text{lap}}(\mathbf{M}, \mathbf{M}^{\diamond}) = \lambda^{\diamond}_{\text{nor}} \sum_i \|\Upsilon_i - \Upsilon^{\diamond}_i\|^2 \ . \tag{3.8}$$

The Laplacian coordinates (3.7a) utilized in this design enable the predetermination of the energy minimum of the Laplacian loss.

3.1.3 Silhouette-based Supervised Geometry Reconstruction

Subsequently, the objective is to reconstruct the geometry for specific observation perspectives, focusing on the analysis of geometry regularization. To keep the rendering complexity low and provide flexibility for regularization, the reconstruction process is based on a silhouette based reconstruction formulation without simultaneously incorporating texture-based supervision. To ensure that the developments meet the needs of the surgical scene reconstruction, the first step is to place silhouette reconstruction into the surgical context in Section 3.1.3.1. In Section 3.1.3.2, the silhouette loss is specified for supervision. The geometry reconstruction is conducted in Section 3.1.3.3 for multiple given perspective observations, where in Section 3.1.3.4, the scale propagation is addressed based on a single image observation.

3.1.3.1 Background of the Field of Application

Object silhouettes that are clearly distinguishable from their backgrounds provide a useful, albeit coarse, source of information for geometry reconstruction. This is especially helpful in cases where the structural information of the object is ambiguous and no reliable landmark features are present on the observed texture. While the silhouette information is primarily limited to contributing to the scaling of the geometry to match the observed object silhouette, it can still be challenging to fully reconstruct the camera position due to the inherent complexity of the problem. However, the theoretical hypotheses concerning the knowledge of the camera position and the visibility of silhouettes can be applied in a variety of practical contexts, such as laparoscopic surgeries. For example, in robotic-assisted laparoscopic procedures like the cystectomy, measurements of joint positions and the camera endoscope are provided by the robot kinematics. This reduces the reconstruction problem to a geometry adaption for the given observation perspective. Especially in laparoscopic environments, distinguishable and clear landmark features are hardly retrievable in the image observations. In contrast, the silhouette of the target organ is often still identifiable from different angles, allowing for the scaling of the geometry model by matching the observed object silhouette, making the silhouette-based supervision an effective and reliable approach for highly deformable environments with limited clear visual landmark observations.

3.1.3.2 Silhouette-based Geometry Supervision

The silhouette rendering function $\mathscr{R}_{\text{sil}}(\mathbf{M}, \boldsymbol{\phi}_{\text{cam}}) \mapsto \mathbf{I}_{\text{sil}}$, as introduced in Section 2.3.3.2 through (2.28), calculates an intensity-based silhouette rendering $\mathbf{I}_{\text{sil}}$. For each pixel $(\mathtt{h}, \mathtt{w})$, the resulting silhouetted rendering data $\mathbf{I}_{\text{sil}}^{\mathtt{h},\mathtt{w}}$ indicates whether the corresponding image pixel traces through any object area of $\mathbf{M}$ or is part of the background. In contrast, the simplest approach to extract silhouette information from an image

observation $\boldsymbol{\mathcal{I}}$ involves using a threshold value τ_{sil} to classify the image data into its corresponding silhouetted image representation $\boldsymbol{\mathcal{I}}_{\text{sil}}$. In this method, the silhouetted data $\boldsymbol{\mathcal{I}}_{\text{sil}}^{\text{h,w}}$ is labeled as `true` if the respective pixel intensity $\boldsymbol{\mathcal{I}}^{\text{h,w}}$ holds $\boldsymbol{\mathcal{I}}^{\text{h,w}} \geq \tau_{\text{sil}}$, and `false` otherwise. However, for more complex scenes with multiple visible object entities in the image, a more complex classification algorithm is needed to determine the object mask for the respective image data. For segmenting real-world observations, where a simple threshold is insufficient, there are various segmentation techniques, including unsupervised segmentation filters and data-driven network architectures [2, 24, 25, 45, 56, 86, 129, 153]. To maintain the focus and scope of this chapter on geometry reconstruction,the segmentation problem for image silhouetting is not discussed any further in this chapter. Nonetheless, readers can find a more detailed examination of this problem in Chapter 5.

Given the silhouetted data, the respective images must be evaluated for their similarity in order to formulate the optimization objective. In fact, one way to determine the silhouette loss between the masked silhouette images is to use the Euclidean error of the pixel intensity. However, this method of supervision is weakly conditioned and may result in local minima in the case of unfavorable shapes and overlaps. To address this issue, a more well-conditioned optimization problem can be achieved by using the intersection over union loss, also known as the Jaccard index. This loss is calculated as the ratio of the intersection of the two surfaces to the union of the two surfaces

$$\mathcal{L}_{\text{iou}}(\boldsymbol{\mathcal{I}}_{\text{sil}}, \mathbf{I}_{\text{sil}}) = \lambda_{\text{iou}} \frac{\|\boldsymbol{\mathcal{I}}_{\text{sil}} \cap \mathbf{I}_{\text{sil}}\|}{\|\boldsymbol{\mathcal{I}}_{\text{sil}} \cup \mathbf{I}_{\text{sil}}\|} \ . \tag{3.9}$$

The Jaccard index (3.9) is often used as the standard evaluation metric in the literature for measuring the similarity of binary data sets and is used in this manner for learning bounding boxes or binary object classifications [137]. The loss (3.9) reaches its global minima when the two surfaces align coincidentally.

3.1.3.3 Optimization Geometry Reconstruction

The geometry reconstruction process tailored to the silhouetted data results in minimizing the following objective

$$\mathbf{V}^{\star} = \mathbf{V}_0 + \arg\min_{\Delta\mathbf{V}} \sum_{i=t-h}^{t} \mathcal{L}_{\text{iou}}\Big(\boldsymbol{\mathcal{I}}_{\text{sil}}, \mathscr{R}_{\text{silhou}}\big(\mathbf{M}, \boldsymbol{\phi}_{\text{cam},i}\big)\Big) + \mathcal{L}_{\text{nor}}(\mathbf{M}) + \mathcal{L}_{\text{edg}}(\mathbf{M}) + \mathcal{L}_{\text{lap}}(\mathbf{M}) \ . \tag{3.10}$$

The optimization is performed by finding the optimal delta, $\Delta\mathbf{V}$, to the initial vertex positions, $\mathbf{V}_0$, in order to minimize the intersection over union loss, as formulated by (3.9). In addition, the objective function is regularized by the geometry-dependent regularization terms (3.4), (3.6), and (3.7), where the relative importance of these loss terms is controlled by the loss weighting parameters $\lambda_{\text{nor}}, \lambda_{\text{edg}}$, and λ_{lap}.

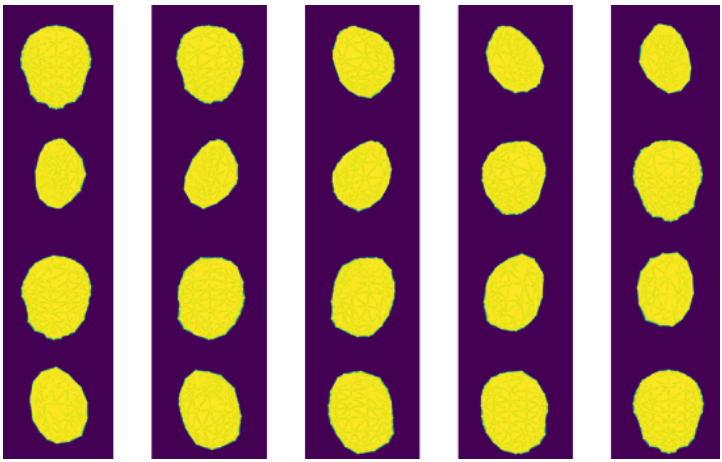
The optimization-based scene reconstruction formulation (3.10) is parameterized by the mesh vertices $\mathbf{V}_j$. This consideration also applies to the general reconstruction problem, where additional factors such as texture parameters are taken into account and added to the optimization's parameter space. Consequently, adapting the parameterized triangle mesh becomes a high-dimensional optimization problem with potentially millions of interdependent parameters, necessitating efficient and numerically stable optimization algorithms.

In this work, the Adaptive Moment Estimation (ADAM) solver is employed for solving the optimization problems considered. The ADAM solver is a suitable choice for this work due to its adaptive learning rate and momentum-based capabilities, which effectively handle the challenges posed by high-dimensional optimization problems. Furthermore, the stochastic nature of the ADAM solver allows for overcoming local minima, improving the chances of finding global optima in complex and non-convex parameter spaces. Additionally, its efficient convergence properties and computational effectiveness make it particularly suitable for large-scale optimization tasks, as encountered in this work. A review and discussion of numerical solvers, including the ADAM solver, can be found in Appendix A.1, where the ADAM update equation is given by (A.2). Furthermore, automatic gradient calculation techniques with respect to the given parameter set are required for precise and dependable optimization. The optimization problem is solved through the use of the ADAM solver (A.1).

For testing, a synthetic data set consisting of 20 samples is generated by rendering a synthetic bladder model. This model is based on the full male anatomy model, which includes both geometry and texture, as presented in the reference [103]. The resulting images can be viewed in Figure 3.1a and are produced from a set of predefined camera perspectives depicted in Figure 3.2a. The corresponding silhouettes, obtained through thresholding, are shown in Figure 3.1b. To reconstruct the bladder's geometry, the optimization problem (3.10) is considered. The resulting mesh reconstruction in ● grey, depicted in Figure 3.2b, represents the converged optimization. The ground truth geometry is overlaid in ● red for comparison. The bladder's geometry can be seen to approximate the ground truth.

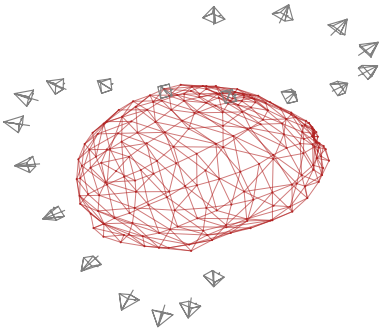
The recorded loss trajectory, as depicted in Figure 3.2d, reaches a minimum around the 100th iteration, at which point the silhouette loss and the deployed regularization losses can be considered to be fully converged. This suggests that beyond this point, the geometry does not continue to adapt to the ground truth as the influence of the regularization terms becomes dominant. However, this is not a concern in this particular scenario as the ground truth data has low resolution and would also incorporate sharp edges in its reconstruction. In real-world applications, smooth and uniform geometries are generally preferred over sharp edges, especially when dealing with fuzzy data such as silhouetted images. Hence, the regularization terms prevent the reconstruction of overly detailed or jagged geometry, resulting in a smoother final model surface for the considered problem.

(a) Textured image dataset; showing a rendered bladder model from 20 distinct perspectives.

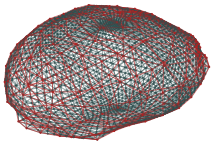


(b) Silhouetted image dataset; corresponding to the textured images shown in (a) and constructed using shape recognition.

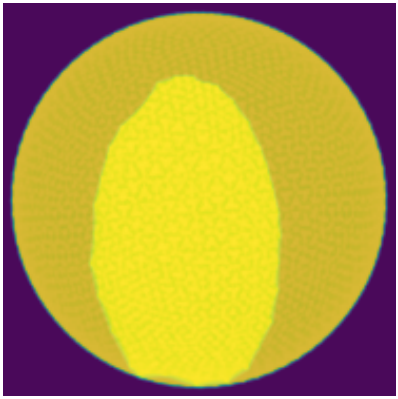
Figure 3.1: Dataset of textured images captured from 20 distinct perspectives, along with corresponding silhouetted images constructed using shape recognition based on the textured image data shown in (a).



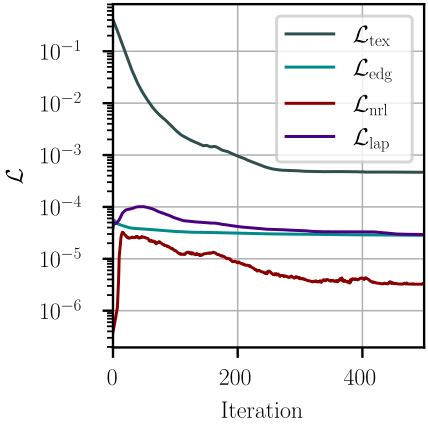
(a) Recording perspectives for the dataset renderings, shown in Figure 3.1a.



(b) Geometry reconstruction vs. model geometry: in grey the adapted model geometry, in red the corresponding ground truth.



(c) The figure shows the superimposition of silhouettes from one of the recording positions, which is highlighted in 3.2a.



(d) Loss trajectory of the geometry adaption.

Figure 3.2: Problem statement for geometry adaptation from multiple perspectives. In (a), the camera perspectives and ground mesh used for data generation are shown. In (b), the adapted model is compared to the ground truth, based on silhouetted image data. For a selected observation pose $\boldsymbol{\phi}_{\text{cam},i=4}$, (c) overlays the corresponding silhouette observed on the image plane with the initial sphere and the respective silhouetted ground truth. Additionally, (d) compares the resulting loss trajectories from the optimization process.

3.1.3.4 Form Preserving Geometry Reconstruction

As the availability of a diverse selection of images from various perspectives is often limited in practice, single-view supervision is considered in the following. The proposed scaling invariant regularization design plays a crucial role in propagating proportions from single-view information to the overall mesh. Based on that, the respective optimization problem is modified to

$$\mathbf{V}^\star = \mathbf{V}_0 + \arg\min_{\Delta\mathbf{V}} \mathcal{L}_{\text{iou}}\Big(\mathcal{I}, \mathscr{R}_{\text{silhou}}\Big(\mathbf{M}, \boldsymbol{\phi}_{\text{cam},i=4}\Big)\Big) + \mathcal{L}^{\diamond}_{\text{nor}}(\mathbf{M}, \mathbf{M}^{\diamond}) + \mathcal{L}_{\text{edg}}(\mathbf{M}) + \mathcal{L}^{\diamond}_{\text{lap}}(\mathbf{M}, \mathbf{M}^{\diamond}) \,, \tag{3.11}$$

analogously to (3.10), where the individual deviations are found in the regularization terms employed. In order to maintain the primary form of the model for all structures, which are not explicitly supervised by the primary silhouette loss (3.9), the shape-preserving (3.5), (3.8) and scale-invariant regularization design (3.6) are deployed.

The resulting mesh $\mathbf{M}^\star$ obtained from the previous multiple view-based reconstruction problem (3.10), where $\mathbf{M}^\star$ is shown in Figure 3.2b, serves as the foundation for testing the template regularization technique. A test scenario involves applying a 25% shrinkage to the mesh $\mathbf{M}_{\text{gt}} \propto \mathbf{M}^\star$ to create a scaled change, with the ultimate goal of fitting the mesh to the target ground truth shape $\mathbf{M}_{\text{gt}}$ through a single silhouette observation based on (3.11). Figure 3.3a presents the initial source mesh, $\mathbf{M}^\star$, colored in ● grey, and the scaled target mesh $\mathbf{M}_{\text{gt}}$, colored in ● red. This figure also displays the selected camera perspective from which the geometry reconstruction is monitored. Figure 3.3b shows the silhouettes of both the output and the target, observed from the corresponding pose, superimposed for comparison.

To validate and test the mesh regularization, the pure scaling from the image plane to the model is employed, utilizing the synthetic scaling of the mesh as a ground truth for comparison with the reconstruction. Both meshes possess the same topology, thus allowing for the assignment of corresponding size information from the current mesh $\mathbf{M}$ to the target mesh $\mathbf{M}^\star$ as $\mathbf{M}_{\text{gt}} \propto \mathbf{M}^\star$. Based on this, the distance similarity of the vertices and the normal similarity can be defined as metrics to evaluate the quality of the reconstruction. The distance similarity $\mathcal{S}_{\text{vert}} = \frac{\mathbf{V}_{\text{gt}} - \mathbf{V}}{\mathbf{V}_{\text{gt}}}$ is defined by comparing the vertex positions of the current observation $\mathbf{V}$ to those of the target observation $\mathbf{V}_{\text{gt}}$, while the normal similarity evaluated analogously by the cosine regularization according to (3.5). In each case, a value close to one corresponds to an optimal level of similarity.

Figure 3.3c shows the resulting loss trajectories, while the respective similarity courses of the parametrized adaption model to the ground truth mesh are displayed in Figure 3.3d for the stated reconstruction (3.3d). As the solution iterates, the reconstruction losses $\mathcal{L}$ decrease, while the model similarities $\mathcal{S}$ increase, indicating the effectiveness of the proposed template regularization design. In the following analysis, the solution progressions are carefully examined and analyzed for a selected weight parameterization, with the aim to avoid any over-interpretation of

individual numerical progressions. Nevertheless, the loss progressions reveal generalizable effects, eliminating the need for detailed parameter studies for a specific weight design.

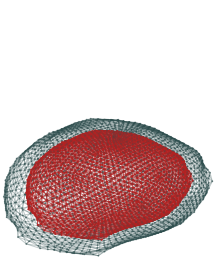
As seen in Figure 3.3c and in the similarity values given in Figure 3.3d, the reconstruction begins to converge around the 700th iteration. The silhouette loss, in particular, is susceptible, displaying high-frequency oscillations. This is due to the fact that even small changes in the mesh vertices may have a significant impact on the resulting image, particularly in areas of overlapping faces. Conversely, hidden vertices may undergo significant spatial changes without any visible effect on the silhouetted image aggregation.

As the initial adaptation model and the ground truth mesh are ideally scaled versions of each other, the template regularizations are zero in the initial iterations. Furthermore, over the progression, the regularization is observed to be at least one order of magnitude smaller than the primary silhouette-based image supervision $\mathcal{L}_{\text{iou}}(\boldsymbol{I}_{\text{sil}}, \mathbf{I}_{\text{sil}})$. Therefore, the geometry adaption is predominately supervised by the silhouette-based image observations, while the regularization becomes relevant only when there is no information available from the primary silhouette-based image supervision. In Figure (3.3d), the shaded area represents the standard deviation of the sensitivities. As overconvergence occurs, the overall mesh undergoes minor adjustments, leading to a consensus in position and orientation of the vertices. Nonetheless, it should be noted that the model may not achieve perfect convergence to 100% similarity or complete consensus to the given ground truth data.

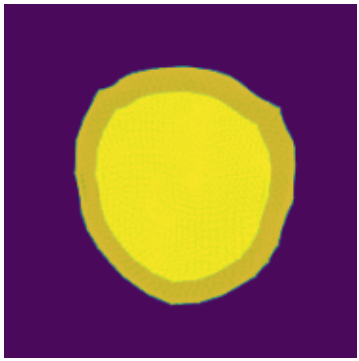
3.2 Texture Reconstruction

For a comprehensive reconstruction, it is necessary to incorporate texture information into the reconstruction process explicitly. As outlined in Section 2.2.4, one approach is to assign texture information to the feature space corresponding to vertex positions. This allows for continuous texture traversal over the surface, facilitating the connection of image patterns with surface textures and corresponding vertex positions. However, the texture representation is limited to coarse meshes, which may not sufficiently capture complex and detailed textures.

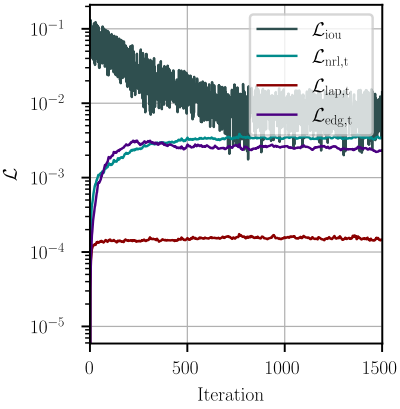
To address the limitation of low surface density in triangular meshes, a mesh subdivision algorithm known as scalable geometry techniques is used [8, 26, 68, 81, 124]. A high mesh resolution is necessary to represent texture information in the corresponding vertex feature space adequately. However, using a high-dimensional mesh representation can make the geometry reconstruction too complex due to the highly coupled geometry regularization, as even a small change in a vertex position can significantly impact all related vertices. Therefore, it is crucial to select a mesh resolution that represents the required level of geometry detail without causing excessive complexity. However, even with an appropriate resolution, the geometry



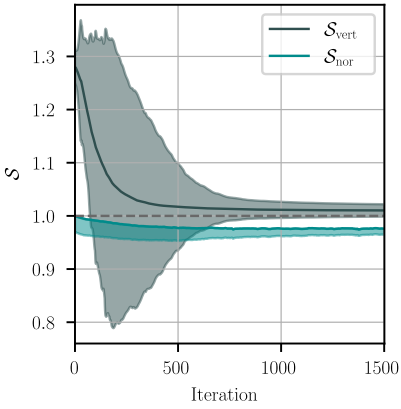
(a) In ● grey, the mesh geometry to be adapted is shown, while the target ground truth geometry is represented in ● red.



(b) Overlaid silhouette image data.



(c) Loss trajectory of the optimization process.



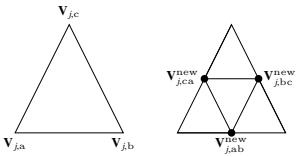
(d) Similarities of the geometry adaptation with respect to the ground truth.

Figure 3.3: Problem statement for geometry adaptation from a single pose. In (a) the unadapted model is compared to the ground truth, considering the given observation pose. The corresponding silhouette from the perspective is shown in (b). Additionally, the resulting loss and similarity trajectories from the optimization process are compared in (c) and (d).

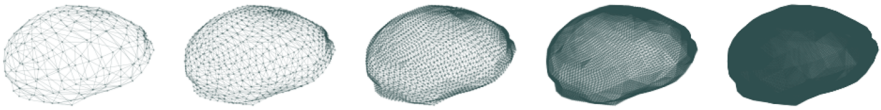
mesh in the corresponding vertex feature space may still not provide sufficient texture representation. To address this issue, a subdivision strategy is followed, which allows for alternating between levels of detail, from a coarse to a highly accurate infinite mesh representation and vice versa, while preserving the correspondences between the respective face topologies. This simultaneously fulfills the requirements for controllable geometric complexity and detailed texture representation [124]. In the following section, the subdivision strategy is presented, where in Section 3.2.2, the optimization problem for a texture reconstruction based on predefined camera perspectives and a predefined model geometry is detailed.

3.2.1 Mesh Subdivision Strategy

Starting with an initial triangular mesh model, the following subdivision procedure generates additional degrees of freedom, enabling an increase in the level of detail. Traditional mesh subdivision methods iteratively subdivide polygons, increasing the number of vertices, faces, and edges and organizing the newly created faces and edges into a revised topology according to predefined rules [68].



(a) Triangle face $\mathbf{F}_j$, which is resolved by subdividing the individual face into four new faces.



(b) Iterative subdivision of a mesh, starting with 212 vertices and going up to 53762 vertices for each of the meshes shown from left to right, with resolutions of 212, 842, 3362, 13442, and 53762 vertices, respectively.

Figure 3.4: Visualization of iterative mesh subdivision using triangle splits as an iterative process.

The face-based method is a common solution to this problem, which involves adding a vertex (the centroid) at the center of each triangle and linking it to the vertices of the initial triangle [8]. However, this generally results in non-uniformly shaped triangles. In contrast, the edge-based subdivision method produces more uniform surfaces by splitting each edge $\mathbf{E}_j$ in the mesh into two individual edges, referred to

as a dyadic split

$$\mathbf{V}_{\mathrm{ab}}^{\mathrm{new}} = \frac{1}{2}(\mathbf{V}_{\mathrm{a}} + \mathbf{V}_{\mathrm{b}}) \ , \tag{3.12}$$

where a new vertex $\mathbf{V}_{\mathrm{ab}}^{\mathrm{new}}$ is inserted at the intersection of $\mathbf{V}_{\mathrm{a}}, \mathbf{V}_{\mathrm{b}} \in \mathbf{E}_j$, defining the edge $\mathbf{E}_j$. In this process, the vertex $\mathbf{V}_{\mathrm{ab}}^{\mathrm{new}}$ is added to the mesh along with a corresponding feature vector $\mathbf{C}_{\mathrm{ab}}^{\mathrm{new}}$, and the mesh topology is revised to include the new edge structure.

New edges are created and connected to new adjacent vertex pairs within the same face, resulting in a triangle being divided into four evenly spaced new triangles, as shown in Figure 3.4. Repeating this procedure establishes a general $\mathtt{itr}_{\mathrm{subdiv}}$-loop subdivision technique. This leads to an increase in the number of faces in the resulting mesh, which is equal to four raised to the power of $\mathtt{itr}_{\mathrm{subdiv}}$. The overall process of subdividing the entire mesh can be summarized in

$$\mathbf{M}^{+} = \mathscr{S}_{\mathtt{itr}_{\mathrm{subdiv}}}(\mathbf{M}) \ , \tag{3.13}$$

where $\mathbf{M}^{+}$ is the output mesh after performing an $\mathtt{itr}_{\mathrm{subdiv}}$-fold subdivision based on an arbitrary input mesh $\mathbf{M}$. This function allows for the efficient and flexible refinement of the mesh to increase the resolution and the capacity of the feature space.

3.2.2 Texture Reconstruction

The subdivision process (3.13) facilitates the representation of the input mesh $\mathbf{M}_{\mathrm{G}}$, and the output mesh $\mathbf{M}_{\mathrm{T}}$ in a single entity for further processing. $\mathbf{M}_{\mathrm{G}}$ resolves the geometry and determines the spatial resolution of $\mathbf{M}_{\mathrm{T}}$ for texture representation. A direct link is established between the two meshes, ensuring that all connections are preserved, and each vertex in $\mathbf{V}_{\mathrm{T}}$ has a corresponding vertex in $\mathbf{V}_{\mathrm{G}}$. This link allows for the determination of the gradient across the texture in $\mathbf{V}_{\mathrm{T}}$ along the correspondence to the vertices of $\mathbf{V}_{\mathrm{G}}$, enabling the use of pure visual image information for geometry adjustments. The interconnection between the meshes allows for detailed and high texture resolution while keeping the geometry complexity limited and manageable.

The problem of mapping texture information onto known geometry can be formulated as an optimization problem

$$\mathbf{C}_{\mathrm{T}}^{*} = \arg\min_{\mathbf{C}_{\mathrm{T}} \in \mathbf{M}_{\mathrm{T}}} \sum_{i=t-h}^{t} \mathcal{L}_{\mathrm{tex}} \left(\boldsymbol{\mathcal{I}}_i, \mathscr{R}\left(\mathbf{M}, \hat{\boldsymbol{\phi}}_i\right) \right) \ , \tag{3.14}$$

where the objective is to adjust the vertex feature $\mathbf{C}_{\mathrm{T}}$ of the mesh $\mathbf{M}_{\mathrm{T}}$ to minimize the texture error between the rendered images and the target images. The texture loss

$$\mathcal{L}_{\mathrm{tex}}(\boldsymbol{\mathcal{I}}_i, \mathscr{R}(\mathbf{M}, \boldsymbol{\phi})) = \lambda_{\mathrm{tex}} \| \boldsymbol{\mathcal{I}} - \mathscr{R}(\mathbf{M}, \boldsymbol{\phi}) \| = \lambda_{\mathrm{tex}} \| \boldsymbol{\mathcal{I}} - \mathbf{I} \|_2^2 \tag{3.15}$$

quantifies the visual difference between synthetic and observed images using the Euclidean distance between all pixel intensities, which is controlled by the weighting parameter $\lambda_{\text{tex}} \in \mathbb{R}^+$. The rendering function $\mathscr{R}$ takes the mesh and pose of the image as inputs and returns the rendered image. The mesh $\mathbf{M}$ represents both the mesh $\mathbf{M}_{\text{T}}$ and the mesh geometry $\mathbf{M}_{\text{G}}$ as the global notation for the model representation. The adaption of the feature space is directly supervised on the image plane, with the rendering function establishing the connection between the 2D image plane and the 3D model representation.

In this specific application, the texture space $\mathbf{C}_{\text{T}}$ is extended through the use of a 3-fold subdivision based on the previously generated mesh geometry $\mathbf{M}_{\text{G}}$, as shown in Figure 3.2b. The generated image and pose data, as depicted in Figures 3.1a and 3.2a, are then employed in the optimization (3.14) for texture reconstruction.

The texture reconstruction for a particular pose is displayed in Figure 3.5a. In this rendering, purely ambient light is applied without any reflection parameters to ensure that the texture values are clearly visible. The reconstruction demonstrates the model's ability to suppress light effects resulting from texture data redundancy in overlapping surfaces or shine effects in the original recordings, as shown in Figure 3.1a.

The same approach is applied with minimal changes to the geometry model, in which the deviation of the geometry is altered by introducing random and unforeseen changes to the vertex positions, accounting for 0.1% of the total size of the geometry dimension. The corresponding texture reconstruction is shown in Figure 3.5b. It is evident that even small deviations in the geometry can cause the texture reconstruction to fail completely. The misalignment between the geometry and texture data makes it impossible to reconcile the interconnected texture values without additional parameterization of the geometry in the reconstruction problem.

3.3 Simultaneous Texture and Geometry Reconstruction

Following individual examination of the optimization problems related to geometry and texture reconstruction, the question naturally arises: what does the reconstruction look like when both geometry and texture are reconstructed simultaneously in a joint optimization?

Therefore, principles of both geometry and texture reconstruction are applied in a holistic optimization based on image observations. To investigate the complexity of the reconstruction jointly, the conflating optimization problem is specified by

$$[\mathbf{V}^\star, \mathbf{C}_{\text{T}}^\star] = \arg\min_{\mathbf{V}_0+\Delta\mathbf{V},\mathbf{C}_{\text{T}}} \sum_{i=t-h}^{t} \mathcal{L}_{\text{tex}}\Big(\boldsymbol{I}_i, \mathscr{R}\Big(\mathbf{M}, \hat{\boldsymbol{\phi}}_i\Big)\Big) + \mathcal{L}_{\text{nor}}^{\diamond}(\mathbf{M}, \mathbf{M}^{\diamond}) + \mathcal{L}_{\text{edg}}(\mathbf{M}) + \mathcal{L}_{\text{lap}}(\mathbf{M}) \quad . \tag{3.16}$$

(a) Texture reconstruction on the given ground truth geometry.

(b) Texture reconstruction on a slightly altered geometry model compared to the ground truth.

Figure 3.5: Texture reconstruction for given geometry and camera perspectives, based on image data similar to the data provided in Figure 3.1a, but with additional structures to enhance landmark information.

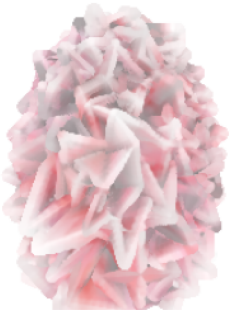
For comparability, the following examples consider a feature space with a total of $|\mathbf{\Delta V}| + |\mathbf{C}_\mathrm{T}| = 14\,284$ elements, comprising $|\mathbf{\Delta V}| = 842$ geometry parameters and $|\mathbf{C}_\mathrm{T}| = 13\,442$ texture parameters. However, the presence of two tasks with distinct dynamics in the optimization increases its complexity.

The complexity of simultaneous geometry and texture reconstruction lies in the strong interconnection of the geometry adjustment. Due to regularization, a change in one vertex directly affects the error similarity of neighboring vertices. Furthermore, texture influences the geometry adaptation, as the texture provides corresponding landmark information, and it influences the geometry adaptation.

To address this complexity, different learning rates are employed for the parameter adjustments during initialization, as dictated by the optimizer design, as discussed in Section A.1. Setting the learning rates is a critical task that can significantly affect the convergence and the final results. Generally, a higher learning rate is recommended for geometry adaptation compared to texture learning. If the texture learning rate is too fast, it can force the geometry to follow the reconstructed texture. Furthermore, due to the interplay between these parameter domains, a stable convergence is generally fragile, which makes the problem even more complex.

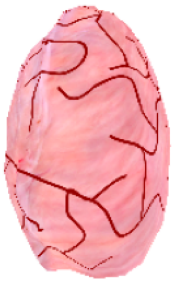
Throughout various parameterizations, it has been necessary to increase regularization to stabilize the optimization process and prevent it from diverging. Moreover, the solution processes observed are sensitive to the weights of the individual losses and the ratio between the texture and geometry learning rates. While testing multi-

ple reconstructions with different weightings and learning rates, satisfactory reconstruction quality has only been achieved to a limited extent. However, a slightly higher learning rate for texture generally has a positive effect on stabilizing the overall optimization process, as the mapped texture pattern can be leveraged for geometry adjustment.



(a) Initial condition generated from a initial spherical geometry model.

(b) Initial condition with noisy geometry and poor texture conditions.



(c) Initial condition with noisy geometry but image data with enhanced landmark information.

Figure 3.6: Simultaneous geometry and texture reconstruction based solely on image observations. The results are obtained from various initial inputs and different texture conditions.

To understand the performance of simultaneous texture and geometry reconstruction and its inherent complexity, three cases with separate, distinct conditions are evaluated and compared to each other. The first case involves general adaptation starting from an uninitialized unit sphere, as demonstrated in the reconstruction procedure for geometry reconstruction in Section (3.1.3.3). The second case involves adapting from a mesh which is slightly noisy but close to the ground truth, as considered in the previous texture reconstruction, with an average deviation of 1.5% from the ground truth mesh dimensions. The third case is examined with the same noised initial mesh geometry, but it uses additional structures to support and increase the available landmark information in the image data. The respective results are shown in Figure 3.6.

Figure 3.6a shows the reconstruction starting from a unit sphere, which demonstrates a very fragile reconstruction. The texture and geometry cannot be stabilized simultaneously in this case, and only the available silhouette information in the data supports the adaptation from a spherical form to the rough silhouette form of the bladder, even though the result is far from the original ground truth. Even for minor geometry deviations, as considered in the second case, the surface texture and geometry still show fluctuations and unreliable adaptation, as shown in Figure 3.6b.

Stable results are achieved only by incorporating essential landmark information, as depicted in Figure 3.6c, which highlights the need of unique landmark information in the image data for successful reconstruction and stabilization of the geometry adaptation. The extraction of reliable landmark structures of cystoscopic images is discussed in Chapter 5.

However, the resulting reconstruction for all examined cases is not satisfactory, as the inclusion of texture does not improve the reconstruction quality when considering the more robust silhouette information, which is also implicitly included in the image information. In summary, the complexity of the optimization problem makes it difficult to stabilize and prone to descending into various sub-minima. Therefore, the simultaneous reconstruction formulation is not suitable for the intraoperative real-world application addressed at this stage and remains a current research topic.

3.4 Summary & Conclusion

In this chapter, the methods for reconstructing scene geometry and texture are investigated both separately and within a combined formulation. The focus of the geometry reconstruction was on addressing the overdetermined reconstruction problem through the development of novel regularization losses. To do this, geometry-specific regularization costs were introduced to reformulate the ill-posed geometry reconstruction into a well-defined optimization problem, enabling a unique optimal solution. To achieve this, advanced mesh regularization terms were presented. Based on state-of-the-art designs, extended regularization designs were introduced to achieve scale invariance in regularization. In addition, by implementing template meshes, the energy minima of a given geometry can be shifted to a predetermined target geometry, which is particularly useful for geometry matching when only a partial view is available. This approach specifies a default geometry for all unsupervised vertices, providing more accurate reconstructions.

To evaluate the proposed geometry reconstruction method, monitoring was performed on the image plane using silhouette-based techniques. The complexities of cystectomy surgery highlight the need for robust silhouette-supervised geometry reconstructions, further underscoring the proposed method as a proof of concept. Clear object silhouettes, marked by a distinct boundary between the object and its background, offer reliable, albeit coarse, data for geometry reconstruction. Such silhouettes are especially valuable when reconstructing objects with ambiguous structural details that lack visible reliable landmark features. The complexity is confined to the geometry parameters, ensuring robust application with a predefined camera position. In this context, accurately representing proportions becomes essential. However, the constraints of laparoscopic conditions mean a lack of trustworthy texture information for geometry reconstruction. Nonetheless, the representation of proportions and geometric forms greatly supports automated information assignment and other spatial data algorithms. With the proposed scale invariant regularization

design, it is possible to propagate scaling or distortion observed on the image plane to the entire model geometry.

In addition, a subdivision technique was utilized to integrate the texture and geometry, enabling geometry parameterization with high resolution, while maintaining manageable complexity. By observing the texture, corresponding adjustments can be made to the geometry. However, it is essential to note that attempting to reconstruct geometry and texture simultaneously using only texture observation is unstable and fails due to the overdetermined nature of the problem. Even with the use of a silhouette loss to reinforce the texture at the same parameterization of texture and geometry, the optimization problem's complexity makes it vulnerable to quickly falling to a local minimum. Consequently, new methods must be developed that go beyond synthetic reconstruction to enable the reconstruction of geometry and texture in real-world applications.

Inverse Differentiable Rendering

4

In the preceding chapter, the main emphasis was on the methodological concepts for geometry and texture reconstruction. The central focus of this chapter is to present a new approach for reconstructing the camera perspective by proposing the concept of an inverse differential rendering approach. Portions of the proposed concept have been published in [117]. The proposed method maps pixel information from the image plane to the 3D mesh surface. It can be thought of as a back projection, referred to as an inverse rendering process due to the reversed input-output flow. Importantly, this formulation couples the projected information, including the surface intersection, normal, and feature value, directly to the mesh parameters, such that any update to the geometry results in a direct and inherent update of the re-projected surface information.

For context, the differentiable (forward) rendering function, introduced in Chapter 2.3, maps the respective model description to the synthetic image plane depending on the respective camera perspective. The differentiable formulation provides a means to quantify the impact of model parameters on image pixel intensities. This, in turn, enables the model to be adjusted as required to match the corresponding image observations.

It is worth noting that a spatially defined objective function is not differentiable due to the discrete images rasterization of the image rendering. This is also true for the presented differentiable (forward) rendering, which only facilitates differentiation of the aggregated pixel intensities. The integer-based image indices prevent the differentiation of spatially defined objective functions across different pixel locations. Nevertheless, to formulate objective functions that can lead to robust reconstructions and enable the solution of geometry matching and pose reconstruction problems, it is essential to incorporate the spatial information of pixel locations.

The inverse rendering concept proposed in this chapter allows for mapping pixel locations to the underlying mesh model. This is made possible by the fact that intersection points are differentiable with respect to the model parameters. The inverse rendering approach can be used to develop new methods for the camera pose reconstruction. As an illustration, consider the scenario where point-based landmark pairs are given in an image. The corresponding point patterns can be aligned directly on the model surface using a tailored loss formulation that incorporates the surface

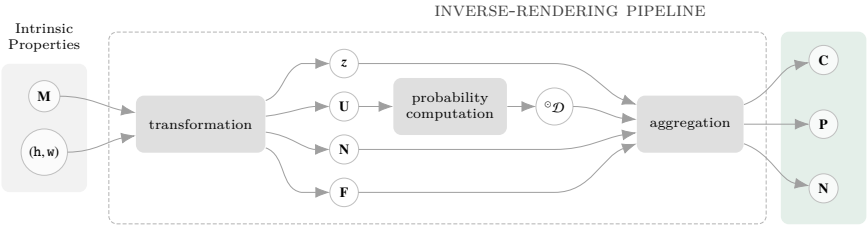


Figure 4.1: Flow diagram of the presented concept of a differentiable inverse mesh rendering pipeline.

information, including the locations and respective surface normals. This approach helps prevent singularities that may occur when aligning patterns based solely on their 2D image locations.

The proposed inverse rendering design raises several specific and relevant research questions that are addressed in this chapter. These questions, which pertain to the design of the proposed inverse rendering function, include:

- How can the information from the image be consistently and accurately mapped back onto the surface without encountering singularities or discontinuities that arise due to the discrete enumeration of the faces in the mesh topology?
- For application, how is optimization-based camera pose reconstruction formulated by employing the inverse rendering concept for a given set of point correspondences?
- How does the inverse rendering process compare to the forward rendering approach in terms of performance, accuracy, and robustness, and in what contexts does it offer distinct advantages?

The design of inverse rendering is presented in the following Section 4.1. The reconstruction of the camera pose based on pre-assigned point matches is addressed in Section 4.2. The impact of the control parameters of the inverse rendering function on the convergence of the camera pose reconstruction is analyzed in Section 4.2.1. Section 4.2.2 investigates the design of the objective function for pose reconstruction by comparing the use of surface positions and normal information from the inverse rendering function with 2D landmark correspondences as commonly exploited in the literature. The purpose of this analysis is to determine the contribution of each approach to the overall performance of the application.

4.1 Inverse Rendering Concept for a Differentiable Back-Projection

For reconstruction, the discrete face indices description presents similar challenges for the inverse rendering process, as do discrete faces on the discrete pixel rasterization

in the forward rendering process. For example, a gradient can be created for the intersection point with respect to the given barycentric coordinates within the face boundaries. However, when back-projected intersection points are close to a face boundary, there is no information available in the corresponding gradient to indicate which neighboring face the back-projection should be adjusted to. The absence of information in the gradient impedes the reduction of the defined loss caused by discrete face indexing of the mesh topology. Furthermore, the back projection process also faces a depth discontinuity issue when faces move in front of other faces across successive iterations.

The proposed inverse rendering process, as shown in Figure 4.1, is based on a probabilistic formulation similar to the differentiable (forward) rendering approach described in Chapter 2.3. However, its primary objective is to evaluate the reliability of a pixel's intersection with the corresponding mesh faces, rather than quantifying the mapping of any mesh information onto the respective pixel color intensity.

This process involves aggregating multiple back-projections of a pixel through probability weighted recombination, resulting in a weighted average that enables a continuous flow of information that can be represented in the gradient. Thus, the gradient provides sensitivity information about the weights of individual back-projections, guiding the direction of adjustment in the optimization process. For the inverse rendering concept, a probability distribution is established in Section 4.1.1 to address discontinuities at face boundaries. Building on this, a depth-dependent weight design is proposed to address depth discontinuities. The final aggregation is then performed while taking into account the designed weighting, whereupon the inverse rendering concept is verified for the camera pose reconstruction problem.

4.1.1 Sensitivity of a Face Intersection for a given Pixel Re-Projection

The back-projection of a pixel location $(\mathtt{h}, \mathtt{w})$ onto a face $\mathbf{F}_j$ is calculated by tracing the flow of information from the camera origin through the corresponding pixel onto the mesh surface. This process can be seen as a linear algebraic back-projection following (2.12), where the resulting intersection point on the mesh is denoted by ${}^{\star}\boldsymbol{P}_j^{\mathtt{h},\mathtt{w}}$. Therefore, the intersection point is determined through the analytical ray tracing calculation, as specified by (2.12b). Focusing on the pose reconstruction, the objective is to adjust the camera perspective so that the pixel's intersection point $\boldsymbol{P}$ coincides with the target position $\boldsymbol{P}^{\star}$ on the mesh geometry. Moreover, in this manner, the respective normal $\mathbf{N}$ and feature $\mathbf{C}$ information can be included in the objective design, depending on the specific prerequisites of the problem.

To determine how the camera perspective must be adjusted to minimize the formulated optimization objective, the intersection point $\boldsymbol{P}$, normal $\mathbf{N}$, and feature $\mathbf{C}$ must be respectively differentiable with respect to the camera perspective. Since the following inverse rendering design comprehensively is used to determine the information on the mesh surface for a given pixel $(\mathtt{h}, \mathtt{w})$, the general rendering variable

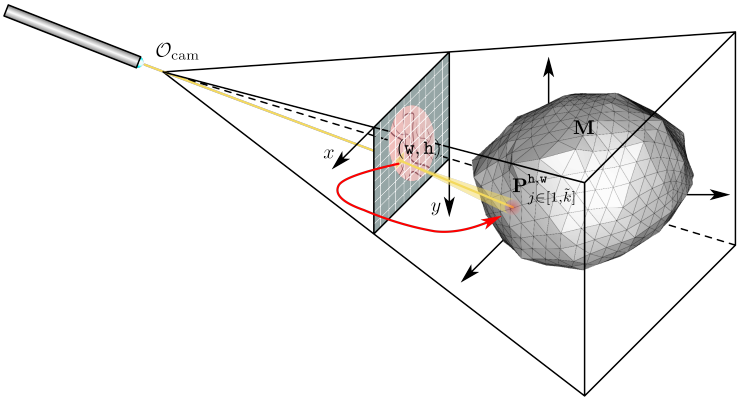


Figure 4.2: Illustration of the proposed back-projection process. A truncated cone is created around the ideal back-projection point ${}^{\star}\boldsymbol{P}_j^{\text{h,w}}$, which represents the sphere of influence that controls the neighboring back projections.

$\boldsymbol{X}$ is introduced for general notation, covering $\boldsymbol{X} = [\boldsymbol{P}, \mathbf{N}, \mathbf{C}]$, respectively. Thus, the respective gradient can be decomposed through

$$\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\phi}_{\text{cam}}} = \frac{\partial \boldsymbol{X}}{\partial \mathbf{F}} \frac{\partial \mathbf{F}}{\partial \boldsymbol{\phi}_{\text{cam}}} + \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\phi}_{\text{cam}}} \ . \tag{4.1}$$

The gradient is factorized by the Barycentric parameterized face planes $\mathbf{F}$ and the depth $\boldsymbol{z}$ of the face information relative to the camera perspective $\boldsymbol{\phi}_{\text{cam}}$.

As the intersection points $\boldsymbol{P}$ are calculated, the partial derivatives $\frac{\partial \mathbf{F}}{\partial \boldsymbol{\phi}_{\text{cam}}}$ and $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\phi}_{\text{cam}}}$ are determined for the respective data $\boldsymbol{X}$ by differentiating the camera projection model in (2.5). However, as discussed before, the gradients $\frac{\partial \boldsymbol{X}}{\partial \mathbf{F}}$ and $\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{z}}$ are subject to discrete constraints of the face planes. Changes in mesh overlaps can cause discontinuities, as shown in Figure 2.17. To overcome these issues, a probability-based formulation is employed that resolves discontinuity in the gradients $\frac{\partial \boldsymbol{X}}{\partial \mathbf{F}}$ and $\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{z}}$. This is achieved through the soft rasterization design described in Section 4.1.2, whereupon the corresponding point aggregation is outlined in Section 4.1.3.

4.1.2 Soft Rasterization

To aggregate the necessary adjacency information around the ideal ray intersection point ${}^{\star}\boldsymbol{P}_j^{\text{h,w}}$, a blur uncertainty $\boldsymbol{\delta}_{\boldsymbol{p}} \in \mathbb{R}^2$ is introduced on the pixel position $\boldsymbol{p}$ through

$$\tilde{\boldsymbol{p}} = \boldsymbol{p} + \boldsymbol{\delta}_{\boldsymbol{p}} \ . \tag{4.2}$$

Moreover, the vectorial blur uncertainty

$$\boldsymbol{\delta}_{\boldsymbol{p}} = \left\{ \boldsymbol{\delta}_{\boldsymbol{p},\{x,y\},j} \in \mathbb{R}^2 \mid \boldsymbol{\delta}_{\boldsymbol{p},\{x,y\},j} \backsim \mathcal{N}(0, \sigma_{\text{inv}}^2), \quad j \in [1,k] \right\} \tag{4.3}$$

is designed based on normally distributed data courses in each direction. The notation $\boldsymbol{\delta}_{\boldsymbol{p},\{x,y\},j} \backsim \mathcal{N}(0, \sigma_{\text{inv}}^2)$ specifies that each respective direction $\boldsymbol{\delta}_{\boldsymbol{p},x}, \boldsymbol{\delta}_{\boldsymbol{p},y}$ in the set $\boldsymbol{\delta}_{\boldsymbol{p}}$ is statistically normally distributed with zero mean and variance of σ_{inv}^2. Based on that, an analytical intersection ray is defined as

$$\tilde{\mathbf{R}}^{\mathtt{h},\mathtt{w}} = O_{\text{cam}} + \mu_{\text{ray}} \left(\begin{bmatrix} \tilde{\boldsymbol{p}}_x \\ \tilde{\boldsymbol{p}}_y \\ f \end{bmatrix} - O_{\text{cam}} \right) , \tag{4.4}$$

which includes the necessary spatial information for a soft aggregation design. Thus, based on the spatially distributed ray bundle the respective surface intersection points $\tilde{\boldsymbol{\mathcal{P}}}_{j\in[1,\tilde{k}]}^{\mathtt{h},\mathtt{w}}$ are determined through the analytical relations given by the analytical projection model (2.12). These intersection points serve as the set of intermediate auxiliary information used to quantify the spatial dependence around the ideal point intersection ${}^{\star}\boldsymbol{P}_j^{\mathtt{h},\mathtt{w}}$. Each individual ray is assumed to intersect the mesh n_{isec} times, resulting in a total of $k \times n_{\text{isec}}$ intersections. However, since the number of intersections at different depths can vary, the index for the total length of the resulting point cloud is specified by $\tilde{k}$. Indeed, it is not necessary to require a continuous coverage of intersection with all faces given in the mesh. Therefore, limiting the number of evaluations to $\tilde{k}$ is more efficient and computationally practical while still maintaining a high level of accuracy.

As shown in Figure 4.2, the ray bundle is associated with a pixel $(\mathtt{h},\mathtt{w})$ and parameterized by the control parameter σ_{inv}, which progressively blurs the ideal ray in the form of a cone. For any pixel $(\mathtt{h},\mathtt{w})$, intersection information is covered in $\tilde{\boldsymbol{\mathcal{P}}}_{j\in[1,\tilde{k}]}^{\mathtt{h},\mathtt{w}}$, and the Euclidean intersection coordinates $\boldsymbol{\mathcal{P}}_j^{\mathtt{h},\mathtt{w}}$ are calculated according to (2.12). The task at hand is to aggregate the $\tilde{k}$ distinct temporary auxiliary points $\tilde{\boldsymbol{\mathcal{P}}}_{j\in[1,\tilde{k}]}^{\mathtt{h},\mathtt{w}}$ into a final 3D back-projection $\boldsymbol{P}^{\mathtt{h},\mathtt{w}}$ for the pixel $(\mathtt{h},\mathtt{w})$ on the mesh surface $\mathbf{M}$.

To determine the contribution of each intermediate intersection point $\tilde{\boldsymbol{\mathcal{P}}}_{j\in[1,\tilde{k}]}^{\mathtt{h},\mathtt{w}}$ with the mesh surface to the final endpoint aggregation $\boldsymbol{\mathcal{P}}_j^{\mathtt{h},\mathtt{w}}$, a probabilistic weight design ${}^{\odot}\mathcal{D}_j^{\mathtt{h},\mathtt{w}}$ is proposed, which is tailored to the spatial aggregation. Here and in the following, the subscript notation $\odot$ is used to refer to the inverse rendering concept. The distribution design in this context is guided by the differentiable design used for the (forward) rendering, as detailed in (2.17), while also addressing the particular challenges posed by the 3D mesh surface model. In order to avoid discontinuities at face boundaries, the influence of each intersection point $\boldsymbol{\mathcal{P}}_j^{\mathtt{h},\mathtt{w}}$ is weighted by its minimal distance to the corresponding face boundaries. The distribution

$${}^{\odot}\mathcal{D}_j^{\mathtt{h},\mathtt{w}} = \text{sigmoid}\left({}^{\odot}\delta_j^{\mathtt{h},\mathtt{w}} \frac{\left({}^{\odot}d_j^{\mathtt{h},\mathtt{w}}\right)^2}{\sigma_{\text{inv}}} \right) \quad j \in \left[1, \tilde{k}\right] \tag{4.5a}$$

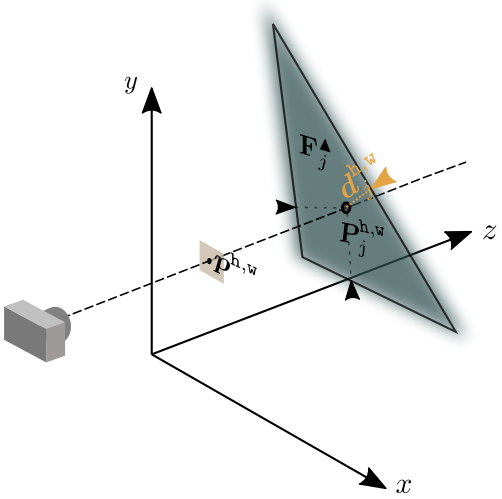


Figure 4.3: The closest distance from the face intersection point $\boldsymbol{\mathcal{P}}_j^{\mathrm{h,w}}$ to the respective face boundaries $\mathbf{F}_j^{\blacktriangle}$ of face $\mathbf{F}_j$ is specified by ${}^{\odot}d_j^{\mathrm{h,w}}$. This distance is used to measure the fragility of an intersection point $\boldsymbol{\mathcal{P}}_j^{\mathrm{h,w}}$ to become a non-intersection point ($\boldsymbol{\mathcal{P}}_j^{\mathrm{h,w}} \notin \mathbf{F}_j^{\blacktriangle}$) if the camera is marginally displaced.

is designed to reduce the weight of critical projections that are close to face boundaries, as they are more likely to obstruct the gradient calculation. The minimal Euclidean distance of a point $\boldsymbol{\mathcal{P}}_j^{\mathrm{h,w}}$ to any boundary of its corresponding face $\mathbf{F}_j$, denoted as ${}^{\odot}d_j^{\mathrm{h,w}}$, is shown in Figure 4.3. The Boolean information

$$ {}^{\odot}\delta_j^{\mathrm{h,w}} = \left\{ +1,\ \text{if } \boldsymbol{\mathcal{P}}_j^{\mathrm{h,w}} \in \mathbf{F}_j^{\blacktriangle};\ -1,\ \text{otherwise} \right\} \tag{4.5b} $$

is determined by whether the ray bundle $\tilde{\mathbf{R}}_j^{\mathrm{h,w}}$ intersects the spanned face plane $\mathbf{F}_j$ inside the specified face boundaries $\mathbf{F}_j^{\blacktriangle}$ as specified by $\boldsymbol{\mathcal{P}}_j^{\mathrm{h,w}} \in \mathbf{F}^{\blacktriangle}$ or intersects the face plane $\mathbf{F}_j$ outside the respective given face boundaries $\boldsymbol{\mathcal{P}}_j^{\mathrm{h,w}} \notin \mathbf{F}^{\blacktriangle}$.

To prevent discontinuities at face boundaries in the aggregated point cloud $\tilde{\boldsymbol{\mathcal{P}}}_{j\in[1,\tilde{k}]}^{\mathrm{h,w}}$, projection points that are close to face boundaries or do not pass through the plane within the face area are given less weight in the distribution ${}^{\odot}\mathcal{D}$. It is important to note that the distribution and distances are defined on the surface plane of the respective faces and are processed in 3D space, in contrast to 2.17, where distances are defined directly on the image plane.

The sigmoid function ensures that ${}^{\odot}\mathcal{D}$ is continuous between zero and one, with σ_{inv} controlling the distribution's sharpness. As σ_{inv} increases, the back projections

become sharper, and the influence of $\boldsymbol{\mathcal{P}}^{\mathtt{h},\mathtt{w}}$ on the unblurred ideal intersection ${}^{\star}\boldsymbol{P}_j^{\mathtt{h},\mathtt{w}}$ becomes stronger. On the other hand, decreasing $\sigma_{\text{inv}} \mapsto 0$ spreads the output $\boldsymbol{\mathcal{P}}^{\mathtt{h},\mathtt{w}}$ more widely, which in turn strengthens the influence of more distant faces on the gradient formation due to their increased significance.

4.1.3 Aggregation of Spatial Surface Information

The distribution design in (4.5a) effectively addresses discontinuities at face boundaries. However, to circumvent optimization impasses potentially arising from alternating overlapping faces, it is crucial to incorporate depth information into the final aggregation of $\boldsymbol{X} = [\boldsymbol{P}, \mathbf{N}, \mathbf{C}]$. This approach enables the assignment of more significant influence to intersection points situated closer to the image plane during the aggregation process, while reducing the impact of those at greater distances. This is because the probability of intersecting hidden faces increases with depth when viewed from the camera's perspective, as illustrated in Figure 2.17 in Chapter 2.3.

To accomplish this, the aggregation weight for an auxiliary point $\boldsymbol{\mathcal{P}}_j^{\mathtt{h},\mathtt{w}}$ is determined by considering the corresponding depth information z_k, which is incorporated into the weight design through

$$ {}^{\odot}\boldsymbol{w}_j^{\mathtt{h},\mathtt{w}} = {}^{\odot}\mathcal{D}_j^{\mathtt{h},\mathtt{w}} \exp\left(z_j^{\mathtt{h},\mathtt{w}} / \gamma_{\text{inv}}\right) . \tag{4.6} $$

The control parameter γ_{inv} can be adjusted to fine-tune the object's transparency and control the consideration of hidden faces. For example, setting the control parameter $\gamma_{\text{inv}} \mapsto 0$ has the effect that only the closest face in view is respected. To determine the final surface information for $\boldsymbol{X} = [\boldsymbol{P}, \mathbf{N}, \mathbf{C}]$ the weighted average is calculated based on the given weighting (4.6). While only the intermediate intersection points $\boldsymbol{\mathcal{P}}$ are relevant for the distribution design (4.5a), the respective normals $\boldsymbol{\mathcal{N}}$ and feature values $\boldsymbol{C}$ are required for the aggregation design. These are determined correspondingly to the intermediate intersection points $\boldsymbol{\mathcal{P}}$. Thus, the aggregation function is defined for the auxiliary state representation $\boldsymbol{\mathcal{X}} = [\boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{N}}, \boldsymbol{C}]$ by

$$ \boldsymbol{X}^{\mathtt{h},\mathtt{w}} = \frac{\sum_j {}^{\odot}\boldsymbol{w}_j^{\mathtt{h},\mathtt{w}} \boldsymbol{\mathcal{X}}_j^{\mathtt{h},\mathtt{w}}}{\sum_j {}^{\odot}\boldsymbol{w}_j^{\mathtt{h},\mathtt{w}}} . \tag{4.7} $$

For notation, the overall inverse rendering process is summarized in the inverse rendering function:

$$ [\boldsymbol{P}, \mathbf{N}, \mathbf{C}] = {}^{\odot}\mathscr{R}_{\boldsymbol{\phi}_{\text{cam}}}^{\boldsymbol{P},\mathbf{N},\mathbf{C}}(\mathbf{M}, (\mathtt{h}, \mathtt{w})), \tag{4.8} $$

which maps the information of a pixel $(\mathtt{h}, \mathtt{w})$ to a given mesh surface $\mathbf{M}$, based on the corresponding camera perspective $\boldsymbol{\phi}_{\text{cam}}$. The notation ${}^{\odot}\mathscr{R}^{\boldsymbol{P}}$,${}^{\odot}\mathscr{R}^{\mathbf{N}}$, and ${}^{\odot}\mathscr{R}^{\mathbf{C}}$ are used to respectively specify the inverse rendered intersection point $\boldsymbol{P}^{\mathtt{h},\mathtt{w}}$, the unit normal $\mathbf{N}^{\mathtt{h},\mathtt{w}}$, and the corresponding feature aggregation $\mathbf{C}^{\mathtt{h},\mathtt{w}}$.

4.2 Inverse Rendering-based Pose Reconstruction

The inverse rendering process facilitates the integration of spatial distribution for corresponding pixel information into the surface model representation. This integration allows for the establishment of a gradient-based reconstruction problem, which is grounded in related point landmark information observed on the image plane. In particular, the relative gradient-based camera pose reconstruction leverages information extracted from the image plane to reconstruct corresponding points on the model, directly incorporating both 3D point information and surface normals through the optimization problem

$$\boldsymbol{\phi}_{\text{cam}}^{\star} = \arg\min_{\boldsymbol{\phi}_{\text{cam}}} \sum_{i=t-h}^{t} \mathcal{L}_{\text{euc}}\Big(\boldsymbol{P}^{*}, {}^{\odot}\mathscr{R}^{\boldsymbol{P}}(\mathbf{M}, \boldsymbol{\phi}_{\text{cam}})\Big) + \mathcal{L}_{\text{nor}}\Big(\mathbf{N}^{*}, {}^{\odot}\mathscr{R}^{\mathbf{N}}(\mathbf{M}, \boldsymbol{\phi}_{\text{cam}})\Big) \,. \tag{4.9}$$

The objective is to match the inverse rendered point cloud ($\boldsymbol{P}$) to the target point cloud ($\boldsymbol{P}^{*}$) by adjusting the camera location $\boldsymbol{\phi}_{\text{cam}}$. This problem is illustrated in Figure 4.4 and describes the optimization problem as stated in (4.9). The loss formulation for this problem relies on the Euclidean distance $\mathcal{L}_{\text{euc}}$ between corresponding point matches and also considers the normal similarities $\mathcal{L}_{\text{nor}}$ of the associated intersection points, as described in (3.4) and (3.5). The included angles between corresponding normal directions quantify the normal similarity by

$$\mathcal{L}_{\text{nor}}(\mathbf{N}, \mathbf{N}^{*}) = (\cos \measuredangle(\mathbf{N}, \mathbf{N}^{*}) - 1)^{2} \,. \tag{4.10}$$

4.2.1 Control Parameter Influence on the Reconstructions Performance

In the following study, the differentiability of the inverse rendering process is analyzed with respect to the control parameters of the blur factor σ_{inv} and the number of re-projections k for the stated optimization problem (4.9).

To demonstrate the influence of the control parameters σ_{inv} and k on the solution process, a set of predetermined target points $\boldsymbol{P}^{\odot}$ and normals $\mathbf{N}^{\odot}$ are exploited, which are illustrated in Figure 4.4. The loss history for the iteratively solved optimization (4.9) is shown in Figure 4.5. First, note that for $\sigma_{\text{inv}} = 0$, the loss decreases initially for a limited number of iterations before remaining constant for subsequent iterations. This behavior is expected, as the gradient with respect to the barycentric coordinates provides information on the decrease in loss. However, when an inverse rendered point is located close to a face boundary, the aggregation is based on the single face plane, which provides no additional information on the direction in which the camera needs to be adjusted to include further face projections. As a result, it is not possible to assign weights to the spatially distributed intersection points used to calculate the actual gradient.

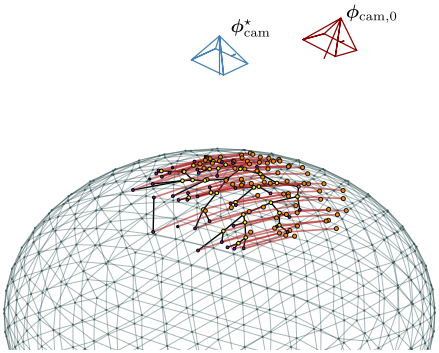


Figure 4.4: To reconstruct the camera position, the pose $\boldsymbol{\phi}^{\star}_{\text{cam}}$ is reconstructed starting from the initial camera position $\boldsymbol{\phi}_{\text{cam},0}$. An Euclidean distance error measure, highlighted in ● red, is defined as the Euclidean distance between corresponding node pairs. The black graph represents the initial graph patterns that are already represented by the model initialization at the current time step, while the observed patterns on the mesh surface are shown in ● yellow.

In contrast, for $\sigma_{\text{inv}} > 0$, the weights receive an information flow, with larger values of sigma increasing the spatial extent of influence on the surface. However, it is essential to note that increasing σ_{inv} does not necessarily improve the performance of the solving process. In fact, an increase in σ_{inv} results in a decrease in accuracy, as observed in the plateauing loss histories for the respective optimizations. Remarkably, the solving process performs comparably well even for a single $k = 1$ and $\sigma_{\text{inv}} > 0$. In this scenario, the information is randomly distributed to neighboring faces between iterations, as defined in (4.3). A continuous gradient is maintained across the barycentric coordinates if the corresponding intersection point hits a suitable surface. However, for finer meshes, the barycentric coordinates may only cover shorter distances, requiring more information to be contributed through the arbitrary aggregation of new face information in the general optimization process.

4.2.2 Comparison of Different Formulations for the Pose Reconstruction

The optimization objective outlined in (4.9) utilizes the inverse rendering approach, which combines 3D points with normal directions to incorporate model-specific spatial information. To evaluate the benefits of this approach, it is compared to the conventional triangulation-based 2D supervision. Instead of transferring image landmarks $\boldsymbol{p}_I$ to the model, as in (4.9), the pattern correspondences $\boldsymbol{P}^{\star}$ represented on the 3D model surface are brought to the image plane through the analytical camera projection model

$$\boldsymbol{p}^{\star} = \mathbf{M}(\boldsymbol{\phi}_{\text{cam},i})\boldsymbol{P}^{\star}, \tag{4.11}$$

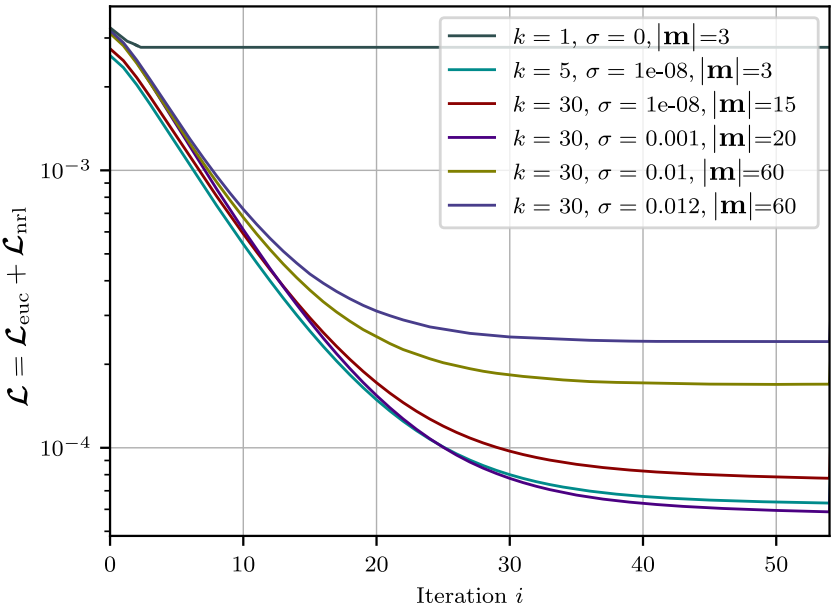


Figure 4.5: The overall error trajectories of the reconstruction formulation are influenced by design parameters k and σ_{inv}. These parameters govern the differentiability of the inverse rendering approach, while the number of matches used, $\|m\|$, impacts the conditioning of the objective function.

ensuring full differentiability by following the analytical camera model (2.5). The distances between the structures represented by the point clouds $\boldsymbol{p}^{\star}_{\text{vas}}$ and $\boldsymbol{p}_{\boldsymbol{I}}$ can then be compared on the image plane through the Euclidean distances between corresponding landmark locations, leading to the optimization problem

$$\boldsymbol{\phi}^{\star}_{\text{cam}} = \arg\min_{\boldsymbol{\phi}_{\text{cam}}} \sum_{i=t-h}^{t} \|\boldsymbol{p}^{\star}_{\text{vas}} - \boldsymbol{p}_{\boldsymbol{I}}\|_2^2 \,, \tag{4.12}$$

which is supervised through point correspondences evaluated by their respective 2D distances on the image plane.

The camera pose reconstruction typically requires at least three matching landmark features between corresponding image data. In contrast, the inverse rendering formulation only requires two matching point pairs between an image observation and a feature point specified on the model surface, relying on the intersection points and normal data with the mesh surface. Nonetheless, given that this optimization

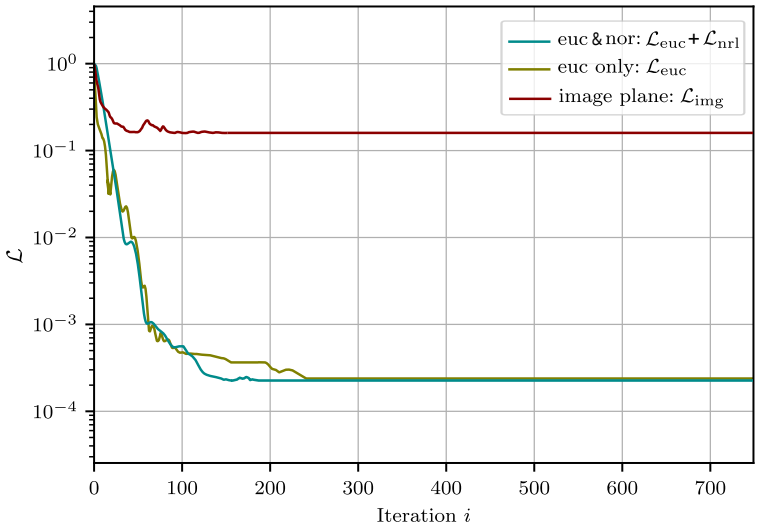


Figure 4.6: Comparison of error trajectories between different definitions of the optimization objective. The objective function defined on the image plane is compared to the objective functions defined directly on the model surface, both with and without using the respective normal directions of the intersection points.

problem is typically regarded as overdetermined, with more constraints than necessary, a robust solution is derived by minimizing the deviation between the overdetermined dataset. This approach contributes to a more dependable reconstruction. The process of identifying landmark points with corresponding matches $\mathbf{m}_{\boldsymbol{p}_I \leftrightarrow \boldsymbol{P}^\star}$ and detecting outliers is covered in Chapter 5 and Chapter 6.

The exploitation of normal information in the reconstruction process leads to an enhanced convergence performance, as depicted in Figure 4.6. This method outperforms the exclusive penalization of Euclidean 3D distances for reconstruction and is also more effective than the image plane-based supervision (4.12).

To ensure more equitable comparisons, the error sum of the total loss is normalized by its initial loss, represented as $\tilde{\mathcal{L}} = \mathcal{L}/\mathcal{L}_0$, , in the multicriteria optimization described by (4.9). While this normalization helps to mitigate dependence on the specific weight of the objective loss, it cannot fully eliminate the influence of particular hyperparameters within the objective functions. Nevertheless, it is crucial to recognize that the unique design of these objective functions complicates comparisons based solely on loss trajectories.

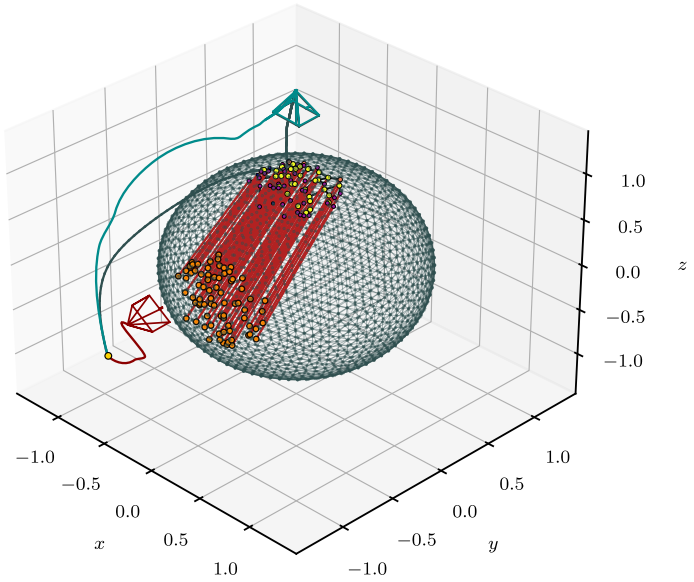


Figure 4.7: The objective function of pose reconstruction problems is defined as the deviation of patterns from the initial camera pose with pre-defined correspondence to the target pattern. The resulting camera poses along the optimized adjustment trajectories are shown, with supervision on the image plane indicated in ● red, supervision using Euclidean information indicated in ● turquoise, and the reconstruction trajectory including both in the optimization objective indicated in ● grey.

Thus, a comparison is only possible to a limited extent. Nevertheless, the inverse rendering-based reconstruction method distinguishes itself as a more effective and resilient technique when compared to the image-based reconstruction, particularly in challenging scenarios where the initial pose is displaced considerably from the target camera pose, as demonstrated by the results in Figure 4.7.

In this case, the reconstruction on the image plane shows that the pose reconstruction terminates in a local minimum, as observable in Figure 4.7. This sub-optimal solution is character-

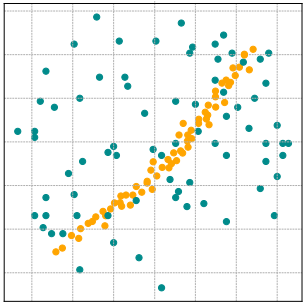


Figure 4.8: Point observations $\boldsymbol{p}^{\star}_{\text{vas}}$ and $\boldsymbol{p}_I$ on the image plane of the scene given in Figure 4.7.

ized by a uniformly distributed loss among the corresponding node pairs $\boldsymbol{p}^{\star}_{\text{vas}}$ and $\boldsymbol{p}_{\boldsymbol{I}}$ on the image plane. The corresponding landmarks on the mesh $\boldsymbol{P}^{\star}$ are projected onto a line-shaped trace on the image plane $\boldsymbol{p}^{\star}$ at the perspective of the terminated pose, as shown in Figure 4.8. This suggests that the camera captures a cross-sectional view of the target area, resulting in a singularity of the corresponding objective function. In contrast, the 3D objective function defined in equation (4.9) reaches the optimal solution smoothly, regardless of whether normals are included in the optimization objective or not.

The incorporation of normal directions in the camera pose reconstruction results in a smoother pose adaption towards the target pose, compared to using only surface positions, which shows numerous fluctuations. Furthermore, it is observed that when using a normal loss, the pose adaptation is perpendicular in the adaptation direction in the final stages before reaching the optimal pose. This indicates that the normal loss has already converged, and the camera is now adjusting along the z-axis in the camera coordinate system to optimize the remaining Euclidean loss and align precisely with the target pose. However, it is possible to reduce this effect by decreasing the weight of the normal loss, without compromising the robustness of the results.

The inclusion of normal information in the optimization objective leads to a more stable and reliable reconstruction than an image-based pose reconstruction. The normal loss aids in achieving a stable convergence towards the optimal pose by providing a robust specification of the gradient direction and preventing the termination in sub-optimal local minima. Conversely, the projection of 3D model surfaces onto the image plane introduces ambiguity, as multiple points in 3D can correspond to a single image point. This can reduce the amount of information available in the 2D image, resulting in a tendency towards convergence in neighboring minima. Although errors can be minimized through the use of appropriate weighting and regularization, the Euclidean surface error design is more reliable as it is not subject to any projection ambiguity. Additionally, the incorporation of directional spatial information further enhances the robustness of the pose reconstruction, as evidenced by the results presented in Figure 4.7 and Figure 4.8. This is particularly relevant in challenging intraoperative scenarios where significant deviations occur from the initial to the optimal solution.

4.3 Summary & Conclusion

In this chapter, the concept of inverse rendering was presented and applied to the issue of camera pose reconstruction. The proposed inverse rendering concept involves remapping image information from the pixel to the model representation to enable the differentiability of model-level information (surface point intersections, corresponding surface normals, and texture features) across discrete surface boundaries. Therefore, a probabilistic description of the reliability of individual projection

bundles of pixel information was formulated. This enables the differentiation of intersection points along surface boundaries and the computation of continuous and invariant weights, providing sensitivities of the respective mesh faces. The differentiable formulation of inverse rendering not only facilitates the transfer of information from the image to the model level but also enables the utilization of the resulting data in gradient-based reconstruction problems.

In the context of pose reconstruction, utilizing corresponding 3D landmark locations and corresponding normal directions on the mesh surface may result in a more robust reconstruction than the traditional image plane-based loss definition. The 3D and normal direction formulations produce a reliable gradient directed towards the global minimum. The normal direction specifies the orientation, resulting in robust convergence. In contrast, significant deviations in the initial conditions during image plane reconstruction often result in constellations where the optimization process terminates in local sub-minima. This is due to the projection ambiguity along the projection line, resulting in singularities that worsen the conditioning of the objective function compared to using 3D information directly. The inverse rendering approach has proven to be an effective method for transferring observations from 2D to the 3D model scene, offering superior accuracy and reliability compared to conventional supervised 2D reconstruction methods. Therefore, it is a valuable tool for intraoperative applications. However, the availability of corresponding landmark data is still a prerequisite for inverse rendering-based pose reconstruction, which remains an outstanding task for the real-world application of the outlined pose reconstruction capability.

5 Landmark Extraction

To realize an accurate localization, the captured images must possess clearly retrievable and well-distinguishable patterns. This requires that a given image contains features that are characteristic of the location of the captured surface so that upon reacquisition of the surface, the respective image observation can be unambiguously assigned to the same feature pattern. Owing to their location specificity, these features are referred to as landmark features and facilitate orientation.

Identifying unique landmark features is fundamental for reconstruction approaches that rely on the specific location of features within an image, such as the pose reconstruction method described in the previous chapter. Nevertheless, the reconstruction concept of this work was initially motivated by an image-to-image correlation on a pixel level rather than relying on a sparse pattern representation. However, as discussed in Section 3.3, even for the image comparison of a synthetic database, the geometry reconstruction can lead to an ill-posed optimization problem, which tends to become trapped in a local minimum. This is expected to be even more pronounced for real-world intraoperative image data, which is often noisier and non-uniform.

To mitigate this limitation, it is essential to isolate the significant structures to facilitate a more robust reconstruction objective. This involves reducing the information content to only the most significant and reliably visible structures and eliminating interfering and inconsistent structures. However, extracting the Landmark feature for intraoperative applications presents a scientific challenge on its own. The so-called ORB landmark features are commonly used as the state-of-the-art in robotics for landmark identification and localization problems, as reviewed in Section 1.3. However, as demonstrated in Figure 5.2, ORB features are unable to capture the local characteristics of cystoscopic images accurately. This may severely impact subsequent landmark matching since ORB features rely on brightness and color differences, making them susceptible to inaccuracies in the presence of noise or blurriness.

In this study, the visible vascular structures on tissue surfaces are used as landmarks for intraoperative orientation during cystoscopic interventions, as shown in Figure 5.1. These structures can be represented as graphs, providing a deformation-invariant feature space that is robust to changes in graph length caused by deformation. Despite potential changes in graph length caused by deformation, the

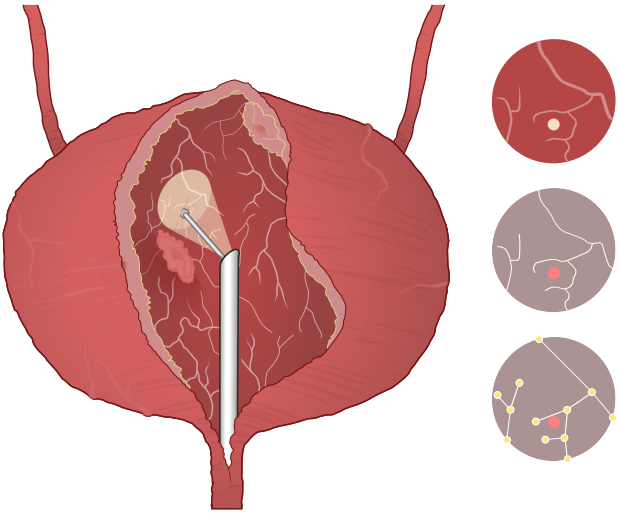


Figure 5.1: The connectivity of the blood vessels is invariant under deformation and thus provides robust landmark information for reconstruction. The representation of the vascular network as a graph, with nodes representing vessel bifurcations and edges along the vessel trajectories, offers a deformation-invariant description and the ability to match corresponding nodes across multiple observations.

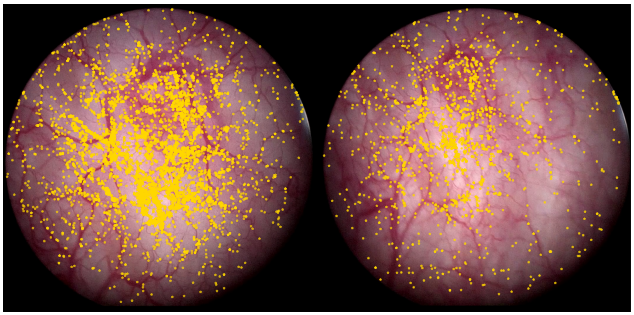


Figure 5.2: Detection of ORB features [109] in a succeeding pair of cystoscopic image observations. The identified ORB feature locations are indicated in yellow. The inconsistent and noisy distribution of feature locations makes it challenging to match corresponding features across the respective image observations.

underlying structure remains unchanged, leading to the use of the term 'deformation-invariant feature space' to describe the underlying landmark representation [7, 116, 150, 151]. During deformation the inter-connectivity of the vascular structure remains unchanged even if the graph is deformed spatially. Thus, the vascular structure cannot intersect in an unexpected and different pattern due to deformation. This makes the graph representation an effective representation for deformable vascular structures in endoscopic images, ensuring the consistency of the graph structure. Moreover, the graph representation redefines the matching problem, enabling the adaptation of a robust graph matching procedure to align individual graphs extracted from different observations, which is discussed in the up-followed Chapter 6. The main objective of this chapter is to extract reliable graph features from an intraoperative image observation while addressing the following challenges:

- Separating visible vascular structures suitable for landmarking from the observed intraoperative image data.
- Representing the identified image structures using a graph representation suitable for graph matching.
- Make the graph extraction robust against small modifications in intensity and blur in the raw data.

To address the problem of vascular pattern and graph extraction, a multi-step solution is proposed. The procedure is illustrated in Figure 5.3. First, a preprocessing classification network is employed to enhance the data quality by separating unusable image samples and regions. Second, a filter design is utilized to extract the dominant vascular structures, ensuring only relevant information is used in subsequent steps. Additionally, curvature information is identified to improve the distinguishability of landmark representation and aid in the graph-matching task. Finally, a data-driven approach is introduced through a network architecture designed to handle changes in image observations, such as blurs or lighting conditions, thereby improving the robustness of the graph-based landmark representation.

These approaches are outlined in the following sections: preprocessing in Section 5.1, structure segmentation in Section 5.2, pattern-based graph extraction in Section 5.3, data driven-edge extraction in Section 5.4, and attribute extraction to enhance specific landmark information in Section 5.5.

5.1 Preprocessing of the Image Data

For the endeavor of this work, the University Hospital of Tübingen has provided the raw video data of a cystoscopic intervention. To extract pertinent information from the video data, several preprocessing steps are required, including trimming, sampling, and resizing of the image data, as illustrated in Figure 5.4. These steps enable the decomposition of the video material into individual images that can be used as input for further processing in a persistent data format.

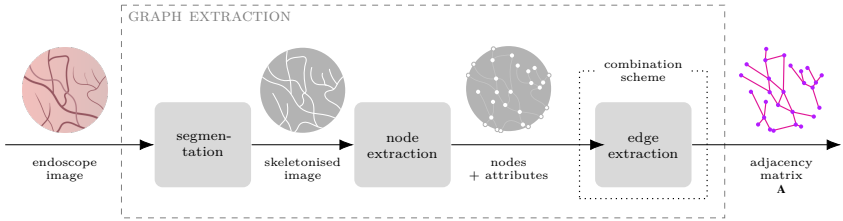


Figure 5.3: Graph Extraction Pipeline including the following preprocessing stages: Segmentation & Preprocessing: Replaces the segmentation step in the original graph extraction procedure. Node extraction & Graph extraction: Extracts nodes and node attributes from the skeletonized image output of the previous network. Edge extraction & Graph extraction: Completes the graph by extracting the edges from the skeletonized image.

Non-contributory video segments that do not assist with the localization task must be eliminated. As these sections are recorded, the vascular structure is invisible, or the view is disturbed. For instance, when the endoscope is removed from the bladder for cleaning, or the view is blocked by cut tissue, as seen in the data shown in Figure 5.5, the resulting images do not contain any useful landmark information and even may harm the localization algorithm later. Thus, disturbed images must be disregarded. In addition to entirely none usable images, certain images may have sub-regions that are suitable for landmark identification. Therefore, pixels that represent moving objects such as tools, air bubbles, and resected tissue must be individually identified, as these can disrupt the orientation process. Figure 5.5 presents a set of disturbing images. Some images are obstructed by air bubbles, as depicted in Figure 5.5a. Others are blocked by the moving electrical cutting loop, as shown in Figure 5.5b, or covered by cut and flowing tissue, as illustrated in Figure 5.5c.

Thus, the objective is to generate for each image $\boldsymbol{I}_i$ its corresponding mask $\mathbb{M}_i$, which accordingly restricts the usable image space. Therefore, a Convolutional Neural Network (CNN) architecture is employed to segment the image data. More specifically, the so-called U-Net architecture is used in this work as a state-of-the-art network architecture for the image segmentation task. The U-Net network architecture was

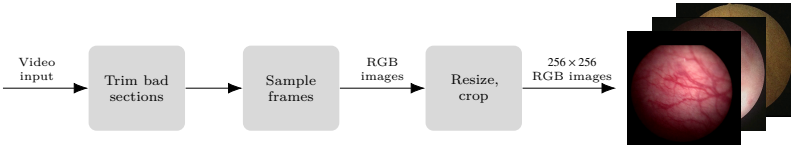
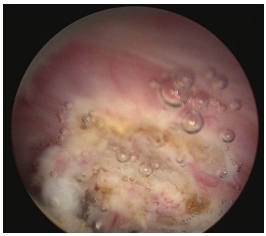
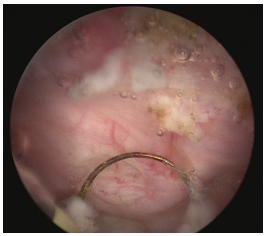


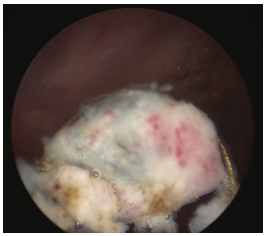
Figure 5.4: Pre-processing steps for sampling subsequent images from raw video data.



(a) Obstructed view due to air bubbles and cauterized tissue.



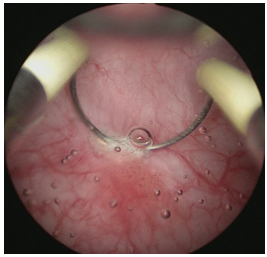
(b) Obstructed view due to the moving electrical cutting loop.



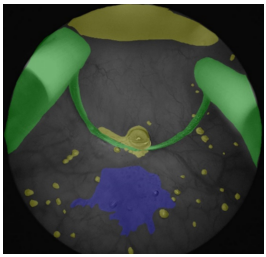
(c) Obstructed view due to the freely flowing tissue sample.

Figure 5.5: Video frames captured with an obstructed view.

first introduced in [106] for biomedical image segmentation and further leveraged and extended in [45, 88, 108, 155] for wide range of image segmentation tasks. The U-Net architecture is referred to by its distinctive U-shaped network architecture that combines a contracting path for capturing context and a symmetric expanding path for precise localization based on a convolutional filter design. The network design allows for accurate segmentation of corresponding images based on the learned filter parameters. A general overview and discussion of the U-Net network architecture can be found in Appendix A.2. The specific architecture used in this work is based on the ResNet50 backbone [118]. It encompasses a set of 25 million trainable parameters, which are optimized during training.



(a) Input image.



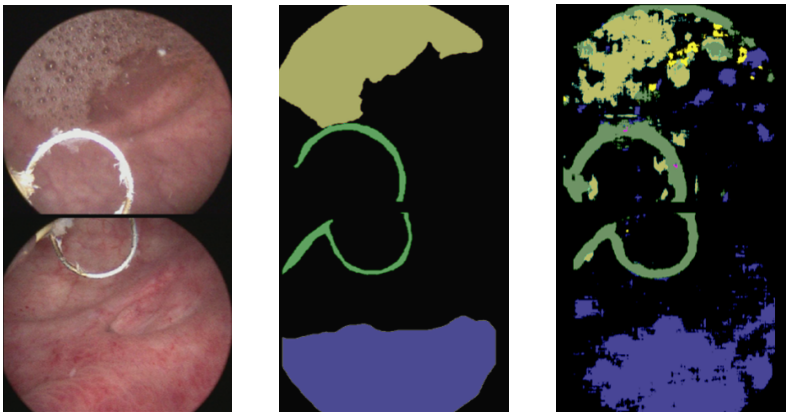
(b) Manually annotated ground truth.

Figure 5.6: Training data comprising input images and their corresponding labeled ground truth segmented images. The images depict cystoscopic procedures using a electrode resection loop, with segmentation performed for the ● tool, ● bluer and bubbles, ● suspicious tumorous tissue, and (by default) background pixels.

For training the parameterized network, a data set of 1600 hand-labeled images was built. These images are diversified through augmentation techniques such as image

rotation, translation, and mirroring to enhance the training process. The most significant perturbations in the image can be detected by identifying classes such as 'tool,' 'bubble,' or 'tumor,' for which the network has been trained. Figure 5.6 displays a hand-labeled dataset with the corresponding network predictions.

The training process was performed for over 230 epochs with an 80/20 data split between training and validation data. The results of a validation batch for the fully trained network are depicted in Figure 5.7, which showcases two randomly selected samples. The prediction, as demonstrated in Figure 5.7c, accurately segments the tool, electrical loop, and water bubbles, as depicted in the input data presented in Figure 5.7a. While the network predictions may be less reliable in detecting suspicious tumor tissue, this does not affect its performance in masking the raw data for landmark identification. However, tumor classification is not covered in this study and would entail a larger dataset, leading to an expansion of the current research objectives. Ultimately, data-driven image segmentation facilitates the identification of potential regions in the image that are suitable for landmark extraction. These regions are specified in the image-specific mask $\mathbb{M}_i$ and allow for the detection of reliable and persistent pixels that potentially can serve as landmark features, thus enhancing the accuracy of the orientation process.



(a) Input Images. **(b)** Ground Truth. **(c)** Predictions.

Figure 5.7: A comparison of a given input image, the corresponding ground truth labels, and network prediction based on the input image data. The ground truth and prediction are color-labeled according to the following categories: ● tool, ● bluer and bubbles, ● suspicious tumorous tissue, and background pixels (default). The tool and water bubbles are detected satisfactorily compared to the ground truth. However, the prediction for suspicious tissue is over-classified, which is a known limitation of the current approach and one that requires further investigation and improvement.

5.2 Vascular Pattern Segmentation

Vascular structure segmentation in cystoscopy images is a challenging problem due to the poor quality of the images, which often contain signal noise and distortion effects and vary significantly in lighting conditions and imaging perspectives. This leads to significant variations in the appearance of blood vessels between subsequent frames. In the literature, there are both supervised and unsupervised approaches to this problem. Manually annotated training data is used to train pixel-based discriminators in supervised methods, including deep neural networks and Gaussian process architectures. As shown in [47, 54], these approaches are effective in classifying retinal blood vessels. However, the specific characteristics of cystoscopy images, such as relatively low resolution and high levels of noise, pose unique challenges for blood vessel segmentation in this context. Although there is a wealth of research on the segmentation of vascular structures, a large portion of the latest publications is focused on retinal image segmentation and rely on supervised segmentation techniques [15, 47, 54, 92, 129, 155].

For retinal treatments, there is usually an extensive database available as data is collected at the time of diagnosis and treatment and is often annotated for diagnostic purposes. However, the vascular structure of the urinary bladder is not typically a diagnostic indicator. Therefore, pre-labeled data is not readily available for this type of data. Given this, unsupervised classification methods are preferred in this study, as they can be more effectively adapted to the characteristics of cystoscopy and offer a higher degree of segmentation sensitivity without the need for an annotated dataset.

The overall pattern segmentation flow followed in this work is illustrated in Figure 5.8. The goal is to represent the vascular structures in the image space as a binary narrow line structure. Therefore, an unsupervised segmentation filter is first applied to separate the vascular structures from the surrounding tissue. Next, the resulting structures are binarized and thinned to a single pixel width, forming a skeleton that highlights the primary paths of the vascular structures.

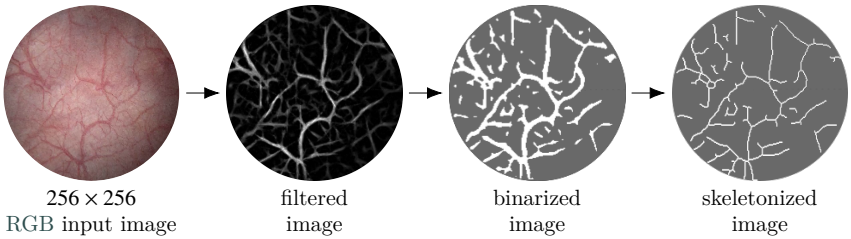


Figure 5.8: Workflow for preprocessing images to enhance main visible structures in a binary format, to be used as landmark information.

5.2.1 Segmentation Filter for Vascular Structures

The filter design employed in this work is based on multiple shifted filter responses to detect pathlines and bar structures, as presented in [5]. The filter is designed to segment blood vessels regardless of their orientation, which is critical for its robustness. Furthermore, the filter's response is developed through a flexible template structure, allowing for precise regulation of its selectivity via a customizable parameterization procedure.

As an initial preprocessing step, the input colored image $\boldsymbol{\mathcal{I}}$ is converted to its corresponding grey-scale image representation $\boldsymbol{\mathcal{I}}_{\text{grey}}$ to focus on the pattern of vascular structures within the image. This reinforces the segmentation design to rely on actual pattern courses rather than specific color distributions, which can be influenced by patient-specific factors. The texture contrast of a given color distribution also highly depends on the patient's bladder condition, which can be influenced by age and disease. In contrast, grey-scale images provide a more accurate representation of the structural course and interconnectivity. The conversion to grey-scale is achieved by calculating the average of the red, green, and blue channels at each pixel. This results in a single grey-scale intensity $\boldsymbol{\mathcal{I}}_{\text{grey}}^{\mathtt{h},\mathtt{w}}$ value for each pixel $(\mathtt{h},\mathtt{w})$.

The core design of the segmentation filter is based on the Difference of Gaussian (DoG) filter

$$\mathtt{DoG}_{\sigma^2_{\mathtt{DoG}}}(\boldsymbol{p}^{\mathtt{h},\mathtt{w}}) = \frac{1}{2\pi\sigma^2_{\mathtt{DoG}}} \exp\left(-\frac{\left(\boldsymbol{p}_x^{\mathtt{h},\mathtt{w}}\right)^2 + \left(\boldsymbol{p}_y^{\mathtt{h},\mathtt{w}}\right)^2}{2\sigma^2_{\mathtt{DoG}}}\right) - \frac{2}{\pi\sigma^2_{\mathtt{DoG}}} \exp\left(-2\frac{\left(\boldsymbol{p}_x^{\mathtt{h},\mathtt{w}}\right)^2 + \left(\boldsymbol{p}_y^{\mathtt{h},\mathtt{w}}\right)^2}{\sigma^2_{\mathtt{DoG}}}\right). \quad (5.1)$$

The DoG filter is a kernel function that compares the pixel intensities of adjacent pixels with respect to a reference pixel at position $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$. The filter consists of a passing central region, which allows certain pixel intensities to pass through the filter unchanged, and a blocking edge region, which blocks or attenuates other pixel intensities. Moreover, the sensitivity of the passing filter shape region can be controlled by adjusting the filter variance $\sigma^2_{\mathtt{DoG}}$. In image processing, the DoG filter is commonly used to detect specific patterns within an image, as it can effectively highlight differences in pixel intensities. This filter is widely used for edge detection, feature extraction, and other image-processing tasks. It is a well-established method in the literature and has been employed in various research studies, such as [6, 7]. Figure 5.9 illustrates the corresponding filter design, providing a visual representation of its distribution.

Thus, for a given image the corresponding filter response is given by the convolution of the image with the DoG filter kernel (5.2) where the convolution operation is given by

$$\boldsymbol{\mathcal{I}}_{\mathtt{DoG}}\left(\boldsymbol{p}_x^{\mathtt{h},\mathtt{w}}, \boldsymbol{p}_y^{\mathtt{h},\mathtt{w}}\right) = \boldsymbol{\mathcal{I}}_{\text{grey}} \circledast \mathtt{DoG}_{\sigma^2_{\mathtt{DoG}}}. \quad (5.2)$$

To tailor the filter design to the segmentation of vascular line structures, a segmentation filter is developed that involves multiple sampling procedures around the

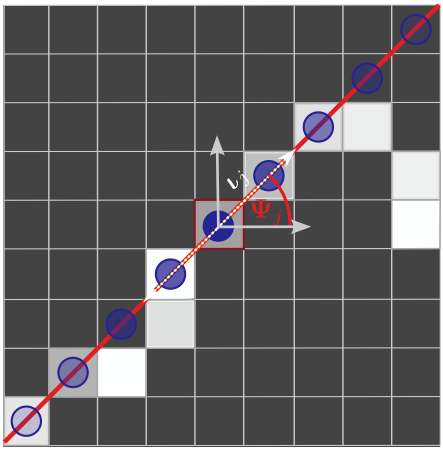


Figure 5.9: Determination of segmented structure space $\mathcal{I}^{\mathtt{h},\mathtt{w}}_{\mathrm{sgt}}$ through pixel-wise aggregation. A pixel at location $(\mathtt{h},\mathtt{w})$ is identified as a bar structure if adjacent pixels in the Gaussian-filtered image $\mathcal{I}_{\mathtt{DoG}}$ individually exhibit high intensities. In this segmentation approach, closer neighboring points carry greater weight than more distant pixels. The evaluation is considered for all pixels along a straight line, parameterized by the direction $\boldsymbol{\Psi}_j$ and the distance $\boldsymbol{\iota}_i$.

reference point, as illustrated in Figure 5.9. Concentric circles are sampled around the spatial reference pixel location $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$, where the corresponding set

$$\Omega = \left\{ \left(\sigma^2_{\mathtt{DoG}}, \boldsymbol{\iota}_i, \boldsymbol{\Psi}_j \right) \,\middle|\, \boldsymbol{\iota}_i \in \boldsymbol{\iota},\ \boldsymbol{\Psi}_j \in \boldsymbol{\Psi} \right\} \ , \tag{5.3}$$

is parameterized by the standard deviation $\sigma^2_{\mathtt{DoG}}$, the distance $\boldsymbol{\iota}_i \in \boldsymbol{\iota}$, and the orientation $\boldsymbol{\Psi}_j \in \boldsymbol{\Psi}$. These parameters are chosen from sets of pre-defined set of distances $\boldsymbol{\iota}$, and orientations $\boldsymbol{\Psi}$. A given distance parameter $\boldsymbol{\iota}_i$ specifies the radial distance from the reference pixel $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$, while the orientation parameter $\boldsymbol{\Psi}_j$ specifies the angle of rotation of the sampled points around the reference pixel $(\mathtt{h},\mathtt{w})$.

Thereby

$$\boldsymbol{\iota}_i = \left\{ \pm \frac{l}{|\boldsymbol{\iota}|} \,\middle|\, \boldsymbol{\iota}_i \in \boldsymbol{\iota} \right\} \tag{5.4}$$

determines a pre-parameterized distance set, where the specified length l is uniformly sampled in the positive and negative directions by a total number of $|\boldsymbol{\iota}|$, while

$$\boldsymbol{\Psi}_j = \left\{ \frac{\pi}{|\boldsymbol{\Psi}|} \,\middle|\, \boldsymbol{\Psi}_j \in \boldsymbol{\Psi} \right\} \tag{5.5}$$

is a uniformly distributed orientation set used to represent the respective circular information around the referenced pixel position $\boldsymbol{p}_{x,y}$. Hence, for a given parameter

triple $\Omega_{i,j}$ the corresponding euclidean position is given by $\Delta \boldsymbol{p}_{x,\{i,j\}} = -\boldsymbol{\iota}_i \cos \boldsymbol{\Psi}_j$ and $\Delta \boldsymbol{p}_{y,\{i,j\}} = -\boldsymbol{\iota}_i \sin \boldsymbol{\Psi}_j$.

To gather the intensity information of adjacent pixels for a given pixel position $\boldsymbol{p}_{x,y}$, the DoG filter (5.1) is applied to the image while taking into account the set of parameters (5.3), which includes the radial distance and orientation to the reference pixel $(\mathtt{h}, \mathtt{w})$. By sampling multiple points at different orientations and distances around a reference pixel $\boldsymbol{p}_{x,y}$, the corresponding angle $\boldsymbol{\Psi}_j^{\star}$ that leads to the maximum intensity set is determined by

$$\Omega_{\boldsymbol{\Psi}_j^*} = \max_{\boldsymbol{\Psi}_j \in \boldsymbol{\Psi}} \left\{ \sum_j \boldsymbol{\mathcal{I}}_{\mathtt{DoG}} \left(\boldsymbol{p}_x^{\mathtt{h},\mathtt{w}} - \boldsymbol{\iota}_i \cos \boldsymbol{\Psi}_j,\ \boldsymbol{p}_y^{\mathtt{h},\mathtt{w}} - \boldsymbol{\iota}_i \sin \boldsymbol{\Psi}_j \right) \middle| \boldsymbol{\iota}_i \in \boldsymbol{\iota} \right\} . \tag{5.6}$$

As a result, the specified subset $\Omega_{\boldsymbol{\Psi}_j}$ represents the line structure with the strongest filter responses of (5.1). The angle $\boldsymbol{\Psi}_j$ specifies the direction of the bar structure, where the set $\Omega_{\boldsymbol{\Psi}_j^*}$ is a discrete sample set that defines the identified bar structure's location, orientation, and distance from the reference location $\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$.

Figure 5.9 depicts the specification of a template pattern, where the intensities passing through the segmentation process are detected through the intensity search (5.6). In the example shown, there are five different passing points with the bar for each circle. The number of points depends on the complexity chosen for the parameter set (5.3).

The final objective is to merge the intensity set specified by (5.6) into a single final segmentation signal for the pixel at position $\boldsymbol{p}_{x,y}$. Finally, the image segmentation $\boldsymbol{\mathcal{I}}_{\mathrm{sgt}}$ for the given input image $\boldsymbol{\mathcal{I}}$ is aggregated through the weighted average

$$\boldsymbol{\mathcal{I}}_{\mathrm{sgt}}^{\mathtt{h},\mathtt{w}} = \left| \left(\prod_{i=1}^{|\Omega|} (\Omega_{\boldsymbol{\Psi}_j, \boldsymbol{\iota}_i, \sigma^2_{\mathtt{DoG}}}(\mathtt{h}, \mathtt{w}))^{\omega_i} \right)^{1 \Big/ \sum_{i=1}^{|\Omega|} \omega_i} \right|_{\mathrm{T}} \tag{5.7}$$

$$\text{with} \quad \omega_i = \exp -\frac{\boldsymbol{\Psi}_j^2}{2\kappa}, \quad \kappa = \frac{1}{3} \max_{i \in \{1 \ldots |\Omega|\}} \{\boldsymbol{\iota}_i\} ,$$

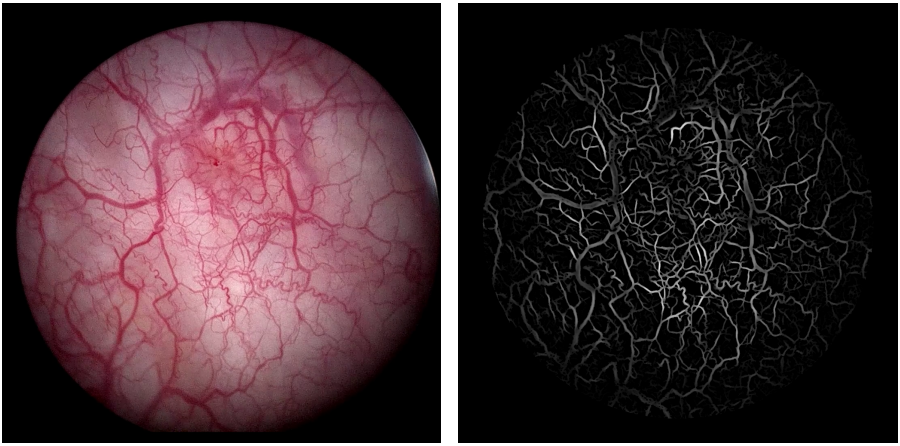
by respecting the DoG filter design (5.2) and the pixel intensity search (5.6). The operation $|\cdot|_{\mathrm{T}}()$ thresholds the aggregation to the continuous image space between zero and one such that $\boldsymbol{\mathcal{I}}_{\mathrm{sgt}}^{\mathtt{h},\mathtt{w}} \in [0, 1]$. Furthermore, the respective segmented pixel intensity $\boldsymbol{\mathcal{I}}_{\mathrm{sgt}}^{\mathtt{h},\mathtt{w}}$ determines if a pixel at position $(\mathtt{h}, \mathtt{w})$ constitutes a bar structure in the filtered image space $\boldsymbol{\mathcal{I}}_{\mathtt{DoG}}$. Specifically, this occurs when all shifted responses in (5.7) are greater than zero along the direction with maximum intensity response in (5.6). However, as the individual DoG filters move farther from the kernel center, their contribution to the overall signal decreases

Table 5.1: Filter parameters.

parameter	value range
$\boldsymbol{\iota}$	$[0,\ 7.5]$
$\boldsymbol{\Psi}$	$\left[-\frac{\pi}{4},\ \frac{\pi}{4}\right]$
$\sigma^2_{\mathtt{DoG}}$	0.6

due to respective weighting. The segmentation filter only activates when all DoG filters are active, creating a smooth transition at blood vessel ends. Due to the multiplicative concatenation in the aggregation (5.7), the segmentation is continuously attenuated at a vessel end.

Figure 5.10b shows cystoscopic images that are processed using the described filter design. The blood vessels of the urinary bladder are detected and highlighted, demonstrating a strong detection quality. The filter parameters were determined through an iterative procedure, and the resulting values are listed in Table 5.1. In order to achieve the objectives of further pattern-based reconstruction, it is essential that large, widely distributed blood vessels are accurately segmented and sharply imaged at this stage of the process.



(a) Cystoscopic input image $\boldsymbol{\mathcal{I}}$ after masking and rescaling.

(b) Segmented vascular structures in the segmented image $\boldsymbol{\mathcal{I}}_{\mathrm{sgt}}$.

Figure 5.10: Segmented image $\boldsymbol{\mathcal{I}}_{\mathrm{sgt}}$, where each greyscale pixel intensity corresponds to the confidence level of the pixel $(\mathbf{h}, \mathbf{w})$ belonging to a vessel structure in the input image.

5.2.2 Histogram Equalization of the Image

The intensity of a segmented image indicates the level of confidence in the segmentation result, indicating whether a pixel is part of a bar structure (i.e., a vascular structure) or not. To ascertain the discrete shape information, it is necessary to binarize the continuous classification intensities. A general threshold function may be employed for this purpose. Consequently, pixels with intensity values surpassing a specified threshold are classified as part of the structure, while those with intensity

values below would be classified as background pixels. However, the pixel intensities can significantly vary from one image to another (e.g., due to different exposures), so a single global threshold is unsuitable for evaluating all images. Instead, a dynamic threshold adjustment is required that adapts to each image individually. The core idea of automated threshold adjustment is to minimize the binarization error that may occur when pixels are incorrectly categorized. Following [87], an adaptive binary categorization concept is formulated by maximizing the variance between the two classes and minimizing the variance within each category.

In the initial stage, the image intensities of $\boldsymbol{I}_{\text{sgt}}$ are considered in an equivalent integer pixel grayscale color space representation $\boldsymbol{I}_{\text{int}}^{\mathtt{h},\mathtt{w}}$ with an intensity range of $[0, 255]$. As introduced in Section 2.2.5.1, there is an equivalent integer representation in the range of $[0, 255]$ for the continuous intensity values of $[0, 1]$, which is primarily considered in this work. The integer representation allows for constructing a histogram of sorted intensity values $\boldsymbol{I}_{\text{int}}^{\mathtt{h},\mathtt{w}}$ for all pixels $(\mathtt{h}, \mathtt{w}) \in \mathbb{H} \times \mathbb{W}$.

Subsequently, the so-called Otsu threshold τ_{otsu} following the work [87] of Nobuyuki Otsu is determined by calculating the histogram of the input image and the corresponding probabilities for each intensity level, $\boldsymbol{I}_{\text{int}}^{\mathtt{h},\mathtt{w}} \in [0, 1, 2, ..., L-1]$, where L represents the number of intensity levels. An example of this can be seen in the histogram plot shown in Figure 5.11, which displays the distribution of segmented pixel intensities from the sample in Figure 5.10b sorted in ascending order. Next, the algorithm identifies the optimal threshold value $\tau_{\text{otsu}} = \tau^{\star}$, which maximizes the

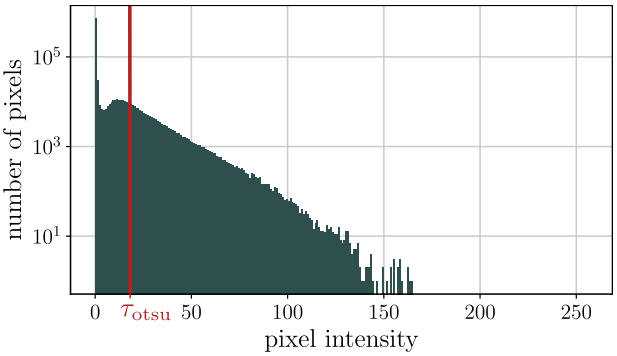


Figure 5.11: Histogram of pixel intensities in the image, with the Otsu threshold indicated by a vertical red line. The Otsu threshold separates the image into two classes based on pixel intensity, with all pixels above the threshold representing vascular structures and all pixels below the threshold belonging to the background. In the resulting segmented image, non-vascular areas are black, which results in a high number of black pixels.

between-class variance

$$\sigma_b^2(\tau) = \omega_0(\tau)\omega_1(\tau)(\mu_0(\tau) - \mu_1(\tau))^2. \tag{5.8}$$

Therein, $\omega_0(\tau)$ and $\omega_1(\tau)$ denote the probabilities that the intensity, $\boldsymbol{\mathcal{I}}_{\text{int}}^{\mathtt{h},\mathtt{w}}$, of pixel $(\mathtt{h},\mathtt{w})$ is either smaller or larger than the threshold value τ, respectively. Additionally, $\mu_0(\tau)$ and $\mu_1(\tau)$ represent the average intensities of the two classes, which depend on the threshold value τ.

Finally, the Otsu threshold τ_{otsu} is then determined by the parameterization of τ that maximizes the variance $\sigma_b^2(\tau)$ between the structure and background pixels to achieve an optimal threshold classification for the given pixel set. This facilitates an optimal adaptation of the segmentation to the conditions of the individual image data. For the example considered in Figure 5.10b, the corresponding thresholded image pair is shown in Figure 5.12a. The vascular thicknesses are classified contiguously across the image matrix, while non-essential areas are classified as background. Nevertheless, due to the binary thresholding, interruptions of vascular structures can be observed that are actually expected to be contiguous. This issue is further addressed in Section 5.4 in the context of the data-driven graph extraction process.

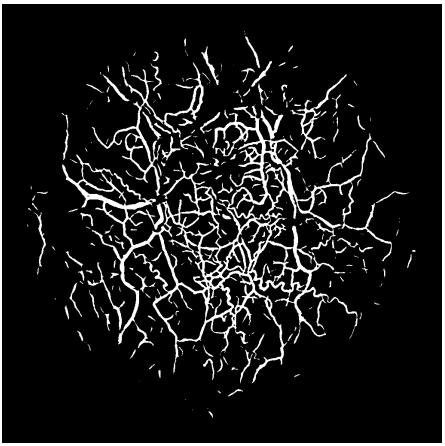
5.2.3 Structure Skeletonization

In order to extract meaningful features from the binarized structures, it is necessary to reduce the structures to a one-pixel-thin line, a process known as skeletonization or thinning in literature [63]. The skeletonization problem has been approached in various ways, including the wave expansion method presented in [11]. This method starts at the structure boundaries of the object and propagates inward to define the skeleton structure along the line of coincident wavefronts. However, the analytic formulation of this method results in a computationally expensive optimization process that can be impractical to solve in practice.

In addition to the analytical approach, there exist various unsupervised thinning methods that iteratively sample the shape from the boundary and progressively remove selected pixels until only a single-pixel thin skeleton remains. These methods are often very similar to the optimal analytical solution [67].

In general, iterative thinning methods analyze the Figure Eight-Neighborhood $\mathcal{N}_8(\mathtt{h},\mathtt{w})$ of each pixel $(\mathtt{h},\mathtt{w})$ to determine whether it should be removed or kept as part of the skeleton structure. The Figure Eight Neighborhood $\mathcal{N}_8(\mathtt{h},\mathtt{w})$ of a pixel $(\mathtt{h},\mathtt{w})$ covers all surrounding pixels $(\mathtt{h},\mathtt{w})_{\text{adj}}$ that are directly adjacent to the central pixel $(\mathtt{h},\mathtt{w})$ within a 3×3 image pixel raster [33]. To simplify the process and achieve an optimal skeleton solution that closely approximates the analytical model, an iterative thinning approach is followed in this work.

Applying the iterative thinning algorithm to the binarized image produces the skeletonized image depicted in Figure 5.12b. Alongside the more prominent vascular



(a) Threshold binarization.

(b) Skeletonized image with inverted pixel intensity space for improved visualization.

Figure 5.12: The intraoperative image shown in Figure 5.10b is segmented and binarized using Otsu thresholding. The resulting thresholded image is then skeletonized, which involves thinning all connected structures in the thresholded image to a one-pixel width line structure. Note, the skeletonized image is presented with inverted pixel intensities so that the black background is white and the white structures are shown in black to enhance their visibility for presentation purposes.

structures, there are some isolated elements composed of only few pixels. However, these sparse structures have limited informational value due to their poor connectivity and unreliable recognition potential. Furthermore, tissue deformation or camera perspective shifts can cause them to disappear, which makes them unsuitable as reliable landmark features.

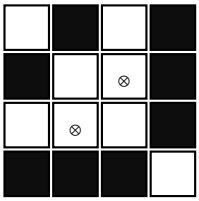
5.2.4 Pruning of Isolated Pixel Clusters

Clear pixel orderings are necessary to extract a graph from a skeletonized image by pre-defined rules. Additionally, any resulting ambiguous pixel clusters after skeletonization must be identified and resolved to represent the skeletonized image in a clear and well-defined form that is suitable for graph extraction [54, 154].

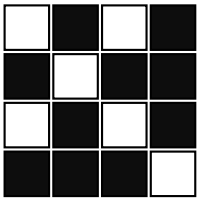
Figure 5.13 displays a critical pixel cluster pattern at a skeletonized vascular intersection, where it is unclear which pixels should be classified as crossing pixels. Therefore, the thinning process needs to be refined to establish unambiguous criteria for extracting node features based on their pixel neighborhood. Ambiguous pixel patterns after skeletonization have been widely recognized in the literature and dis-

cussed in various publications [54, 67, 141, 150, 151, 154]. An iterative selection procedure to remove ambiguous pixel patterns from skeletonized fingerprint images is presented in [154].

Following the work of [152], all segmented pixels are individually scanned by checking their respective Figure-Four neighborhood. The Figure-Four Neighborhood $\mathcal{N}_4((\mathtt{h},\mathtt{w}))$ of a pixel $(\mathtt{h},\mathtt{w})$ covers all surrounding pixels that are adjacent to the central pixel $(\mathtt{h},\mathtt{w})$, excluding the diagonal neighborhood in the respective 3×3 image raster [33, 54, 152]. A pixel is considered ambiguous if there are more than two white pixels in its Figure-Four neighborhood. In the example shown in Figure 5.13, the pixels labeled as ⊗ are considered ambiguous because they have more than three neighboring white pixels within their neighborhood of four. Iteratively, these pixels are checked and then either removed or declared distinct [152].



(a) Pixel accumulation at a vessel junction with defective pixels marked as ⊗.



(b) Clear structures after eliminating defective pixels.

Figure 5.13: Image clusters at pixel level. In (a) Pixel accumulation at a vessel junction with defective pixels marked as ⊗. The clear structures at the intersection of two curve structures after eliminating defective pixels are presented in (b) .

Therefore, the pattern is iteratively adjusted by removing a particular erroneous pixel and re-checking the remaining listed ambiguous pixels in its $\mathcal{N}_4(\mathtt{h},\mathtt{w})$ neighborhood. If the neighborhood of a particular pixel becomes unique with less than two white pixels after the removal of the erroneous pixel, that pixel is removed from the list of ambiguous pixels[152]. This delete and re-check procedure is performed iteratively until no more ambiguous pixels can be identified. Through this process, a clear skeleton structure is established.

5.3 Graph Extraction for Landmark Representation

For graph extraction, graph nodes are identified by scanning the skeletonized and pre-processed structure information for specific patterns. In addition, corresponding edge information is derived by tracing the skeletonized pixel paths.

For notation, a simple graph $\mathcal{G}(\mathbf{E}, \mathbf{V})$ can be defined as a structure that consists of nodes $\mathbf{V} = \{n_i\}$ and edges $\mathbf{E} = \{e_k\}$. In this work, only undirected graphs are considered, where the direction of the edges $\mathbf{E}$ is insignificant. The matrix formulation of the graph not only provides a structured data representation, but also enables various matrix operations for the analysis and correlation of pattern information. The following Section 5.3.1 presents the node extraction process, followed by the edge extraction process in Section 5.3.2. These pattern extraction methods are essential to construct the graph data, which is then represented in Section 5.3.3. This aids in addressing ambiguities like node sequencing and parallel edges.

5.3.1 Node Point Extraction

In this work, graph nodes are categorized as either end nodes or crossing nodes. End nodes are located at pixels where the vascular structures terminate, while crossing nodes are located at pixels where two or more structure paths intersect. To determine the node classes, i.e., endpoints and intersections, the $\mathcal{N}_8(\mathtt{h}, \mathtt{w})$ neighborhood for each pixel in the selected pixel set is examined for concise patterns. Following the proposed algorithm in [54], a pixel $(\mathtt{h}, \mathtt{w})$ can be uniquely classified if one of the following pattern structures appears in its neighborhood $\mathcal{N}_8(\mathtt{h}, \mathtt{w})$:

- One white neighbor pixel identifies a structure end.
- Two neighboring skeleton pixels identify edge structures.
- If there are exactly three white neighboring pixels, a crossing node can be identified.

If any of the described patterns is present in the $\mathcal{N}_8$ neighborhood of a pixel, it can be confidently classified as an edge pixel, an end pixel, or a crossing pixel. For example, Figure 5.14 shows a vessel end at the pixel level. The pixel labeled as $\otimes$ contains a single white pixel in its $\mathcal{N}_8$, allowing the pixel to be classified as an end node.

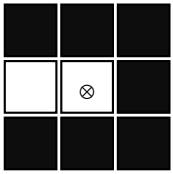


Figure 5.14: Structure end at pixel level, where the corresponding end node pixel is labeled as $\otimes$.

However, there are cases where there are more than three white neighboring pixels in the pre-cleaned skeleton structure. This occurs because the skeleton structure cannot

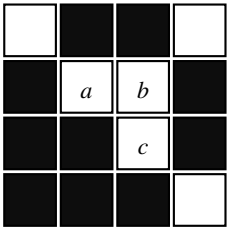


Figure 5.15: Node extraction at pixel level. Each of the pixels a, b, and c have three white pixels in their $\mathcal{N}_8$ neighborhood. According to the predefined node extraction rules, pixel b is selected a graph node.

compensate for all configurations, as attempting to accommodate every configuration would result in broken structure segments. Figure 5.15 presents an example of this, depicting a four-part bifurcation at the pixel level. Pixels a, b, and c could potentially be classified as intersection nodes because they each have three white pixels in their $\mathcal{N}_8$ neighborhood. However, an additional condition must be met for them to be accurately identified as crossing nodes. Specifically, the neighboring pixels with three white pixels in their $\mathcal{N}_8$ neighborhood must not be connected by their $\mathcal{N}_4$ neighborhood. As a result, pixels a and c cannot be classified as nodes since they are connected to pixels b and a, respectively, by their $\mathcal{N}_4$ neighborhood. The same is valid for pixel c, which is connected to pixels a and b by its $\mathcal{N}_4$ neighborhood. Only pixel b satisfies this condition and can be properly identified as a crossing node.

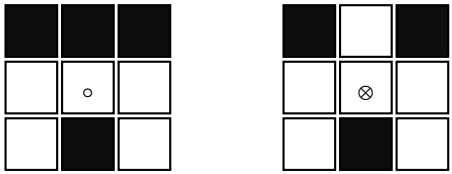


Figure 5.16: Central and right column: central node with more than three white pixels in the corresponding figure-eight neighborhood. The pixel marked as ∘ should not be identified as a graph node, while the pixel marked as ⊗ can be uniquely identified as a crossing node.

To accurately identify node features in an image, it is necessary to impose an additional constraint on pixels with more than three white pixels in their $\mathcal{N}_8$ neighborhood. Such a pixel must have at least one neighboring pixel that is not connected to any other neighbor in the $\mathcal{N}_4$ neighborhood to be classified as a node [154]. This is because the presence of a high number of white pixels in the $\mathcal{N}_8$ neighborhood does not necessarily indicate a node. It could simply be a strongly bent curve struc-

ture. For example, consider Figure 5.16. Without the additional constraint, the pixel labeled ⊗ may not be accurately classified as a node due to the multiple white pixels in its $\mathcal{N}_8$ neighborhood. However, the presence of an isolated white pixel in the $\mathcal{N}_4$ neighborhood enables the pixel to be confidently classified as a node when the referred constraint as propsped in[154] is applied.

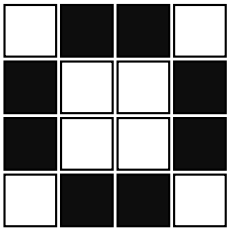


Figure 5.17: Ambiguous structural intersections, wherefore additional criteria are needed.

Furthermore, as shown in Figure 5.17, there are specific structures for which none of the criteria for node classification described above apply. For these fairly rare occasions, further small-step queries would be required to provide an unambiguous classification. Nevertheless, despite the limitation of the scope of the pattern queries, the vast majority are correctly detected, and only rare outliers are observed that are misclassified. Figure 5.18 presents the node point extraction applied to the skeletonized image shown in Figure 5.12b. The end noes are depicted in ● red and crossing nodes are depicted in ● blue.

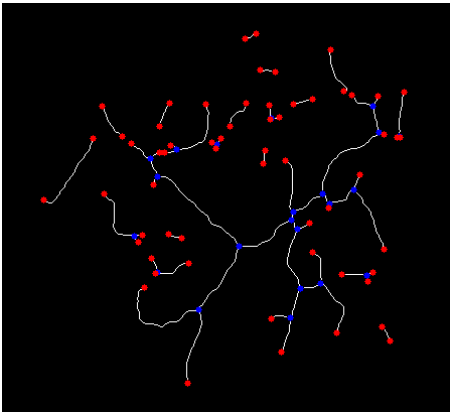


Figure 5.18: Node classification: endpoints in ● red and crossing nodes in ● blue.

5.3.2 Edge Extraction

To build the graph, it is essential to determine all corresponding edge pixels. Starting from the extracted node pixel positions $\boldsymbol{p}_n$, the corresponding end nodes of the edges are traced through the eighth neighborhood along the skeletonized structure. The default pattern along the structure is characterized by two pixels $(\mathtt{h}, \mathtt{w})$ and $(\mathtt{h}, \mathtt{w})_{\text{adj}}$, where $(\mathtt{h}, \mathtt{w})_{\text{adj}} \in \mathcal{N}_8(\mathtt{h}, \mathtt{w})$. These two pixels can be traversed along the skeletonized structure by following the sequence $(\mathtt{h}, \mathtt{w}) \rightarrow (\mathtt{h}, \mathtt{w})_{\text{adj}}$ to iteratively determine the corresponding end node $(\mathtt{h}, \mathtt{w}) \rightarrow (\mathtt{h}, \mathtt{w})_{\text{end}}$. However, when dealing with more complex structures, it becomes necessary to remove pixels that have already been visited from the tracing space to avoid revisiting them.

The exploration process ends when all possible path structures have been thoroughly traversed, starting at individual pixel nodes $(\mathtt{h}, \mathtt{w})_n \in \boldsymbol{p}_{\mathbf{V}}$ and ending at their respective end nodes $(\mathtt{h}, \mathtt{w})_{n \rightarrow \mathsf{adj}(n)}$. Next, the edges connected to a specific node $(\mathtt{h}, \mathtt{w})_n$ are analyzed, and the graph information is synthesized with respect to its adjacent node and edge information. The corresponding pixel set $\boldsymbol{p}_e$ is then integrated into the feature space for each edge e, allowing for lightweight pattern analysis through the extracted graph representation. If necessary, in-depth analysis of the structural information can be performed using the corresponding pixel set $\boldsymbol{p}_e$ linked to the related edge e.

5.3.3 Adjacency-based Graph Representation

Based on the identified node and the edge information, the respective data can be unified in an adjacency matrix representation. The square adjacency matrix $\mathbf{A} \in \mathbb{Z}^{n \times n}$, with n as the number of nodes in the graph, maps the edge information $\mathbf{E}$ into its respective matrix entries. Thereby, the matrix row i corresponds to the i-th node in the graph, and the entry a_{ij} in that particular row refers to the number of edges connecting node i to node j. Since the structure only allows for the description of undirected edge information, the entry a_{ij} must be equal to a_{ji}, and as a result, the adjacency matrix $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$ is symmetric. The adjacency matrix representation allows for a distinguishable but deformation-invariant data representation, as demonstrated in the graph representation in Figure 5.19. The node and edge information of the graph encode deformation-invariant properties by preserving the interconnection structure of the nodes, irrespective of

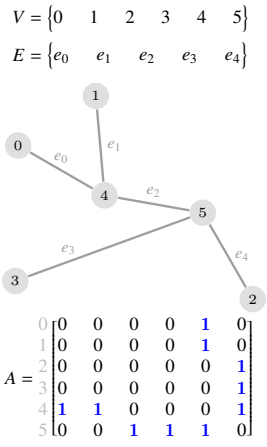


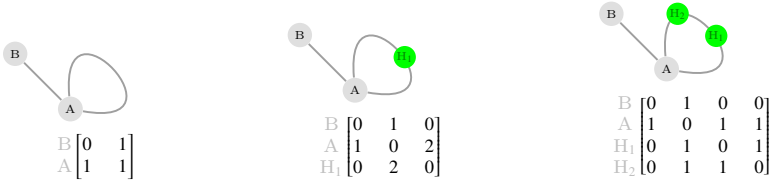
Figure 5.19: Example of a simple graph with undirected edges, Graph **A**.

any spatial deformation that may occur, provided that the node arrangement is not disrupted.

A loop pattern is formed when edges have the same start and end node. This pattern can create difficulties in graph-based representations, making it challenging to extract edge attributes and leading to ambiguities in the matrix representation. As a result, graph matching may be hindered.

Graph-based representations may encounter issues when handling edges that form loops, meaning structures that have the same start and end node. Extracting edge attributes from these edges can create ambiguities in the matrix representation, which can hinder the process of graph matching. This is evident in the adjacency matrix of Figure 5.20a, where nonzero entries on the diagonal indicate the presence of loop structures. To create an unambiguous representation for feature extraction and graph matching, auxiliary nodes are inserted into the loop structures. The original looping edge is replaced by two parallel edges between the original node and an auxiliary node, as shown in Figure 5.20b. This process results in the insertion of an auxiliary node at the center of the original loop, and the extension of the adjacency matrix with a new column and row containing only zeros on the diagonal.

However, the creation of multiple adjacencies between the node pair through the use of parallel edges can be problematic as it lacks a clear correspondence between the entries in the adjacency matrix and the actual edges in the graph, which is crucial for graph matching. To address this issue, another auxiliary node, as shown in Figure 5.20c, is inserted to ensure an unambiguous representation of the corresponding structure. This insertion extends the adjacency matrix with another row and column and eliminates loop structures and parallel edges. As a result, the graph data is exclusively represented by binary-valued adjacency matrices with zeros on the diagonal, enabling unique constraints on graph attributes and facilitating graph matching.



(a) Graph with a loop. **(b)** Graph with parallel edges. **(c)** Graph with helper nodes.

Figure 5.20: Loops and parallel edges in this graph create problematic structures that can hinder analysis and interpretation. Therefore, helper nodes are introduced to break up self-adjacency and to retain the adjacency matrix as a binary-valued matrix.

Based on the graph extraction presented in this section, the resulting graph extraction for the initial image observation shown in Figure 5.10a, processed into the

skeletonized image shown in Figure 5.12, is presented in Figure 5.21. The graph representation facilitates a distinguishable and deformation-invariant data representation. However, for the shown example, there are several interruptions in the graph pattern that do not match the visible vascular pattern courses in the underlying input image.

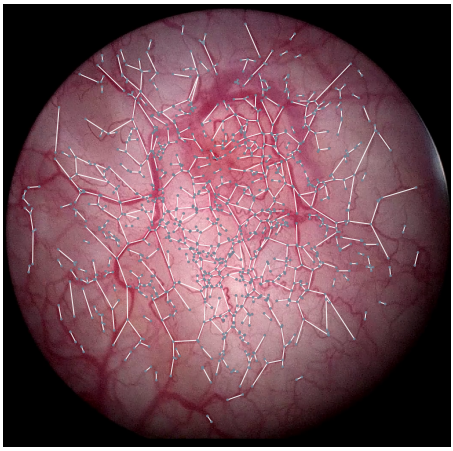


Figure 5.21: The extracted graph overlaid on the original input image observation $\boldsymbol{I}$. The graph structure represents the main courses of the visible vascular structures.

5.4 Data Driven Edge Extraction

The graph extraction process described previously is guided by well-defined rules and procedures to detect vascular structures and patterns accurately. The resulting graph should be consistent, regardless of the perspective or conditions of the image capture from which the structures are extracted. To achieve this, a template pattern-based recognition is applied to skeletonized image data, enabling a well-defined identification of both node points and edge information. However, image processing can lead to uncertainty in the process, as slight changes in lighting can cause variations in the representation of vascular structures. This uncertainty is further compounded by the binarized filter response, which can lead to interruptions along edge structures at different locations, as illustrated in Figure 5.22. To mitigate this problem, a data-driven edge extraction network (EdgeNN)is proposed to limit the interruption of edges, ensuring that the extracted graph is robust and that similar images consistently yield similar or identical graphs. The given task is to determine the adjacency matrix based on the skeletonized and filtered image and the extracted nodes as input data. However, addressing varying dimensions problems

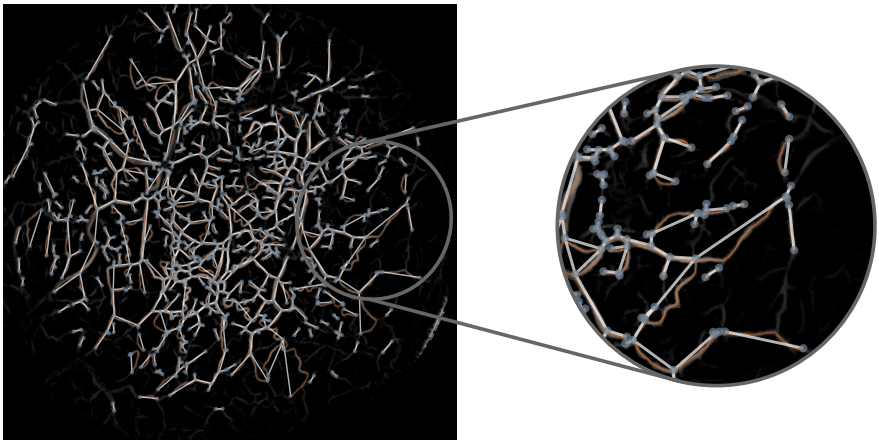


Figure 5.22: The extracted graph is superimposed onto the filtered image, with the skeletonized structures highlighted in orange. In the close-up view, it is apparent that the skeletonization and graph extraction processes are interrupted at various vascular structures that may appear to be connected when analyzing the filtered image data.

of varying node numbers among the respective images is highly challenging as the network architecture is, in general, fixed in its dimension and still needs to adapt to different input sizes while maintaining high accuracy. This can be difficult to achieve with a fixed architecture, as the model needs to handle unseen combinations of nodes and edges during training, leading to poor performance and difficulty in generalizing to new data.

The problem of handling varying dimensionality in neural networks is often addressed through a 'fixed-sized' method, where input and output data are zero-padded to fit a matrix of constant dimensions. However, this approach can be computationally expensive and require significant storage overhead, potentially losing the advantage of fast prediction times. In contrast, this work employs a 'divide and conquer' approach to address the problem of varying dimensionality in neural networks. Therefore, the graph prediction problem is reduced to the problem of predicting the existence of an edge between a single pair of nodes first, and the overall graph prediction is then built based on a cumulative edge prediction scheme. Therefore, an effective concatenation-based network architecture is exploited. Training and validation are performed on a consistent, balanced dataset to ensure robust and generalizable results. To predict the complete graph, a set of single-edge predictions are made for a batch of potential node pair combinations, which are then used to construct the final adjacency matrix. This approach effectively handles the varying dimensional classification problem by reducing it to a fixed dimension problem, thus providing a

generalizable solution for any graph dimension.

In addition, this approach enables fast and real-time latency evaluations, taking advantage of the 'embarrassingly parallel' nature of neural networks. Using neural networks on graphics processing units (GPUs) or other high-performance architectures such as field-programmable gate arrays (FPGAs) can easily and quickly carry out classification tasks, potentially outperforming the current process of extracting a graph from an endoscopy image, which involves several computationally intensive steps to identify the individual nodes and edges of the graph.

In the following Section 5.4.1, the design of the network architecture, including the data format, the generation of the training data, and the training, is presented. In Section 5.4.2, the comprehensive data-driven graph prediction is presented by introducing a combination scheme for simultaneous edge prediction. Finally, the results are evaluated in Section 5.4.3.

5.4.1 Network Design

The detection of suspicious broken structures in dependent courses is addressed through the use of a data-driven edge extraction network. As a prerequisite, preprocessed data including filtered and skeletonized images are considered as input data, in addition to extracted node positions.

The simultaneous use of skeletonized and filtered images enables the extraction of high-level information. The skeletonized image reveals the main structures, while the filtered image provides the necessary context to analyze the reliability of questionable boundaries. The network design combines two distinct output predictions to reveal ambiguous edge patterns. The first class, y_{skel}, classifies whether there is an edge structure between two node points based on the pixel-based edge extraction method described in Section 5.3.2. Including the graph pattern labels of the pixel-based edge extraction encourages the network to focus on the main patterns accurately.

Additionally, a second output class y_{sgt}, is implemented to enable the network to generalize based on the given context. The y_{sgt} class considers both the filtered image context and the skeletonized structure to detect interconnected structures, even when the skeletonized structure is interrupted along the course of the vascular structure. The goal is to replicate the prior pixel skeleton-based outcomes in y_{skel}, while minimizing any uncertainty in y_{sgt}. Incorporating a diversified data representation, as illustrated in Figure 5.23, enhances the network's generalizability and robustness.

5.4.1.1 Input/Output Data

The input data is represented in a multi-channel format with dimensions $256 \times 256 \times 4$. This encompasses: the segmented image $\boldsymbol{I}_{\text{sgt}}$, the skeletonized image $\boldsymbol{I}_{\text{skel}}$, the

node positions, $\boldsymbol{I}_p$, and a sparse binary channel $\boldsymbol{I}_c$. Notably, $\boldsymbol{I}_c$ marks the (x, y) coordinates of the node pair to predict, with its only non-zero entries indicating these locations. The outputs from the node extraction network serve as inputs to the edge prediction network. This is complemented by the inclusion of an additional input channel, an image matrix $\boldsymbol{I}_c$, which encodes the specific node pair combination whose neighborhood is to be predicted.

The skeletonized data offers insights into the evolution of structures based on the prior extraction procedure. Furthermore, filtered data aids in discerning whether trajectories are connected, distinct, or mistakenly recognized as separate due to binary Otsu thresholding and skeletonization.Taking into account the full context given by the segmented image data $\boldsymbol{I}_{\text{sgt}}$ enhances the ability to identify trajectories as connected.

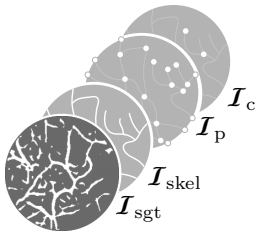


Figure 5.23: The input image $\{\boldsymbol{I}_{\text{sgt}}, \boldsymbol{I}_{\text{skel}}, \boldsymbol{I}_{\text{p}}, \boldsymbol{I}_{\text{c}}\}$ to the edge extraction network consists of a skeletonized image, the corresponding node positions, and a sparse image matrix indicating the node pair for which the adjacency is to be predicted.

A tuple $[y_{\text{skel}}, y_{\text{sgt}}]$ is designed to serve as an output, summarizing the active input nodes in the image $\boldsymbol{I}_{\text{c}}$ and indicating the presence of potentially broken adjacency edges based on the comparison between the skeletonized and filtered images. Specifically, y_{skel} is intended to be consistent with the results obtained from traditional methods utilizing skeletonization images. Additionally, y_{sgt} is designed to predict the validity of the edge information, determining if it is broken based on the skeletonized image but recognized as continuous based on the filtered output image y_{skel}. Where respectively, each output $[y_{\text{sgt}}, y_{\text{skel}}]$ is a binary value, either 0 or 1, corresponding to a prediction of a `non-edge` or an `edge` respectively. The encoding details for these outputs are provided in Table 5.2.

Table 5.2: Edge prediction cases.

Name	Description
`GOOD`	`node_adjacencies == node_degrees`
`OK`	`node_adjacencies < node_degrees`
`BAD`	`node_adjacencies > node_degrees`

In the preliminary design phase of the proposed network, it was observed that keeping the node positions $\boldsymbol{I}_{\text{p}}$ as input to the EdgeNN is critical for the prediction task. This is particularly relevant

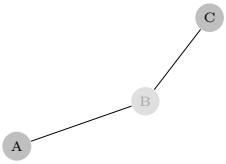


Figure 5.24: A node pair A-C that is potentially problematic during edge prediction. Nodes A and C are not directly connected, and therefore, there is no adjacency between them.

when edge prediction is performed on a pair of nodes such as the one shown in Figure 5.24. In this example, the node pair is connected by a skeletonized structure, but, according to the graph notion of A-C, no adjacency relation should be predicted for this pair, in order to determine the adjacency matrix according to the pixel-based edge extraction established in Section 5.3.2. Instead, the node pair is only indirectly connected to a common node B. Based on the node positions $\boldsymbol{\mathcal{I}}_{\mathrm{p}}$, the network has information about the distribution and existence of all other nodes, including the intervening node B, which prevents the false assumption of adjacency due to the visible skeletonized structure passing over node B.

5.4.1.2 Network Architecture

The network architecture is based on a modified version of the VGG-16 neural network. The VGG-16 neural network, proposed in 2014 by the Visual Geometry Group at the University of Oxford, is a widely used convolutional neural network architecture for image classification tasks. It is characterized by its use of multiple convolutional and max pooling layers to extract features from an image and its final layers perform classification based on those features. For a more in-depth understanding of the VGG architecture, a general overview is provided in the Appendix A.3. The design of the proposed EdgeNN builds on this architecture by utilizing a modified

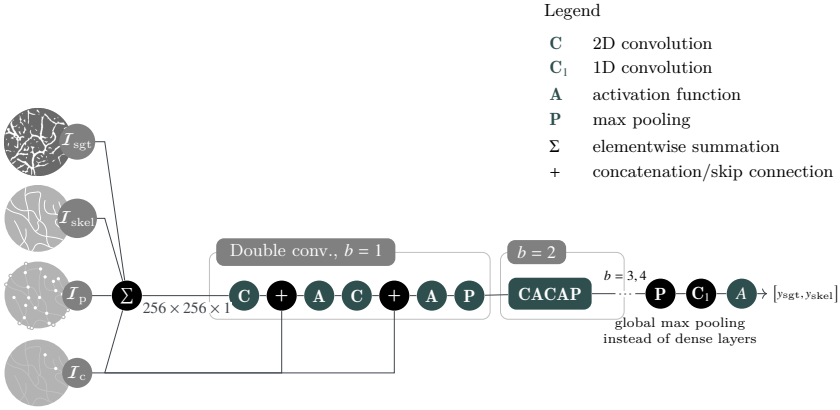


Figure 5.25: The proposed EdgeNN architecture is a modified version of VGG-16 [121], where the modifications are depicted as black nodes. The input data, shown on the left, is a 3-channel image $\left[\boldsymbol{\mathcal{I}}_{\mathrm{sgt}},\ \boldsymbol{\mathcal{I}}_{\mathrm{skel}},\ \boldsymbol{\mathcal{I}}_{p},\ \boldsymbol{\mathcal{I}}_{c}\right]$. The network has four convolution blocks in total, $b = 1, \ldots, 4$. The main modifications occur in the first convolution block $b = 1$. The dense layers in the original VGG-16 are replaced with a global max pooling layer and a 1D convolution.

version of the VGG-16 to better suit the specific conditions and problems of edge extraction. The proposed architecture of the EdgeNN is illustrated in Figure 5.25.

The EdgeNN is made up of four convolutional blocks, with the first two containing two convolutional layers, and the last two containing three convolutional layers. The initial block, $b = 1$, has a number of convolutional filters, $f_1 = 6$. Each subsequent block, b, makes use of $f_b = f_1 \cdot 2^{b-1}$ kernel in the convolutional layers. To ensure preservation of information, particularly in the sparse input channel containing pixels of the $\boldsymbol{\mathcal{I}}_c$ node pair, the input layers are summed element by element to form a $256 \times 256 \times 1$ matrix before being passed through the convolutional layers. To obtain the node pair information, the node pair channel is added to the end of the convolution outputs of the first double convolution block.

5.4.1.3 Data Generation

The training data for the first output, denoted as y_{skel}, is derived from the pixel-based graph extraction method discussed in Section 5.3.2. This technique facilitates automated labeling of any video material sourced from endoscopy for y_{skel}. It is noteworthy that the ground truth for y_{sgt} inherits the limitations of the extraction algorithm and is influenced by the sensitivity of the prevailing conditions.

Figure 5.26: Artificial test sphere.

To procure ground truth data for the second output, y_{sgt}—aimed at categorizing unwanted interruptions and cohesive edges—the following strategies are employed:

(i) Data undergoes manual scrutiny to rectify any undesirable breaks in the adjacency design, wherein the output y_{sgt} is manually set.

(ii) To reduce the extent of manual data inspection, synthetic interruptions are infused into the skeletonized data. This is achieved by arbitrarily choosing white pixels and adjusting both the chosen pixel and its adjacent pixels to zero. The magnitude of this alteration is controlled by a hyperparameter, selectable as a random integer up to a specified cap, ensuring the dataset's relevance to the problem at hand. Given the synthetic nature of these changes, the ground truth is inherently extracted from the original graph structure, inclusive of all modifications.

(iii) A prime dataset is crafted from a fabricated test setting, depicted in Figure 5.26. This environment emulates vascular structures, and the corresponding skeletonized pattern can be reliably extracted without any discernible interruptions.

By combining these approaches for data generation, the risk of missed or ambiguous inconsistencies in the manually reviewed dataset—such as undetected outliers

leading to incorrect labeling—is minimized. This reduction in inconsistencies enables the network architecture to effectively generalize and renders the presence of outliers insignificant.

5.4.1.4 Training

Typically, only a small number of possible node combinations are actually connected, resulting in imbalanced training data between connected and unconnected node pairs. To enhance the model's generalization, the training data is balanced by pre-selecting an equal number of connected (`True`) and unconnected (`False`) node samples in each epoch. In Figure 5.27, a portion of the ground truth dataset for edge connections for connected and unconnected samples is shown. Specifically, the overall dataset is imbalanced with a distribution of 80% non-edge and 20% edge pairs. In addition, during training, batch normalization is applied after each convolutional layer to normalize the weight distribution and stabilize the learning process. The model is trained using the Adam solver (A.2) with a learning rate of $\alpha_{\mathrm{lr}} = 1 \times 10^{-3}$. In total, the EdgeNN has 68 089 trainable parameters and is trained over 150 epochs with a batch size of eight images per epoch step. Figure 5.28 shows the training and validation losses.

The network is starting to converge from the first epoch, although the convergence rate is lower compared to the epochs ranging from 10^{th} to 50^{th} epochs. As the validation loss is consistently small and comparable to the training loss, the network can generalize well on the training data and is considered fully trained by the 80^{th} epoch. The trained model is evaluated on a test set of 486 images. Four-node pair combinations were chosen for each image in a ratio of 1:1 edge/non-edge combinations. The evaluation metrics include precision, recall, and the F_1 score, which were averaged over the 486×4 node combination pairs based on a true/false database. The precision and recall were found to be 0.997 and 0.995, respectively, and the F_1 score was 0.996. The respective metrics accuracy, precision, recall, and F_1 score - used for evaluating binary classifications - are specified in Appendix A.4.

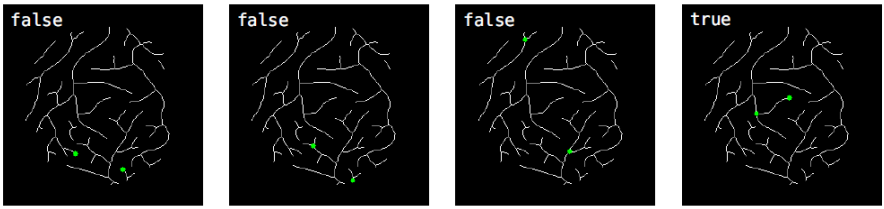
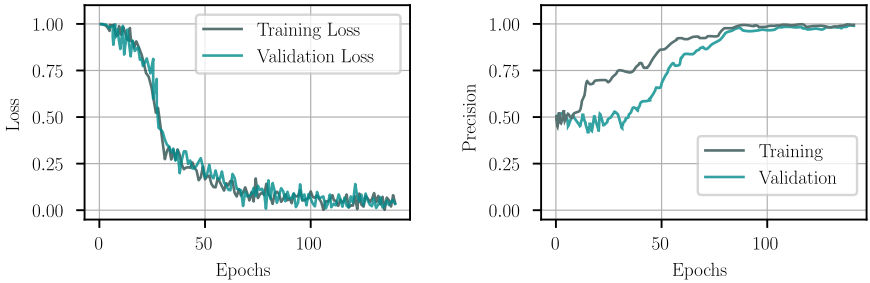


Figure 5.27: Randomly selected node pair combinations in ● green from the set of extracted graph nodes. The accompanying text provides ground truth information on whether a connection exists through the skeletonized structure or not.



(a) Training and validation losses.

(b) Training and validation precisions.

Figure 5.28: Training and validation losses of the baseline EdgeNN. Best validation precision: 0.9873 at epoch 127. Best validation loss: 0.014 at epoch 109.

5.4.2 Adjacency Combination Schemes

As the trained EdgeNN model only predicts the presence of edges between one node pair combination, multiple predictions must be consolidated to infer the overall adjacency structure of a graph via an appropriate adjacency combination scheme. This results in an adjacency matrix representing the graph's overall topology. Evaluating the complete graph in a brute-force manner involves the evaluation of all possible node pair combinations, which in general, pose a computationally expensive procedure. To address this issue, the parallelization capabilities of GPUs is leveraged to accelerate the evaluation time. However, this may still be limited by the GPU memory size, resulting in the evaluation of smaller subsets of combinations, denoted as $\mathbf{C}_k \subset \mathbf{C}$, where $\mathbf{C} = \left\{c_{ij} \mid i,j = 0,\ldots,n-1\right\}$ and $|\mathbf{C}| = n(n-1)/2$. To mitigate this limitation, a k-nearest neighbor (**knn**) search is employed to prioritize the evaluation of the most likely combinations of nodes for adjacent edges by identifying the

$$\mathbf{N}_{k_{\mathtt{EN}},i} = \mathtt{knn}(k_{\mathtt{EN}}, v_i) \tag{5.9}$$

nearest nodes for each node in the graph, reducing the number of evaluations for each edge from n to $k_{\mathtt{EN}}$. To create a pool of potential node pair combinations, the **knn** algorithm is used to identify the $k_{\mathtt{EN}}$-nearest neighbors for each node, resulting in a set $\mathbf{C}_k$. To eliminate duplicates, this set is denoted by $\mathbf{C}_{k_{\mathtt{EN}}\neq}$ with

$$\mathbf{C}_k = \bigcup_{v_i \in \mathbf{V}} \left\{\left\{v_i, v_j\right\} \mid v_j \in \mathbf{N}_{k,i}\right\} \tag{5.10a}$$

$$\mathbf{C}_{k_{\neq}} = \mathtt{unique}(\mathbf{C}_k), \quad \text{with } |\mathbf{C}_{k_{\neq}}| \leq n \cdot k \ll n(n-1)/2 \ .$$

Edge prediction for the majority of appearing edges can be efficiently performed with a single pass of the optimized CUDA implementation of the **knn** algorithm, as seen through empirical evidence.

Thus, the EdgeNN takes a batch of combinations, $\mathbf{C}_{k_{\text{EN}}} \subset \mathbf{C}$, as input and produces the corresponding batch output of edge predictions, $a(\mathbf{C}_{k_{\text{EN}}})$. Then the predictions are combined into a final $n \times n$ adjacency matrix $\mathbf{A}$, where the number of neighbors is initialized with $k_{\text{EN0}} \leq n$. To improve the comprehensive graph prediction, the algorithm iteratively searches through more combinations until all relevant ones are identified, thus improving execution accuracy.

The knn evaluation is a technique that can be used to extract edges from a graph with a high degree of accuracy while limiting the number of evaluations required. This technique works on the premise that nodes that are closest to each other are more likely to be connected. To expand the range of potential edge combinations, the algorithm is applied iteratively, with the value of k being gradually increased in each iteration.

A loop evaluation scheme is proposed as a new approach to enhance the accuracy of the evaluation process. This scheme incorporates an early stopping criterion based on pre-determined node degrees, which can be obtained by analyzing the neighboring pixels as described in Section 5.3.1. If a node's predicted edges match its pre-determined degree, it can be excluded from further evaluation.

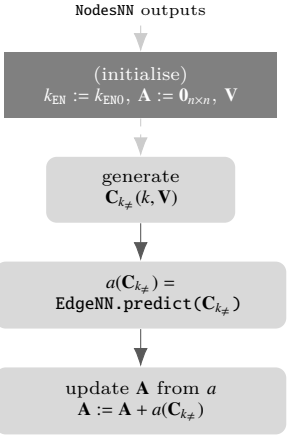


Figure 5.29: The Base Adjacency Combination (BAC) scheme for combining edge predictions over a set of node pair combinations $\mathbf{C}_{k_{\neq}}$. The prediction results on $\mathbf{C}_{k_{\neq}}$ are directly written into $\mathbf{A}$.

The algorithm operates as follows: For the current evaluation iteration, the maximum number of evaluable execution tuples is processed within the given hardware constraints. Each edge is evaluated by examining distant nodes that have not been considered as long as the edge prediction for the node does not match its pre-determined degree. If a node is found to have the same number of predicted edges as its pre-determined degree, it is excluded from further evaluation. A more detailed description of the evaluation set and termination criterion is presented in the Appendix A.5.

5.4.3 Evaluation

The prediction performance is demonstrated in Figure 5.30, where the predicted graph $\mathbf{A}$ is overlaid onto the original skeletonized structure. It can be observed that the model accurately identifies true positive adjacency entries.

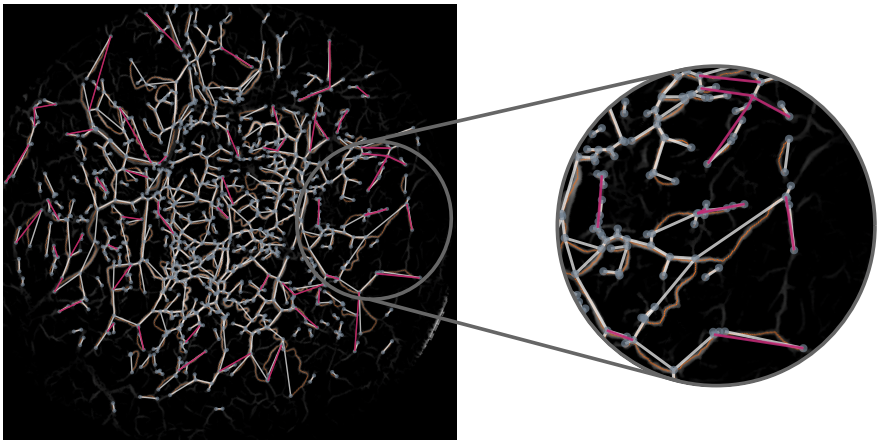


Figure 5.30: A sample prediction of the adjacency matrix based on y_{skel} and the corresponding graph adjacency perception based on y_{sgt}, where y_{skel} and y_{sgt} are not aligned with each other. The data-driven prediction exhibits a more robust graph pattern compared to the underlying filtered image data. Graph edges are considered connected, where graph patterns according to y_{skel} are interrupted. The predicted edges are shown in ● grey and in ● pink (pink lines ≙ predicted edges) overlaid on the original skeletonized structure in ● orange.

The network predicts the problematic cases with high accuracy, and is able to generate the graph with a single iteration for a small number of nodes (less than 80) and $k_{\text{EN0}} = 20$. The network evaluations are performed for a total of $n * k_{\text{EN}}$ nodes, resulting in a complete extraction of the corresponding graph. This supports the knn execution strategy, where edge overlaps caused by short-term interruptions in the skeletonized course are located close to each other and thus can be evlauted in a single iteration.

The overlaps between nodes are identified as connected graph edges. These edges are represented in red or purple, depending on the prediction values y_{skel} and y_{sgt}, respectively. When predicting the overall network, the decision based on the graph map results in low latency execution times. Table 5.3 shows the execution time complexity and accuracy for the simplest case of k_{EN0} as a function of the size of k_{EN}. The execution time increases with k, but remains highly satisfactory even for small values of $k_{\text{EN}} = 10$. For k_{EN} up to $k_{\text{EN}} = 8$, the prediction

Table 5.3: Evaluation of the simple combination scheme.

k_{EN}	Precision	Recall	Time in s
1	0.948	0.576	0.001
2	0.839	0.745	0.001
4	0.793	0.873	0.001
8	0.779	0.953	0.001

times are very small, at $t = 0.001\,\mathrm{s}$. However, as k_{EN} increases, the F_1 value decreases. This is because the precision of the prediction declines with increasing k_{EN}, while the recall improves.

In conclusion, the method of **knn** is efficient in catching problematic edge overlaps, while the execution time remains satisfactory even for small values of k_{EN}. Additionally, the graph can be extracted through edge extraction in a filter-based way and all edges are further evaluated and rechecked by the given data-driven evaluation. However, this is not the focus of further discussion as the methods are given and all implementation aspects are specific to the application.

5.5 Node and Edge Attributes

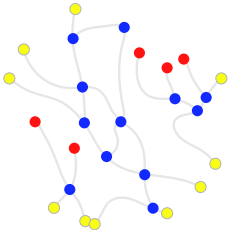
To enhance the available information for matching corresponding graph patterns, additional attributes of the observed pattern are incorporated into the graph representation. Therefore, in the following analysis, edges and node attributes are evaluated independently.

5.5.1 Node Attributes

The node attributes relevant to this work are node positions and node types. Node positions are trivially the Euclidean coordinates. For an image observation, the respective pixel coordinates $\left[\boldsymbol{p}_x^{\mathrm{h,w}}, \boldsymbol{p}_y^{\mathrm{h,w}}\right]$ are considered. The node points provide spatial ratios and are essential for graph matching, reconstruction, and localization purposes. In addition, to each node a node type is assigned, which indicates whether it is a helper node or located at an intersection, the end of a blood vessel, or the edge of the image area, as shown in Figure 5.31. This information provides a reliability estimate of the node positions, which is especially important for graph matching and reconstruction. Crossing nodes can be precisely and unambiguously discerned by their positions. However, endpoints of a blood vessel may be observed at varying surface positions for different observations of the same structure. Since blood vessels do not terminate abruptly but rather exhibit a gradual transition from a distinctly visible to a uniform surface texture, their position is subject to various influencing factors. These factors encompass the image observation's contrast, which is impacted by shadowing determined by the illumination and the endoscope's viewing direction.

As a result, crossing nodes present a greater degree of dependability concerning spatial information in comparison to end nodes. These nodes supply vital structural details, such as the direction and path of the corresponding edge. However, their spatial positions themselves are not utilized as they might be inconsistently observed over different observations. The reliability of helper nodes is primarily determined by the types of neighboring nodes and the modification of the curvature pattern,

although there is a well-defined procedure for their insertion. Therefore, the position of a helper node may be critical for matching processes but may only have moderate significance for the actual reconstruction objective.



Node types	Marker colour
-	-
END	red
CROSSING	blue
BORDER	yellow

Figure 5.31: Node type encoding.

5.5.2 Edge Attributes

To provide more descriptive structural information, additional attributes are defined for edges, including length and curvature attributes. The edge extraction process classifies the pixels of the corresponding edge structure, which can serve as raw edge attributes $\boldsymbol{p}_e$. However, these attributes are insufficient for comparing arbitrary edges in their raw form during graph matching. To address this issue, further processing is conducted to derive comparable and strongly characterizing edge attributes suitable for graph matching. The relevant edge curve length is directly determined by the number of pixels N_{px} in the corresponding edge $\boldsymbol{e}$ structure defined in the image space $\boldsymbol{\mathcal{I}}$. This eliminates the need for redundant capture of edge lengths provided by the Euclidean lengths between the respective node locations. The exploration process concludes when all possible path structures have been thoroughly traversed, starting at individual pixel nodes $(\mathtt{h},\mathtt{w})_n \in \boldsymbol{p}_{\mathbf{V}}$ and ending at their respective end nodes $(\mathtt{h},\mathtt{w})_{n\rightarrow\text{adj}(n)}$. Subsequently, the edges connected to a specific node $(\mathtt{h},\mathtt{w})_n$ are analyzed, and the graph information is synthesized with the node information. Furthermore, the corresponding pixel set $\boldsymbol{p}_e$ is integrated into the feature space of the edge information, denoted as $\boldsymbol{e}$. This combination of graph information and corresponding pixel structure enables lightweight pattern analysis through graph representation. If needed, further in-depth analysis of the structural information can then be performed based on the corresponding pixel set $\boldsymbol{p}_e$ linked to the edge.

Furthermore, a polynomial approximation is performed to encode the curvature information of the edge, while the corresponding polynomial coefficients are used as edge attributes. To ensure a unique polynomial approximation of the given set of pixels, a local coordinate system is defined with a reference node at the origin and the alternate edge node located on the x-axis. The pixel set is then rescaled so that

the target node intersects the x-axis at position $x = 1$. As a result, the curvature information becomes scale-independent. Thereby, no redundant scale information is included in the curve attributes. Spatial length information is implicitly mapped by the corresponding node positions. Finally, comparability is ensured by the reference system, which is defined according to explicit rules. In Figure 5.32, the alignment of a local coordinate system to a given edge structure is illustrated for a specific node reference. Since each node of an edge is considered once as a reference node, two polynomial approximation results are obtained for one edge. The directed attribute information provides valuable insights and enables precise similarity checks, leading to a more reliable identification of matches. This is particularly useful for line structures consisting of two nodes and an edge, where the matching task would be ambiguous due to the undirected graph structure.

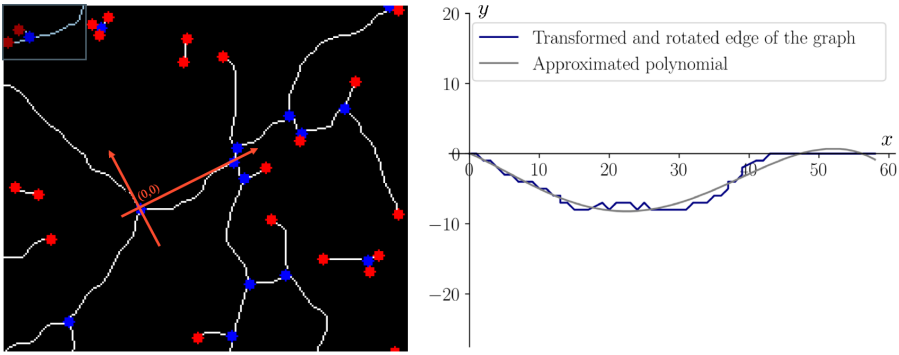


Figure 5.32: Skeletonized edge pixel, where a coordinate system is aligned through the adjacent nodes, and a polynomial is approximated to represent the course of the vascular structure.

To ensure that only strong, descriptive, and robust attributes are considered, the polynomial complexity is limited such that attribute changes are proportionate to the deformation of the edge. This approach provides reliable traceability of information. To achieve this, a polynomial approximation in the form of

$$f(x) = a_2(x - a_s)^2 + a_3(x - a_s)x^3 \tag{5.11}$$

is employed. The polynomial must be sufficiently large to accurately portray the curvature behavior. Simultaneously, its parameter range must not be overly expansive, ensuring that the polynomial coefficients' numerical values maintain their characteristic significance and contribute effectively to the graph matching process. Consequently, a third-order polynomial is utilized, as it precisely represents the curvature behavior while allowing the polynomial coefficients to contribute to the matching process. The constant offset is omitted as the curve must, by definition, intersect the origin. The coefficients a_2, a_3, and a_s are determined as curvature attributes.

The edge lengths N_{px} are assigned to the curvature parameters of the graph as edge attributes, which are summarized in an asymmetric attribute tensor $\overline{\overline{\boldsymbol{A}}}$ of the form

$$\overline{\overline{\boldsymbol{A}}} = \begin{bmatrix} 0 & c_2(v_1,v_2) & c_2(v_1,v_3) & \cdots & c_2(v_1,v_n) \\ c_2(v_2,v_1) & 0 & c_2(v_2,v_3) & \cdots & c_2(v_2,v_n) \\ \vdots & & 0 & & \vdots \\ & & & \ddots & \\ c_2(v_n,v_1) & & \cdots & c_2(v_n,v_{n-1}) & 0 \end{bmatrix} \tag{5.12}$$

for the respective edge attributes $\overline{\overline{\boldsymbol{A}}}_{ij} \in \{\mathsf{N}_{\mathrm{px}}, a_2, a_3, a_{\mathrm{s}}\}$ given in the corresponding graph.

Finally, the curve parameters, along with the node positions, provide a comprehensive representation of the vascular pattern. Figure 5.33 illustrates the superimposition of the vascular structure on the corresponding polynomial approximation for the given curve parameters and node positions. It is evident that the curve parameters effectively approximate the vascular structures.

Additional attributes are conceivable for further description of the depicted structure, such as the thickness of a vascular structure or the color information of the vascular structure (which can partially change in its red hue due to the inclusion of different tissue layers). However, these attributes are not further addressed in this work. Nonetheless, it is worth noting that incorporating such attributes could potentially enhance the overall description and uniqueness of the respective pattern description.

5.6 Summary & Conclusion

In this Chapter, an image processing pipeline was introduced for generating accurate landmark information from cystoscopic image data. The processed image data resulting from the pipeline shows that the cystoscopic image is transformed into a skeletonized representation and a graph representation, providing a sparser but more distinctive encoding of the reliable structures. Figure 5.33 depicts the various stages involved in this process. A network-based approach was applied for pre-evaluation, identifying image regions suitable for landmark extraction through masking and disregarding distorted, inappropriate, and nuisance regions. The network was trained using a data-driven process based on hand-labeled datasets explicitly designed for the cystoscopic environment, excluding artifacts such as tools, water bubbles, obliterated tissue, and resected and floating tissue.

An unsupervised filter design based on Differences in Gaussian (DoG) filters were employed for vascular segmentation. The filter design, devised explicitly for vascular structures, enables flexible parametrization to adjust segmentation sensitivity

according to the dataset at hand. Through binarization and skeletonization of identified structures, specific landmark registration of complete structural trajectories is facilitated. The landmark information is compressed into a limited set of point representations, with intersection points serving as valuable landmarks due to their consistent tissue correspondence, even under deformation. To stabilize landmark assignment, edge information represents the vascular trajectories between the identified nodes, resulting in a graph representation that encodes adjacency information. This transforms the assignment of individual points into a problem of assigning graphs to one another, facilitating pattern matching. The graph extraction process verifies adjacent structures at the pixel level. However, even minor structural disruptions can affect the segmented intensity and result in non-contiguous structures. To address this issue, a data-driven approach was proposed that compensates for these disruptions, accurately identifying visibly connected structures, even those previously considered interrupted by traditional pixel-based edge identification. The results are contingent on the training data used. The network architecture and extraction procedure offer an efficient solution under varying dimension sizes, transferable and expandable to extract vascular structures from other organ sites given appropriate training data. Combining traditional methods with the graph extraction network is also possible, with the latter providing more robust edge extraction. The proposed graph network structure holds the potential to be expanded to extract additional information, such as latency spaces or reliability values, in the future. To enrich the information content of the graph representation, edge curvatures were extracted in addition to edge position information. The graph representations with attribute descriptions provide a mathematical characterization of the underlying structure, thereby enabling the assignment of patterns that belong to each other.

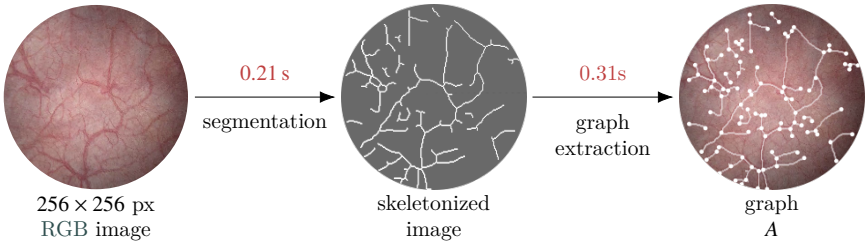


Figure 5.33: Landmark extraction extraction by image segmentation and graph extraction. The intraoperative image is processed into a binary skeletonized representation of the main visible vascular structures. The segmentation enables the extraction of robustly identifiable landmark points at the intersections of vascular structures while preserving the inter-connectivity of the vascular pattern.

6 Graph Matching

The localization and scene reconstruction problem relies on the accurate alignment of corresponding landmark structures. The registration task subsequently follows the preceding graph extraction. The graph extraction task requires the identification of the optimal permutation and selection matrix to match equivalent nodes between similar graphs. The problem can be generally stated as

$$\min_{\Pi} \sum_{i=1}^{n} \boldsymbol{A}_{i,j}[\boldsymbol{A}_i \neq \boldsymbol{A}^*_{\Pi(i)}] \ , \tag{6.1}$$

where Π is the permutation matrix that establishes correspondences between two adjacency matrices, $\boldsymbol{A}$ and $\boldsymbol{A}^*$, which can be considered as noisy versions of each other. The objective of graph matching is to find the permutation matrix Π that minimizes the sum of the mismatched edges between the respective graph representations $\mathcal{G}(\boldsymbol{A})$ and $\mathcal{G}^{\simeq}_{\mathsf{G}}(\boldsymbol{A}^*)$. Therefore, the graph registration process must be robust enough to handle variations in graph representation caused by changes in imaging conditions and deformations.

In this study, a latent node representation is developed to encapsulate all graph information with respect to a specific node, facilitating direct node-to-node comparisons. This representation conveys the connectivity structure of the graph, which is crucial for pattern identification. The underlying concept is that similar graph patterns will exhibit comparable structural contexts.

As demonstrated in [39], an effective method for capturing the connectivity and structural properties of a graph involves mapping the degrees of adjacent nodes into descriptors. These descriptors can then be employed to compare and identify similar patterns within analogous graph representations. In essence, node degrees act as a means to describe and compare the structural surroundings of nodes in a graph, ultimately enabling pattern recognition.

However, this approach can result in significant discrepancies and ambiguities for the descriptor-based matches. For example, a set of matches $\mathbf{m}_{\mathcal{G}\rightarrow\mathcal{G}^*}$ found for graph $\mathcal{G}$ within a target graph $\mathcal{G}^*$ may not necessarily satisfy the reverse $\mathbf{m}_{\mathcal{G}\leftarrow\mathcal{G}^*}$ of the optimal matching for graph $\mathcal{G}^*$ within $\mathcal{G}$. Especially when constructing a global graph model that combines all observed patterns as a navigation map, resolving these ambiguities

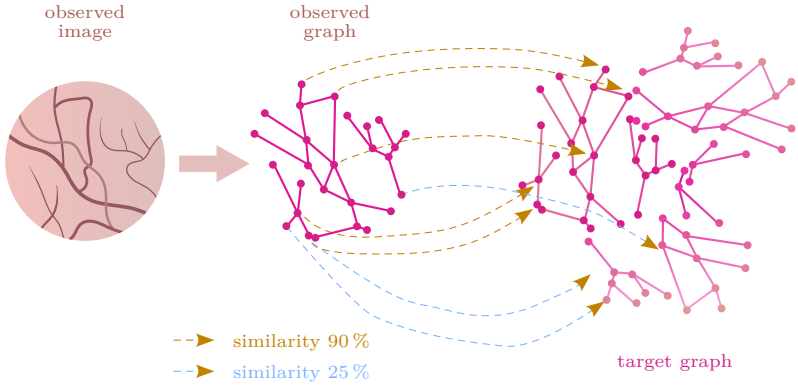


Figure 6.1: A graph matching procedure is applied to the extracted graph from the current intraoperative image by comparing it to subgraphs present in the global graph representation via a similarity measure. The similarity measure compares the adjacencies, along with the attributes of the graph nodes and edges.

and adequately accounting for them is crucial to prevent issues during navigation. The graph matching problem is depicted in Figure 6.1 as it pertains to this work.

The issue of erroneous graph matching is addressed in [104], and specifically, the matching of vascular retina graphs is addressed in [60]. However, the existing literature has not thoroughly addressed the robustness of graph matching under deformation. Furthermore, the methods for retinal mapping presented in [60] are susceptible to failure when Euclidean distances are used as the decision criterion for reliable matches. To address this gap, this work proposes a robust graph-matching approach that accounts for deformation effects in vascular graph registration, which is applicable to both image-to-image observations and image-to-reconstructed 3D surface registration.

This work extends the registration of single image observations from the 2D image plane by merging individual observations into a global representation on a 3D model surface. This approach addresses two challenges: first, the 3D model enables the sequential registration of observed graph information, allowing individual graphs to be merged in sequence and enabling comparison of an observed object to all structures seen up to that point in a single pass. Second, by integrating an embedded geometry reconstruction, it allows for robust graph registration, as deformation effects can be inherently respected. However, it may not always be reliable, particularly in cases of sudden loss of vision, as the imaged surface pattern may show a different surface location when vision is restored.

The following questions arise from the investigation into the combination and matching of graph patterns for navigation and the construction of a global navigation map:

- How can a latent space representation for graph features be designed to produce discriminative and robust node features that are resistant to deformation?
- How can the efficiency of descriptor-based matching, which involves a discrete evaluation of all node combinations between the relevant graphs, be improved regarding real-time performance?
- How can outliers be detected from the set of descriptor-based matches to ensure that the analyzed matches allow for deformation while accounting for pathological conditions?
- How can deformation be addressed such that the descriptor space of the constructed global navigation map can be updated to reflect the observed geometric conditions?
- How can newly observed and unmatched structures be incorporated into a global graph representation, allowing for updates to the navigation map without duplicating the pattern representations?

In this chapter, a three-step procedure is presented to establish a robust and comprehensive graph construction that accommodates all patterns under deformation conditions. The procedure begins with a comparison of structural and feature properties between nodes in different graphs to identify potential node matches, as described in Section 6.1. This results in a set of similarity-based node matches, although some outliers may remain. The second step, presented in Section 6.2, employs a novel outlier removal method based on vascular structures to validate the node matches thoroughly. Finally, Section 6.3 presents a proposed global graph representation consisting of two parts. The first part involves storing the graph on a 3D surface, which allows for restoring the geometry ratios. The second part involves building a verification process that prevents duplication and overloading of similar information during updates of newly observed patterns.

6.1 Descriptor-based Graph Matching

The similarity-based graph matching algorithm relies on descriptor-based similarity comparisons as its cornerstone. The descriptor design serves as a numerical fingerprint that encapsulates the properties of a node and its neighborhood. Based on the similarities of the descriptors, graph matches are determined. The design of the descriptor is presented in Section 6.1.1, and the matching operations are constructed in Section 6.1.2. The matching process is then integrated into a k-dimensional (kd)-search tree, enabling the implementation of an efficient algorithm for performing complex membership searches.

6.1.1 Descriptor Design

The graph properties are encoded in descriptors that facilitate the large-scale matching of pattern information. The structural properties of each node, including adjacency information and attribute properties, are combined into a vector-like descriptor design. These descriptors associate pattern properties with a node's position and enable the matching of corresponding graph nodes by evaluating their similarity. Therefore, the challenge is to design the descriptors so that the structural and attribute properties are distinguishable and resilient to deformation. To make the matching invariant to the initial position, the descriptors must be designed to be spatially invariant. Furthermore, the descriptors must be designed to be minimally impacted by potential disruptions in the graph extraction, ensuring that the information required for correct assignment is preserved even in the presence of missing or additional graph information in the current observation.

6.1.1.1 Embedding of Structural Interconnectivity

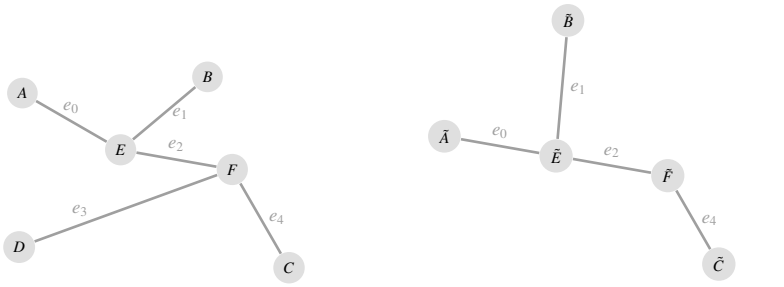
A structural descriptor set $\boldsymbol{d}_{\mathrm{str},u}$ is designed to encode the structural interconnectivity within a graph $\mathcal{G}$ with respect to the reference node u. As an auxiliary descriptor representation, the vector $\tilde{\boldsymbol{d}}^{k}_{\mathrm{str},u}$ is defined to enumerate the number of nodes with degrees equal to or greater than a certain threshold, in accordance with their order within the neighborhood of k steps from the reference node u. To ensure comparability and generalizability, the structural descriptor $\tilde{\boldsymbol{d}}^{k}_{\mathrm{str},u}$ is represented as a fixed-dimensional vector. However, to capture all available structural information beyond the fixed vector dimension, structures with degrees equal to or greater than the maximum degree in the graph, denoted by $n_{\mathrm{str,max}}$, are included in the last entry in $\tilde{\boldsymbol{d}}^{k}_{\mathrm{str},u}$. The embedding process of the graph's structural information is illustrated in Figure 6.2. Furthermore, to make the descriptor more robust and less sensitive to changes in the structural graph topology, element-wise logarithmic scaling and weighting are proposed in (6.2). The descriptor degree and adjacency information is computed by

$$\boldsymbol{d}_{\mathrm{str},u} = \sum_{k=0}^{k_{\max}} \delta_{\mathrm{str}}^{k-1} \ln(\tilde{\boldsymbol{d}}^{k}_{\mathrm{str},u}), \tag{6.2}$$

where δ_{str} assigns less weight to more distant structures.

6.1.1.2 Embedding of Spatial Information

In addition to capturing structural interconnectivity, further structural information is obtained from the node positions themselves. Although node positions are not invariant to rotation and translation, Euclidean dependencies can be incorporated by including edge lengths and angles between edges into the descriptor design, which



$$\left.\begin{aligned}\tilde{\boldsymbol{d}}^1_{\mathrm{str},A} &= [0,0,1,0]\\ \tilde{\boldsymbol{d}}^2_{\mathrm{str},A} &= [1,0,1,0]\end{aligned}\right\} \quad \boldsymbol{d}_{\mathrm{str},A} = [0.5,0,1.5,0]$$

$$\tilde{\boldsymbol{d}}^3_{\mathrm{str},A} = [2,0,0,0]$$

$$\left.\begin{aligned}\tilde{\boldsymbol{d}}^1_{\mathrm{str},F} &= [2,0,1,0]\\ \tilde{\boldsymbol{d}}^2_{\mathrm{str},F} &= [2,0,0,0]\end{aligned}\right\} \quad \boldsymbol{d}_{\mathrm{str},F} = [3,0,1,0]$$

(a) Structural graph embedding of $\mathcal{G}_A$.

$$\left.\begin{aligned}\tilde{\boldsymbol{d}}^1_{\mathrm{str},\tilde{A}} &= [0,0,1,0]\\ \tilde{\boldsymbol{d}}^2_{\mathrm{str},\tilde{A}} &= [1,0,1,0]\end{aligned}\right\} \quad \boldsymbol{d}_{\mathrm{str},\tilde{A}} = [0.5,0,1.5,0]$$

$$\tilde{\boldsymbol{d}}^3_{\mathrm{str},\tilde{A}} = [1,0,0,0]$$

$$\left.\begin{aligned}\tilde{\boldsymbol{d}}^1_{\mathrm{str},\tilde{F}} &= [1,0,1,0]\\ \tilde{\boldsymbol{d}}^2_{\mathrm{str},\tilde{F}} &= [2,0,0,0]\end{aligned}\right\} \quad \boldsymbol{d}_{\mathrm{str},\tilde{F}} = [2,0,1,0]$$

(b) Structural graph embedding of $\mathcal{G}_B$.

Figure 6.2: Comparison of the structural graph embeddings of two similar graphs with distinct structures reveals the impact of structural modifications in the descriptor representation. The embedding vector is calculated based on (6.2), excluding the ln operation to simplify the interpretation of the example.

are independent of any coordinate specification. Therefore, the corresponding edge length for any edge $e_{i,j}$ is determined by

$$\boldsymbol{l}_{i,j} = \|n_i - n_j\| \,, \tag{6.3}$$

where n_i and n_j specify the positions of the start and end nodes of the edge $e_{i,j}$, respectively. In addition, the inclusion angles at node n_i between adjacent edges are determined by

$$\boldsymbol{\alpha}_i = \begin{cases} \left\{ \measuredangle\left(e_{i,j}, e_{i,j+1}\right) \right\} & \forall\, j \in \mathsf{adj}(i) \geq 1 \\ 360^\circ & \text{otherwise} \end{cases} , \tag{6.4}$$

where $e_{i,j\in\mathsf{adj}(i)}$ indicates all edges starting at node n_i and connecting to the adjacent nodes $\mathsf{adj}\,(i)$.

The edge attributes $\boldsymbol{c}_{i,\star}$ along the edge lengths $\boldsymbol{l}_{i,\star}$ pertain to a particular node n_i. The set of intersection angles at node n_i is denoted by (6.4). They vary in dimension depending on the node degree $\mathsf{deg}\,(n_i)$ of the reference node n_i. In order to ensure comparability, it is crucial that the descriptors have the same structure and size

across the entire set. To achieve this, the statistical embedding function

$$\boldsymbol{d}_{\mathcal{X}} = f_{\text{embd}}(\mathcal{X}) = \begin{pmatrix} \sum_j \mathcal{X}_{i,j} \\ \max_j \mathcal{X}_{i,j} \\ \min_j \mathcal{X}_{i,j} \end{pmatrix} \quad \forall i \tag{6.5}$$

is proposed, where $\mathcal{X}$ is the respective embedding vector with an arbitrary dimension. Thus, any general attribute data $\mathcal{X}$ is transferred into a fixed-dimensional vector representation $\boldsymbol{d}_{\mathcal{X}}$, irrespective of the dimensions of $\mathcal{X}$. Applying (6.5) to all nodes, a comprehensive descriptor set is created, comprising the following edge and node attributes in a specified order

- $\boldsymbol{d}_{\boldsymbol{l}} = f_{\text{embd}}(\boldsymbol{l})$ for edge lengths,
- $\boldsymbol{d}_{\boldsymbol{\alpha}} = f_{\text{embd}}(\boldsymbol{\alpha})$ for inclusion angle sets,
- $\boldsymbol{d}_{\boldsymbol{c}} = f_{\text{embd}}\left(\overline{\overline{\boldsymbol{A}}}_{ij}\right)$ for edge curve attributes as defined in (5.12).

The resulting descriptor identities can be combined into a complete descriptor representation $\boldsymbol{d} = [\boldsymbol{d}_{\text{str}}, \boldsymbol{d}_{\boldsymbol{l}}, \boldsymbol{d}_{\boldsymbol{\alpha}}, \boldsymbol{d}_{\boldsymbol{c}}]^{\text{T}}$, which represents the numerical fingerprint of the corresponding structure and attribute information, enabling node-based similarity checks.

6.1.2 Similarity Definition

Once nodes are encoded by their descriptors, their similarities can be compared to identify corresponding node matches between an observed graph $\mathcal{G}$ and a given target graph $\mathcal{G}^*$. The reliability of a match is assumed to be proportional to the similarity of its descriptor. Thus, the match between a node $u \in \mathcal{G}$ and a node $v \in \mathcal{G}^*$ is determined by minimizing their differences in similarity, as stated by

$$\mathbf{m}_{u \leftrightarrow v} = \{u, v\} = \min_{\tilde{v} \in \mathcal{G}^*} \ \mathtt{sim}(\boldsymbol{d}_u, \boldsymbol{d}_{\tilde{v}}^*) \ . \tag{6.6}$$

The similarity between two descriptors is measured by their difference

$$\mathtt{sim}_{\text{euc}}(\boldsymbol{d}_a, \boldsymbol{d}_b) = \|\boldsymbol{d}_a - \boldsymbol{d}_b\| \ . \tag{6.7}$$

Alternatively, the cosine similarity, analogously to (6.8), provides a scaling-invariant similarity measure

$$\mathtt{sim}_{\cos}(\boldsymbol{d}_a, \boldsymbol{d}_b) = 1 - \left(\frac{\boldsymbol{d}_a^{\text{T}} \boldsymbol{d}_b}{\|\boldsymbol{d}_a\| \|\boldsymbol{d}_b\|} \right). \tag{6.8}$$

The cosine similarity is an appropriate choice for comparing spatial attribute information, such as edge lengths, as it is invariant to scaling. This means that identical patterns will result in the same similarity value, while non-uniform deformations of patterns lead to a deviation in the calculated similarity.

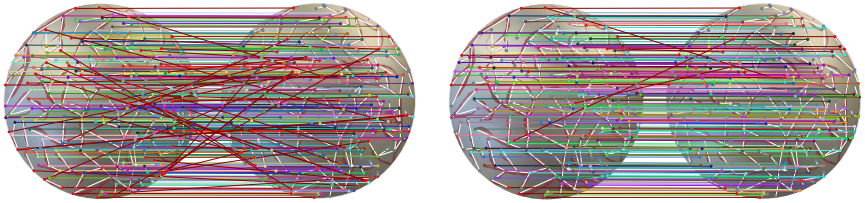
6.1.3 Similarity-based Descriptor Comparison Exploiting a *KD*-Tree Evaluation

If the similarity $\mathtt{sim}(\boldsymbol{d}, \boldsymbol{d}^*)$ between all node combinations in the observed graph $\mathcal{G}$ and target graphs $\mathcal{G}^*$ is calculated individually for all possible combinations $|\mathcal{G}| \times |\mathcal{G}^*|$, the process can become computationally intensive. As the number of nodes in the two graphs increases, the number of comparisons that need to be processed increases exponentially. To minimize the computational burden and improve the efficiency of the matching process, it is beneficial to limit the evaluation to only those combinations that satisfy certain initial similarity checks.

By organizing descriptors in a kd tree data structure, respective descriptors can be pre-sorted based on their value ranges. The kd tree serves as an effective tool for storing and accessing multidimensional data, arranging the data in a tree-like format according to a predefined sorting rule, which facilitates the search and retrieval of desired data points [98]. Employing this approach, a single graph image $\mathcal{G}$ can be compared to a collection of known patterns in $\mathcal{G}^*$ by identifying the most similar nodes in $\mathcal{G}^*$. The kd-tree query aids in an early exclusion of potential matches that do not meet the required minimum similarity threshold. This strategy enhances the matching process's efficiency by reducing the number of explicit similarity evaluations to a preselected set of potential data, resulting in improved overall speed and accuracy.

6.1.4 Cross Check Condition

Figure 6.3a presents a descriptor-based matching process between two successive graph observations. It is evident that a significant number of matches are found. Nonetheless, it is apparent that several outliers are also present, which are highlighted in red. It is important to note that the selected node $u \in \mathcal{G}$ may not necessarily be the optimal choice for the selected node $v \in \mathcal{G}^*$ when searching in the reverse



(a) Identified matches between the most similar descriptors in the left and right graphs.

(b) Descriptor-based matches that meet the cross-check criteria. Remaining outliers are highlighted.

Figure 6.3: Descriptor-based matches found by identifying the most similar descriptors between the left and right graphs. Outliers are highlighted in ● red.

direction. The initial comparison between the two graphs results in several potential matches, but not all of these may be accurate. Some matches may be coincidental rather than reflect similarities between nodes in their respective neighborhoods. To validate the matches between $u \in \mathcal{G}$ and $v \in \mathcal{G}^*$, a cross-check is performed by verifying that each match $\mathbf{m}_{u \leftrightarrow v}$ satisfies the following conditions: (i) for the given node $u \in \mathcal{G}$, v is the most similar node among all nodes in $\mathcal{G}^*$, and (ii) for the given node $v \in \mathcal{G}^*$, u is the optimal node among all nodes in $\mathcal{G}$. This confirms that the match is mutually optimal, as the descriptor matches are the most similar between the two graphs. Verifying the compatibility of matches and avoiding the simultaneous matching of a single node to multiple nodes helps to eliminate false matches. Figure 6.3b displays the matches that satisfy the cross-check criteria. Despite passing the cross-check test, a considerable number of outliers remain present.

These outliers can be detected by contrasting the individual matches against the consensus, which is established by the majority of matches running from the top left to bottom right. By classifying them as deviating from the consensus, these outliers can be removed from the set $\tilde{\mathbf{m}}_{\mathcal{G} \to \mathcal{G}^*}$ of potential matches, where the set $\tilde{\mathbf{m}}_{\mathcal{G} \to \mathcal{G}^*}$ represents all cross-checked matches between the respective two graphs $\mathcal{G}$ and $\mathcal{G}^*$.

6.2 Outlier Removal

Although cross-validation techniques are employed to establish graph correspondences, the descriptor-based similarity comparison is vulnerable to outliers. To address this issue, it is essential to identify and remove outliers from the set of matches based on their similarity. While the random sampling consensus (RANSAC) algorithm is widely regarded as the gold standard for outlier classification in image processing, it may not be suitable for deformable scenes since it relies on a consensus of matches that aligns with the camera model and a rigid scene.

To overcome this limitation, a structure-based outlier removal (SbOR) concept is proposed, which is tailored to vascular structures. This method identifies outliers in a given set of matches based on pathologically feasible registrations. The RANSAC method is presented first in the following section, as it can reliably identify outliers under rigid scene conditions. Next, in Section 6.2.2, the SbOR concept is introduced as a solution to address the issues caused by deforming environments during intraoperative situations. Finally, in Section 6.2.3, the two methods are compared, highlighting their advantages and corresponding inaccuracies. This comparison will help to choose the suitable outlier classification for the current conditions.

6.2.1 RANSAC Outlier Removal

The RANSAC algorithm selects a subset of data points, fits a model to this subset, and assesses the discrepancy between the model fit and the remaining data points.

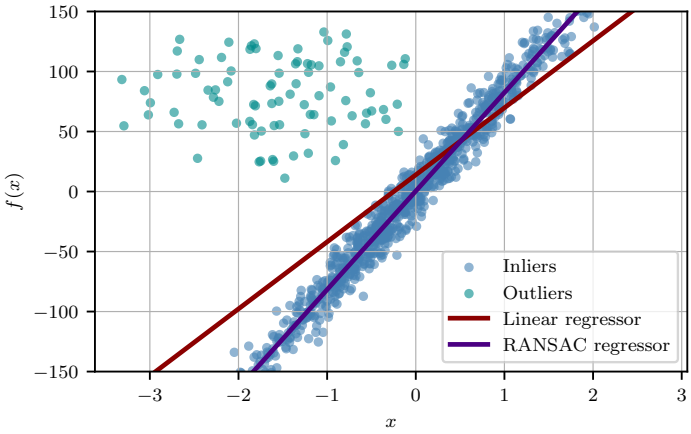


Figure 6.4: RANSAC classification separates data into inliers and outliers by fitting a model to the data and identifying the consensus set. This method is more robust to outliers than the least-squares regression, which does not respect the inlier set and leads to a poor model fit.

The algorithm is presented in Algorithm 1 according to [29]. Ideally, the model fit that minimally deviates from the entire data set of given matches $\mathbf{m}$ contains no unchecked outliers. Based on the model fit $\texttt{Model}^\star$ that achieves the highest consensus, data points are classified as inliers or outliers with respect to the given error tolerance δ_{RANSAC}.

An example of outlier detection based on a linear model is shown in Figure 6.4. In this case, the RANSAC algorithm identifies the outlier data, while the respective model fit —according to the RANSAC regression— follows the consensus of the inlier data points. In contrast, the least squares model fit, which includes all data points, shows significant deviation from the inlier data set and is heavily influenced by the outlier data set, as depicted in Figure 6.4. This example illustrates the effectiveness of the RANSAC algorithm in identifying and removing outliers from a data set.

The triangulation model (2.7) imposes constraints on point matches that are required to align on the image plane for an image-to-image data comparison. A freely moving camera model has six unknown degrees of freedom for rotation and translation. To determine the corresponding parameter space for the model fit $\texttt{Model}^\star$, at least five point matches are required. During each iteration of the RANSAC process, a minimal randomly selected subset $\mathbf{m}_{\text{RANSAC}}$ of five matches ($\|\mathbf{m}_{\text{RANSAC}}\| = 5$) is selected from the original set of matches $\mathbf{m}$ to calculate a model fit $\texttt{Model}^\star$. The resulting model fit of each iteration is compared to the entire set of matches $\mathbf{m}$. Thus, the consensus is established by iteratively checking the current model fit against all

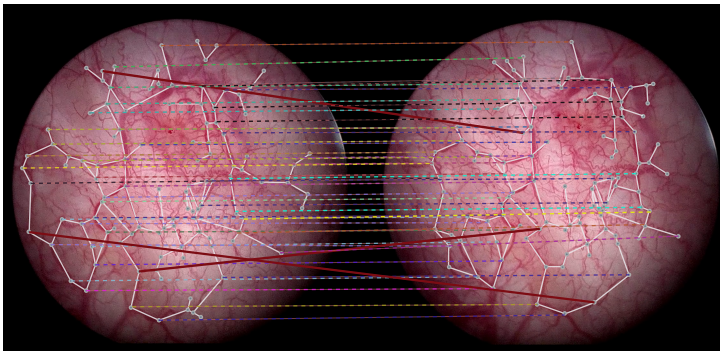
Algorithm 1: RANdom SAmple Consensus (RANSAC) Algorithm

```
Data: m ;                                          /* Descriptor-based matches */
Result: m_RANSAC ;    /* Verified matches that are consistent with the
       model */
nBestInliers ← 0 ;                          /* number of identified inliers */
nIterations ← N_RANSAC ;                          /* number of iterations */
Model ← (2.5) ;                  /* define underlying consensus model */
ErrorModel ← δ_RANSAC ;                              /* error tolerance  */
m̃_RANSAC ← ∅ ;                              /* temporary inlier matches */
for i ≤ nIterations do
    i + + ;
    m^[5] = SelectSubSet(m) ;    /* select a subset m^[5] ⊂ m given by five
     random samples in m */
    Model_i(R, T) = ModelFit(m^[5]) ;  /* exploit m^[5] to identify free model
     paramters R and T for (2.7)  */
    for match m in m do
        if distance(Model_i ↔ m) ≤ δ_RANSAC then
            m̃_RANSAC+ = m;  /* determine set of all points that respect a
             predefined error tolerance  */
        end
    end
    if NrIn(m̃_RANSAC) ≤ nBestInliers then
        nBestInliers = NrIn(m̃_RANSAC);      /* size of best inlier set up to
         this point  */
        Model* = Model(R, T) ;       /* best model fit up to this point */
    end
end
m_RANSAC ← distance(Model* ↔ m) ≤ δ_RANSAC ;          /* determine set of
 inliers according to best model fit seen  */
```

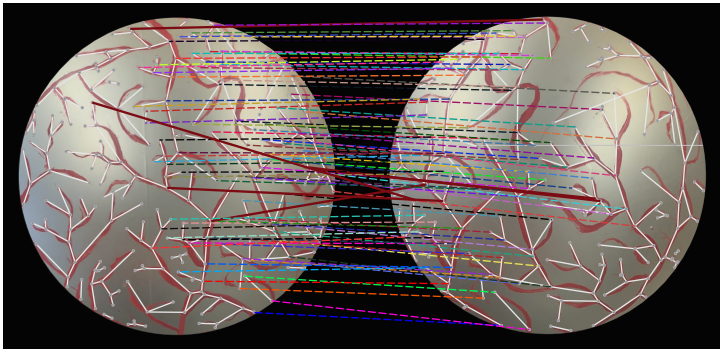
descriptor-based matches $\mathbf{m}$.

Figure 6.5 illustrates an example where the RANSAC algorithm with 40 iterations is applied to detect outlier matches, which are highlighted in red. To establish a consensus, all matches are checked against the camera model. Figure 6.5a shows reliable matches established for two consecutive observations of an intraoperative scene. To further assess the reliability of the RANSAC algorithm, the synthetic bladder model is used to test a more challenging matching problem, where only a limited part of the visible area has a corresponding coincident pattern, as depicted in Figure 6.3b. Despite the increased complexity, the descriptor-based graph matching, along with the RANSAC outlier classification, is able to establish robust and reliable

matches with no visibly undetected outliers. These results prove the effectiveness of the descriptor-based graph matching technique and the RANSAC algorithm in identifying and eliminating outlier matches for challenging matching problems.



(a) Descriptor-based matches with detected outliers for intraoperative data.



(b) Descriptor-based matches with detected outliers for synthetic bladder model.

Figure 6.5: Descriptor-based similarity matches between node pairs. Corresponding pairs are indicated with a dashed line in distinct colors. Detected outliers, identified by the RANSAC outlier check, are highlighted in ● red. Manual examination confirms that the remaining matches $\mathbf{m}_{\text{RANSAC}}$ are deemed correct.

Significant deformation between individual observations can cause the corresponding camera model to become invalid, which in turn leads to failure of the RANSAC procedure. As a result, all identified matches may be classified as outliers since no consensus in agreement with the camera model can be established. To address this issue, the error threshold δ_{RANSAC} for matching onto a rigid camera model (2.7) could be increased. Nevertheless, increasing the error threshold δ_{RANSAC} is insufficient for accurately re-

lating deformed graph patterns to a rigid projection model. This limitation impedes the use of the RANSAC method in addressing the deformation-invariant reconstruction pipeline and achieving the specific goal of a deformation-invariant outlier detection procedure. Moreover, the entire descriptor-based matching approach becomes obsolete without outlier elimination, as unidentified outliers can seriously harm and undermine the entire image-based scene reconstruction.

6.2.2 SbOR Algorithm

To meet the needs of intraoperative applications, a deformation-invariant and structure-preserving outlier removal method is proposed. This method respects pathological conditions and overcomes intraoperative challenges by exploiting the connections between structures, rather than relying on a consensus for a rigid surface description. The method is tailored for vascular structures, which may undergo changes in length ratios and orientation due to applied forces and resulting deformation. However, from a pathological perspective, it can be excluded that blood vessels change their vascular interconnections due to deformation.

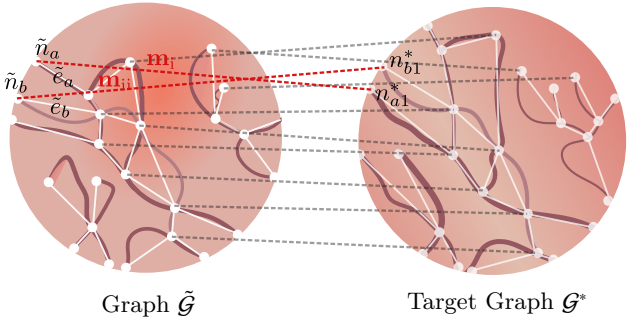
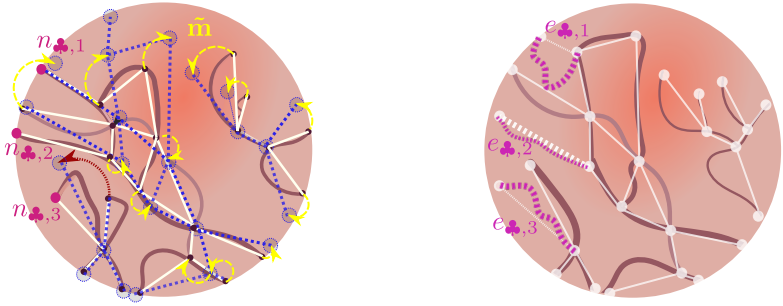


Figure 6.6: Descriptor-based matches between two images, where deformation occurs. For example, two problematic outlier matches m_{i} and m_{ii} are highlighted that must be identified.

The proposed deformation-invariant outlier classification method aims to accurately identify feasible matches under deformation while detecting matches that require self-intersections, which are pathologically infeasible. Figure 6.6 illustrates a destructive match that is not permitted due to the pathological nature of the tissue. Matches associated with edges $\tilde{e}_a$ and $\tilde{e}_b$ of the current graph observation $\tilde{\mathcal{G}}$, which intersect to align with the corresponding target nodes n^*_{a1} and n^*_{b1} of the undeformed target graph $\mathcal{G}^*$, may need to be classified as outliers. Therefore, matches associated with $\tilde{e}_a$, $\tilde{e}_b$, or even all matches associated with the nodes n^*_{a1} and n^*_{b1}, may need to be classified

as outliers. However, graph edges whose correspondence does not constrain the initial and final nodes are unproblematic, as they can move according to the forced deformation in the deformed plane.



(a) The analyzed graph $\tilde{\mathcal{G}}$ in ○ white overlaid with the target graph structure $\mathcal{G}^*$ in ● blue. Descriptor-based node matches between the given graphs are highlighted in ● yellow.

(b) Deformed target graph $\mathcal{G}^*$ with pulled graph edges $e_{\clubsuit}$ that do not have any consecutive match.

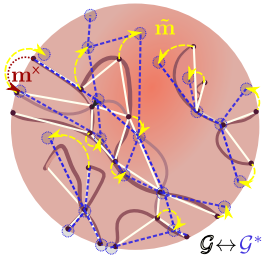
Figure 6.7: Two graphs are shown: a reference graph $\tilde{\mathcal{G}}$ and a deformed target graph pattern $\mathcal{G}^*$. Unmatched nodes, denoted as $n_{\clubsuit}$, are not problematic as they do not cause any self-intersecting patterns and are carried along the deformation process, as illustrated in (b).

Figure 6.7 illustrates an exemplary phenomenon of a tuple of corresponding graph patterns under deformation. Specifically, nodes identified as $n_{\clubsuit}$, which do not have corresponding matches, are impacted by their neighboring nodes that do have corresponding node matches. As a result, relatively displaced edges $e_{\clubsuit}$ are visible in the deformed plane. This displacement is characteristic of the tissue pathology, where the tissue layer between blood vessels remains unchanged while the surrounding tissue layers undergo deformation due to redistribution induced by the matched node points to align the respective graph patterns before and after deformation.

Based on these two scenarios, it is recognized that matches, which lead to self-intersecting edges when matching the respective graph geometry of related nodes in a target graph, are problematic. Consequently, matches that lead to pathological, infeasible geometry adjustments must be classified as outliers. In contrast, edges that do not exhibit any matches are generally unproblematic, as they are not bound by one-to-one correspondences and consequently shift relatively along with the deformed surface geometry. Based on this principle, a deformation-invariant outlier removal procedure is designed to remove pathologically invalid matches, as outlined in Algorithm 2. In the proposed outlier removal approach, the examined graph $\tilde{\mathcal{G}}$ is first pruned to all unproblematic structures by removing all nodes with adjacent edges, where no match is found. Subsequently, the resulting pruned graph $\tilde{\mathcal{G}}^{\ominus}$ is

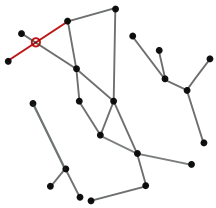
aligned according to the established matches to coincide with their respective target nodes in $\mathcal{G}^*$. Ultimately, the structures aligned to the observed deformation allow for the identification of ambiguous matches $\mathbf{m}^{\mathsf{x}}$ that require self-intersecting edges to follow the deformed target structure $\mathcal{G}^*$.

For a given pair of consecutive graph observations and given matches, Figure 6.8 shows the corresponding pruned graph $\tilde{\mathcal{G}}^{\ominus}$ and the graph geometry adapted such that matching nodes of $\tilde{\mathcal{G}}^{\ominus}$ and $\mathcal{G}^*$ coincide, which thereby reveal all self-intersecting edges.



(a) Current graph extraction $\mathcal{G}$ in ○ white where the target graph $\mathcal{G}^*$ is overlaid in ● blue. The corresponding node matches are specified in ● yellow.

(b) Pruned target graph $\mathcal{G}^{*\ominus}$. All nodes and adjacent edges of $\mathcal{G}^*$ that do not have any corresponding match with $\mathcal{G}$ are deleted.



(c) The pruned graph $\mathcal{G}^{*\ominus}$ is adjusted to align with the current graph extraction $\mathcal{G}$, based on the given descriptor matches $\tilde{\mathbf{m}}$.

Figure 6.8: Outlier matches are detected by observing the self-intersection of the vascular structures in the pruned graph, which resulted from adjusting the given descriptor matches $\tilde{\mathbf{m}}$.

It must be noted that the actual location of the intersection point itself is insignificant since the reproduced deformation is a three-dimensional surface problem. However, the simplified two-dimensional test provides sufficient qualitative information about whether or not the edges would be forced to intersect by any particular match. Thus, the procedure reveals the three-dimensional self-intersection problem qualitatively at limited complexity. The identified intersections are associated with contradictory matches that would cause dubious structural changes. In tracing the intersecting edges, $\tilde{e}_a = \{n_{a1}, n_{a2}\}$ and $\tilde{e}_b = \{n_{b1}, n_{b2}\}$, to all involved nodes n_{a1}, n_{a2}, n_{b1}, and n_{b2}, it may be necessary to remove all associated matches, which can include up to four individual node matches. However, to avoid simply discarding all involved matches, the similarities of the respective descriptors identified for matching is compared to obtain more information about the reliability of the individual matches.

The reliability of a match is assessed by determining the ratio of descriptor similarities $\mathbf{m}_{a1,b*} \leftrightarrow \mathtt{sim}(\boldsymbol{d}_{a1}, \boldsymbol{d}^*_{b*})$ and $\mathbf{m}_{a2,b**} \leftrightarrow \mathtt{sim}(\boldsymbol{d}_{a2}, \boldsymbol{d}^*_{b**})$ for matches to arbitrary nodes n_{b*} and n_{b**} in the target graph. A match is considered an outlier if its descriptor

similarity is significantly lower than the residual similarities related to the intersecting edges. When this occurs, the intersection process is repeated for all matches related to the nodes n_{a1}, n_{a2}, n_{b1}, and n_{b2} to determine whether the self-intersection issue is resolved iteratively. If the retesting resolves the self-intersection point, the remaining matches are considered legitimate and assigned to the set $\mathbf{m}_{\text{SbOR}} \subseteq \mathbf{m}$ that adheres to the pathological constraints under deformation.

For instance, if the intersection between the two edges $\tilde{e}_a$ and $\tilde{e}_b$ persists, the match of the second largest descriptor similarity discrepancy is discarded next, and the procedure is repeated until all involved matches are identified as outliers or the self-intersection point is resolved, as outlined in Algorithm 2. This approach enables the identification and verification of all acceptable matches $\mathbf{m}_{\text{SbOR}} \subseteq \mathbf{m}$ that comply with the pathological constraints under deformation.

Algorithm 2: Graph-based Outlier Removal

Data: $\mathbf{m}$, $\mathcal{G}$, $\mathcal{G}^*$; /* Descriptor-based matches, input graph and the target graph */
Result: $\mathbf{m}^\star$, $\mathbf{m}^{\leftrightarrow}$; /* verfied matches and outlier matches */
$\hat{\mathcal{G}} \leftarrow$ empty;
$\mathbf{m}^\star \leftarrow \mathbf{m}$;
$\mathbf{m}^{\leftrightarrow} \leftarrow$ empty;
$\tilde{\mathcal{G}}^{\ominus} = \text{PrunedGraph}\left(\tilde{\mathcal{G}}^{\ominus}, \mathbf{m}\right)$; /* Prune current graph observation $\tilde{\mathcal{G}}$ to graph with purely critical edges -> $\tilde{\mathcal{G}}^{\ominus}$ */
for *match* m *in* $\mathbf{m}$ **do**
  $e(\{n_1, n_2\}) = \text{getAssociatedEdge}(\mathcal{G}, m)$;
  $\tilde{e} = \{\tilde{n}_1, \tilde{n}_2\} = \text{updateNodePostionsAccordingToTarget}(e = \{n_1, n_2\}, \mathcal{G}^*)$;
  $\hat{\mathcal{G}} += \tilde{e} = \{\tilde{n}_1, \tilde{n}_2\}$; /* add matched edges with updated positions to pruned graph */
end
for *possible edge combinations* e_1, e_2 *in* $\hat{\mathcal{G}}$ **do**
  while hasIntersection(e_1, e_2) **do**
    $n_{e_1,1}, n_{e_1,2}, n_{e_2,1}, n_{e_2,2} = \text{adjacentNodes}(e_1, e_2)$;
    $\tilde{n}_{e_1,1}, \tilde{n}_{e_1,2}, \tilde{n}_{e_2,1}, \tilde{n}_{e_2,2} = \text{adjacentNodes}(\tilde{e}_1, \tilde{e}_2)$;
    $\tilde{n}_{e_\star} = \max_{e_1, e_2} \text{sim}(\boldsymbol{d}_{n\star}, \boldsymbol{d}^*_{n\star})$;
    $m = \text{getCorrespondingMatch}(\tilde{n}_{e_\star})$;
    $\mathbf{m}^\star$ −= m; /* remove outlier match from feasible match list */
    $\mathbf{m}^{\leftrightarrow}$ += m; /* add match to outlier list */
    RemoveOutlierEdgeFromGraph$\left(\hat{\mathcal{G}}, m\right)$
  end
end

6.2.3 Verification of Outlier Classification Concepts based on a Synthetically Generated Data Set

To evaluate the effectiveness and stability of the proposed SbOR detection concept, a synthetic test dataset is created by distorting and randomly deleting graph information. Figure 6.9 displays a representative sample of the generated dataset. The distortion of node points is simulated using the state-of-the-art distortion model (2.3), with carefully selected distortion parameters. In addition, noise is introduced by deleting random nodes and all adjacent edges. Since the modifications are artificially induced, the ground truth node correspondences between the original and distorted graphs are readily available for comparison.

(a) Initial graph extraction $\mathcal{G}$.

(b) Distorted Graph Geometry $\tilde{\mathcal{G}}$ with node and edge deletion in red.

(c) Resulting modified distorted target Graph $\tilde{\mathcal{G}}$.

Figure 6.9: A synthetically generated dataset using image distortion and edge deletion (where edges in red are deleted) is used to simulate deformation and uncertainty in graph extraction. The availability of ground truth data allows for an accurate comparison as the distorted target graph $\tilde{\mathcal{G}}$ is a modified version of the given arbitrary input graph $\mathcal{G}$.

To ensure comparability, the distortion measure

$$\delta_{\text{dist}} = \frac{\|\mathbf{V} - \tilde{\mathbf{V}}\|^2}{\mathbf{V}} \tag{6.9}$$

is defined to quantify the changes in node positions resulting from the application of the distortion models (2.3a) and (2.3b). The distortion parameter δ_{dist} provides a metric for analyzing the impact of spatial changes on outlier detection in the graph. Additionally, a deletion rate δ_{delet} is introduced, where a proportion of nodes in $\mathcal{G}$ and their corresponding edges are randomly selected and removed to produce the modified graph $\tilde{\mathcal{G}}$. To avoid potential bias in the results, the reverse operation of adding arbitrary nodes and edges is omitted for simplicity. Both larger and smaller graphs are considered alternatively for matching to ensure a balanced analysis between larger target and smaller base graph matching. Figure 6.9 depicts a test graph pair, where the modified version is generated by applying distortion and randomly

deleting edges ($\delta_{\text{delet}} = 6\%$). These artificial graph alterations present a degree of complexity that is comparable to actual data exposed to deformation, rendering them appropriate for a realistic assessment of the graph-matching procedure.

The effect of graph modification on the accuracy of the matching process is investigated for the descriptor-based matching procedure with outlier elimination. The test set includes 200 initial graphs that are similar in complexity to the graph displayed in Figure 6.9. The entire test set and corresponding modifications are analyzed for each tested parameter. During each parameter evaluation, the original and modified graphs serve as the output and target graphs alternatively, resulting in a total of 400 graphs analyzed for each parameter set. The descriptor-based matching method is utilized for matching; based on that, the RANSAC and SbOR concepts are applied for outlier elimination. The empirical results are summarized in Table 6.1, demonstrating the impact of modification parameters δ_{dist} and δ_{delet} on the inlier and outlier matches relative to the ground truth. The outlier detection rate (ODR) denotes the proportion of accurately detected outliers in $\tilde{\mathbf{m}}$ compared to the ground truth, while the false none outlier detection rate (FNODR) represents all matches that are incorrectly classified as outliers matches.

The evaluation reveals that the descriptor method generates a dense distribution of potential matches for unmodified data. However, this density decreases as the degree of modifications (measured by δ_{dist} and δ_{delet}) increases. The proportion of outliers in descriptor-based matches also increases with the extent of modification, as confirmed by available ground truth information. Regarding outlier detection, the RANSAC approach demonstrates exceptional precision with minimal bias and can accurately identify outliers while maintaining a consistently low proportion of δ_{dist}. However, as the degree of deformation increases, the RANSAC method progressively misclassifies more matches as false negatives, represented by the FNODR, resulting in the elimination of all matches beyond a certain threshold. Consequently, all available matches are incorrectly classified as outliers. Although it is possible to adjust the error tolerance for the RANSAC method in response to more significant deformation, doing so may result in the exclusion of more outliers. There is a trade-off in accurately detecting outliers, where either misclassifying non-outliers is accepted or increasing the error threshold to maintain inliers but risking unreliable outlier detection. In contrast, the structure-based outlier method is not significantly impacted by deformation. As a result, the ODR and the rate of FNODR remain relatively stable.

After evaluating the data, it is concluded that the RANSAC method is more accurate than the SbOR concept for non-deformed environments. However, the RANSAC method is ineffective at handling deformations, whereas the structure-based method performs reliably in such cases. Furthermore, a closer examination of the outlier matches that were not detected by the SbOR procedure reveals the presence of non-critical, self-intersecting edges. These cases can only occur when references to nodes in the environment do not require intersections with other edges and do not differ signif-

Table 6.1: Outlier classification for a given set of descriptor- based matches $\tilde{\mathbf{m}}$.

	δ_{dist}	δ_{delete}	Outlier Ratio	ODR [a]	FNODR[a]
RANSAC	0.04	0.05	0.04	0.99	0.06
	0.12	0.05	0.06	0.95	0.23
	0.24	0.05	0.28	0.98	**1.0**
	0.04	0.25	0.04	0.99	0.06
	0.12	0.25	0.06	0.95	0.23
	0.24	0.25	0.28	0.98	1.0
SbOR	0.04	0.05	0.04	0.967	0.0772
	0.16	0.05	0.06	0.923	0.0743
	0.32	0.05	0.28	0.956	0.0838
	0.04	0.25	0.04	0.918	0.0623
	0.16	0.25	0.06	0.957	0.052
	0.24	0.25	0.28	0.924	0.083

[a] Calculated as a weighted average over the images in the holdout test set, with the number of nodes per image as weights.

icantly from actual matches for arbitrary degrees of deformation. Therefore, it is necessary to accept some level of uncertainty in graph matching. Furthermore, the SbOR concept is effective at detecting outliers in the presence of deformation. However, its performance is limited by the descriptor-based matching method's ability to handle deformations. As the level of deformation becomes more pronounced, the accuracy of matches obtained through descriptor-based matching decreases significantly, which means that only a few reliable matches can be obtained in cases of extreme deformation. Consequently, the number of usable matches remains insufficient to even apply any outlier classification to the given descriptor-based matches.

6.3 Global Graph Model

The proposed method of descriptor-based matching and outlier removal enables an efficient and reliable identification of node correspondences between two noisy graphs. To compare graphs across multiple image observations, the current graph observation $\mathcal{G}_i = \mathcal{F}_{\text{Graph}}(\boldsymbol{\mathcal{I}}_i)$ is integrated into a global graph representation $\mathcal{G}_{\mathsf{G}}$. This representation includes all previously acquired observations in one domain, allowing for a new observation to be compared with the entire set of previously observed patterns in a single matching process, as depicted in Figure 6.10. A global pattern representa-

tion has significant potential, especially in the context of intraoperative navigation. When there are disturbances in the field of view or significant changes in camera pose $\boldsymbol{\phi}_{\text{cam}}$, the precision of pattern observations may be limited. In such cases, a comparison with only the immediately preceding pattern observations may quickly become inadequate, resulting in the loss of orientation-critical landmark information.

The objective of this process is to establish a reliable mapping of partially matched patterns $\mathcal{G}_i$ to the global graph representation $\mathcal{G}_\mathsf{G}$. This involves registering recognized patterns from the current observation $\mathcal{G}_i$, and updating the global graph representation $\mathcal{G}_\mathsf{G}$ based on observed modifications. However, the precision of the descriptor-based matching procedure is insufficient to merge individual graph observations $\mathcal{G}_i$ accurately. Additionally, matching the graphs can be challenging due to numerous smaller structures that occur irregularly between graph observations.

Therefore, a post-analysis of the descriptor-based matches is necessary to ensure that the merged global graph image is not cluttered with small structures that have no value for matching subsequent structures. This analysis involves assessing the optimal inclusion of each node and edge in the global graph representation. In addition to identifying matches not detected by the descriptor-based procedure, it is also necessary to determine the individual differences between two graphs, $\mathcal{G}_i \leftrightarrows \mathcal{G}_\mathsf{G}$, to deduce the most suitable embedding strategy for each edge and node into the global

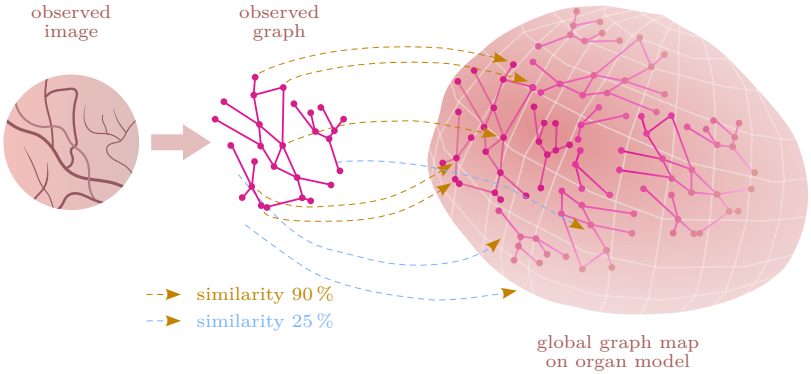


Figure 6.10: To build a global graph representation, a geometry model is used, which enables the geometry update step. This step updates the spatial attributes of the represented pattern, ensuring that the graph attributes remain up-to-date and improving matching for subsequent observations. Moreover, the spatial attribute update step facilitates the identification of newly observed graph patterns not yet represented in the global graph representation. Identifying new observations is critical to update the global representation accurately and to avoid overloading it with unreliable information.

graph representation $\mathcal{G}_\mathsf{G}$. Therefore, a graph-editing procedure is implemented, which transforms the respective graphs into each other, exposing all differences and similarities between the graphs under consideration.

The proposed global graph representation $\mathcal{G}_\mathsf{G}$ is based on descriptor-based similarity registrations. Matches between the current observation and the 3D global graph representation allow for adapting the model geometry of the global graph to the current observation. Given the updated geometry relations, spatially dependent feature descriptors can be updated accordingly. As a result, the updated descriptors facilitate accurate graph analysis by incorporating Euclidean information in the embedding strategy. This enables the identification of differences between the current observation and the represented graph pattern, and enhances the ability to detect all changes in the model representation. In the following Section 6.3.1, the initialization of a global graph representation $\mathcal{G}_\mathsf{G}$ and the corresponding geometry adaptation for a new observation $\mathcal{G}_i$ is proposed. Subsequently, in Section 6.3.2, the graph embedding strategy is presented, effectively expanding the graph to include new data.

6.3.1 Mapping and Geometry Adaption of 3D Model

The adaptability of a triangle mesh enables the mapping of individually observed graph patterns using inverse rendering, as detailed in

$$^{\odot}\mathscr{R}_{\boldsymbol{\phi}_{\mathrm{cam},0}}(\mathbf{M}, \boldsymbol{p}) \mapsto \left[\boldsymbol{P}_{\mathcal{G}_i}, \mathbf{N}_{\mathcal{G}_i}\right] \ . \tag{6.10}$$

This process enables the determination of the spatial positions $\boldsymbol{P}_{\mathcal{G}_i}$ and normal directions $\mathbf{N}_{\mathcal{G}_i}$ of the intersections between nodes and the surface of the geometry model $\mathbf{M}_\mathrm{G}$. As shown in Figure 6.11, the back projection of an initial graph $\mathcal{G}_{i=0}$ onto a unit sphere can be achieved through the application of (6.10). Upon initialization of the spatial representation of the global graph $\mathcal{G}_\mathsf{G}$ on the mesh surface, descriptor-based matching is employed to identify matches between new observations $\mathcal{G}_i$ and the global graph representation $\mathcal{G}_\mathsf{G}$, as depicted in Figure 6.10.

The objective function for supervising the adaptation of geometry is defined on the image plane to align the corresponding graph patterns given by $\mathcal{G}_i \leftrightarrow \mathcal{G}_\mathsf{G}$.

To achieve this, the node points in the 3D world graph are first projected onto the image plane. However, transferring surface points to the image plane directly using the proposed rendering function is not feasible, as it would lead to a discretized representation in the image matrix, thereby compromising differentiability across pixel locations. To address this issue, the node pairs that match between $\mathcal{G}_i$ and the global graph $\mathcal{G}_\mathsf{G}$, denoted as $\boldsymbol{P}^{\mathbf{m}}_{\mathcal{G}_\mathsf{G}}$, are projected onto the image plane according to the analytical camera model (2.5). The projection results in matching node pairs defined on the image plane $\boldsymbol{p}_{\mathcal{G}_i} \leftrightarrow \boldsymbol{p}_{\mathcal{G}_\mathsf{G}}$ without discretization, which ensures differentiability across pixel locations. This procedure is identical to the pose reconstruction method

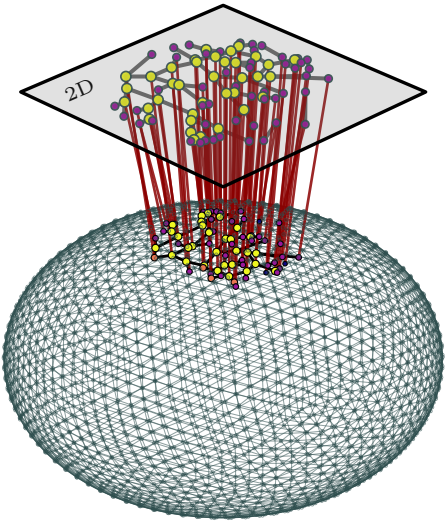


Figure 6.11: Pattern alignment and registration of image planes onto the model surface.

employed in Section 4.2.2 for correlating corresponding node pairs on the image plane. Here, it allows for the formulation of the geometry adaptation

$$\mathbf{V}_{\mathrm{G}}^{\star} = \mathbf{V}_{\mathrm{G},0} + \arg\min_{\Delta\mathbf{V}} \sum_{i=t-h}^{t} \left(\boldsymbol{p}_{\mathcal{G}_i} - \boldsymbol{p}_{\mathcal{G}_{\mathsf{G}}} \right)^2 + \mathcal{L}_{\mathrm{edg}}(\mathbf{M}) + \mathcal{L}_{\mathrm{lap}}(\mathbf{M}) \ , \tag{6.11}$$

where the regularization losses, described in Section 3.1.2, are exploited to promote desirable scaling characteristics in the resulting geometry adaptation.

The matches $\mathbf{m}_{\mathcal{G}_i \leftrightarrow \mathcal{G}_{\mathsf{G}}}$ between partially matching patterns provide the necessary information on how to adjust the surface geometry of the model $\mathbf{M}_{\mathrm{G}}$ to achieve spatially coinciding patterns. As previously outlined in Section 6.2.2, the geometry adaptation can replicate the relevant spatial ratios, at least qualitatively, when a sufficient number of matches $\mathbf{m}_{\mathcal{G}_i \leftrightarrow \mathcal{G}_{\mathsf{G}}}$, is available. In this process, mismatched patterns are also captured, and the corresponding node positions are updated accordingly. This can be observed in Figure 6.11, where the transferred graph patterns of an initial mapping $\mathcal{G}_0$ onto the model surface are displayed. A subsequent graph observation $\mathcal{G}_{i=1}$ is extracted and registered with respect to the nodes of the graph representation $\mathcal{G}_{\mathsf{G}}$ on the image plane, as shown in Figure 6.12, along with the corresponding Euclidean error deviation defined on the image plane.

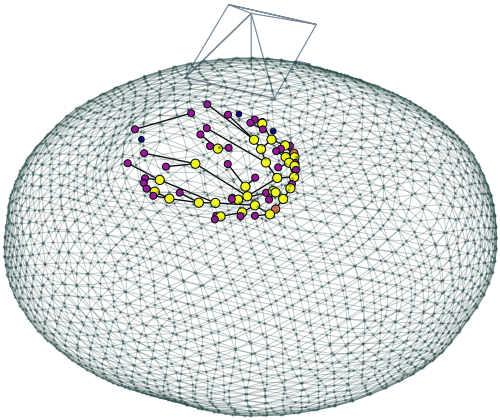


Figure 6.13: The model geometry is adjusted so that the corresponding landmark locations of the current image observation $\mathcal{G}_i$ and the global graph representation $\mathcal{G}_\mathsf{G}$ coincide on the image plane, as shown in Figure 6.12.

The error deviation is minimized to resolve the discrepancy of the corresponding landmark data, and the resulting geometry adaptation is shown in Figure 6.13. Therein, it is observed that the unmapped nodes are correctly entrained by the surrounding node matches, demonstrating the ability of the method to update spatial node and edge features based on the updated model geometry. The proposed method leverages the spatial proportions and distances of patterns to differentiate between a disturbed graph $\mathcal{G}^*$ and a target graph $\mathcal{G}$. In this way, the dependence on distance does not contradict the deformation-tolerant processing. The geometry reconstruction of the 3D model uniformly adapts the non-matching nodes along the deformed surface mesh, creating feasible relations that accurately represent the deformed scene.

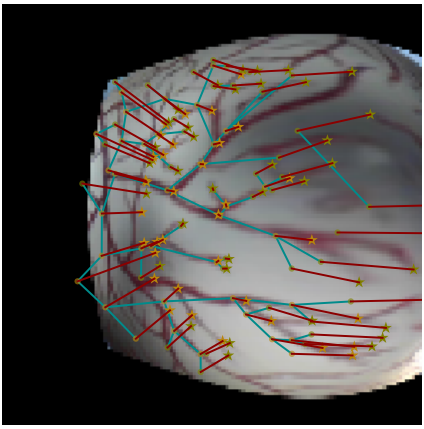


Figure 6.12: Pattern registration of the node location of the current graph observation $\mathcal{G}_i$ with the attributes of the global graph $\mathcal{G}_\mathsf{G}$ on the image plane.

6.3.2 Graph Editing Paths

The remaining task is to identify and update the global graph representation with the current graph data that is not yet included in the global graph representation. For this purpose, the error-tolerant editing distance method, as outlined in [5, 60, 104], is adapted. The fundamental concept behind the edit-bipartite graph alignment is to find a modification policy, denoted as $\pi(\tilde{\mathcal{G}}) \rightarrow \mathcal{G}$, that can transform a noisy graph, $\tilde{\mathcal{G}}$, into a target graph, $\mathcal{G}$, with minimal modifications. By means of this strategy, dissimilarities between the present graph $\mathcal{G}_i$ and global graph representation $\mathcal{G}_\mathsf{G}$ are identified, facilitating the inclusion of the latest observations from $\mathcal{G}_i$ that are not yet added to the global graph $\mathcal{G}_\mathsf{G}$. In other words, this enables the update of the global graph $\mathcal{G}_\mathsf{G}$ with the most recent observations from $\mathcal{G}_i$ that are currently missing. A modification policy, denoted as π, defines a general graph-editing policy capable of transforming any noisy graph $\tilde{\mathcal{G}}$ into a target graph $\mathcal{G}$ by employing a sequence of editing actions a. These actions encompass insertion, deletion, and substitution of nodes and edges [3].

This work discusses editing operations primarily for node operations, as the transition of any edge information is included by considering the edge information as a pair of nodes $e = \{u, v\}$ in $\mathcal{G}$. Thus, the transition of the edge is inherently implied by the corresponding node pair operations. To establish a general notation, the deletion of a node $\tilde{w}$ in $\tilde{\mathcal{G}}$ is represented by $(\tilde{w} \rightarrow \varnothing)$. The reverse notation $(\varnothing \rightarrow \tilde{w})$, in $\tilde{\mathcal{G}}$ refers to the insertion of new node information $\tilde{w}$ into $\tilde{\mathcal{G}}$. Additionally, the substitution of a node $\tilde{u}$ in $\tilde{\mathcal{G}}$ with any chosen node w in $\mathcal{G}$ is denoted as $(\tilde{u} \rightarrow w)$.

The graph editing concept [3] is adapted in this work to address the problem at hand by searching for the policy $\pi(\mathcal{G}_\mathsf{G}) \rightarrow \mathcal{G}_i$ to identify the differences necessary for updating $\mathcal{G}_\mathsf{G}$. However, the complexity of this problem is reduced by considering the following aspects: Firstly, the complexity and redundancy can be reduced by directly assigning each descriptor-based node match as a substitution operation. This means that editing operations for matching node pairs are assigned as substitutions

$$\mathbf{m} \mapsto \{w \in \mathcal{G}_\mathsf{G} \rightarrow \tilde{w} \in \mathcal{G}_i \| \, (w, \tilde{w}) \in \mathbf{m}\} \ , \tag{6.12}$$

and are eliminated from the remaining search space for the corresponding editing policy. If a substitution operation is chosen as an editing strategy from the remaining set, it pertains to a match that has not yet been determined by the descriptor-based matching procedure.

Secondly, the scope of the graph editing search is limited to the currently visible region in the global graph. This exclusion is realized through the design of the mask $\mathbb{M}_{\text{matched}}$ by the convex hull

$$\mathbb{M}_{\text{matched}} = \texttt{ConvexHull}\left(\mathbf{m}_{\mathcal{G}_\mathsf{G}}\right) \ , \tag{6.13}$$

which includes all node positions $\boldsymbol{P}_\mathbf{m} \in \mathcal{G}_\mathsf{G}$ of the currently matched nodes in the global graph. The mask $\mathbb{M}_{\text{matched}}$ is designed to exclude all non-visible regions and

graph patterns outside the surface space of the matched nodes in $\mathcal{G}_\mathsf{G}$ from the editing analysis.

As a result, the search for the transfer policy is restricted to the matching graph entities in $\tilde{\mathcal{G}} = \mathcal{G}_\mathsf{G}^\ominus$, where $\mathcal{G}_\mathsf{G}^\ominus$ is the restricted global surface defined by the mask (6.13). In this way, the overall problem is reduced to the problem of determining $\pi(\mathcal{G}_\mathsf{G}^\ominus) \rightarrow \mathcal{G}_i$, where $\pi(\mathcal{G}_\mathsf{G}^\ominus)$ is the pre-determined editing set policy that includes all substitutions already given by the descriptor-based matches.

Finally, the editing policy is derived by examining each node pair $u, v \in U$ in $\mathcal{G}$ and $\tilde{u}, \tilde{v} \in \tilde{U}$ in $\mathcal{G}_\mathsf{G}^\ominus$ separately, for $w \in \{u, v\}$ based on the following conditions:

- If node $w \in U$ in $\mathcal{G}$ and $\tilde{w} \in \tilde{U}$ in $\mathcal{G}_\mathsf{G}^\ominus$ are present in both graphs, a node substitution is implied.
- If there is a node $w \in U$ in $\mathcal{G}$, but no corresponding node $\tilde{w} \in \tilde{U}$ is found in $\mathcal{G}_\mathsf{G}^\ominus$, then a node insertion ($\varnothing \rightarrow w$) must be performed in $\mathcal{G}_\mathsf{G}^\ominus$.
- Conversely, if there is a node $\tilde{w} \in \tilde{U}$ in $\mathcal{G}_\mathsf{G}^\ominus$, but no node $w \in U$ is found in $\mathcal{G}$, then a node deletion ($\tilde{w} \rightarrow \varnothing$) must be performed in $\mathcal{G}_\mathsf{G}^\ominus$.

However, the outlined approach is ambitious due to the multitude of feasible operation strategies that can be specified in an editing set $\Gamma\left(\mathcal{G}_\mathsf{G}^\ominus, \mathcal{G}\right) = \{\pi_1, \cdots, \pi_k\}$ of feasible adaptation policies.

The established formulations in [3, 5, 60, 104] assume the existence of ideal graphs, where related nodes and edges have the same exact graph representation in their spatial distribution and feature representation. However, in this work, it is essential to consider the graphs as noisy versions of each other due to changes and various influences resulting from subsequent observations. The heavy reliance on ideal graphs can create issues when attempting to identify a suitable editing strategy for noisy graphs. For example, it may result in the deletion of the whole existing information and the insertion of all new information rather than accurately evaluating the similarities between the graph patterns.

To avoid ambiguity, costs are assigned to each operation to discourage excessive changes and obtain the modification policy with the lowest modification cost c. To account for spatial deviations in the matching analysis, the cost design considers the Euclidean distances after the geometry adaptation (6.11). Patterns that are close to each other are supported by low costs and are detected as matching patterns. In the cost design, a recurrence rate is introduced to weight patterns based on retrievable recurrence in observed data. Therefore, the recurrence rate of an edge

$$a_{\text{rec}} = \frac{n_{\text{detect}}}{n_{\text{view}}} \ , \tag{6.14}$$

is assigned to each edge and indicates how often, on average, the edge is rediscovered. Specifically, n_{detect} represents the total number of detections of the edge, while n_{view} represents the total number of times the area covered by the edge is visible within the current mask defined in (6.13). In the cost design it is aimed to assign

lower costs to recurring, well-recognizable structures and higher costs to unreliable structures by taking into account the recurrent detection rates. This approach helps to prioritize the more reliable structures and avoid less reliable structures. Indeed, the edge recurrence rate a_{rec} is defined for all individual edge patterns. However, the editing procedure and cost design are defined on a node level. To relate the edge recurrence rate to the respective adjacent node information, the embedding function defined in (6.5) is deployed, such that the recurrence rate can be integrated into the similarity comparisons and node-based cost design by expanding the respective node descriptors: $\boldsymbol{d}$ += $\boldsymbol{d}_{\text{rec}}$. Finally, the substitution of a node ($w \to \tilde{w}$) is penalized by the Euclidean distances on the mesh surface and the expanded descriptor-based similarity measure through

$$c_{\tilde{w},w} = \|\tilde{w}_{xy} - w_{xy}\| + \|\boldsymbol{d}_{\tilde{w}} - \boldsymbol{d}_{w}\| \ . \tag{6.15a}$$

The substitution cost is designed to consider the variations in sensitivity based on the current geometry prediction $\mathbf{M}_i^\star$ and Euclidean ratios, which ensures a robust and accurate solution to the graph transition process. The cost design supports correspondences between points at close distances, while those at greater distances are more likely to be considered as new individual nodes. To ensure that new node insertions in $\mathcal{G}_{\mathsf{G}}^{\ominus}$ do not introduce redundant information that already exists in its own neighborhood, node insertion actions are penalized by

$$c_{\varnothing,w} = \frac{1}{\sum_i^k \|w - \mathcal{N}_{\mathcal{G}_{\mathsf{G}}^{\ominus},i}(w)\|}. \tag{6.15b}$$

This penalty is based on the Euclidean distance between the new node w and its k spatially closest graph nodes in $\mathcal{G}_{\mathsf{G}}^{\ominus}$, where $\mathcal{N}_{\mathcal{G}_{\mathsf{G}}^{\ominus}}(w)$ evaluates the sum over the k neighboring nodes. The cost for node insertion is calculated as the reciprocal of this sum, resulting in a higher cost for nodes that are very similar to their neighbors. This cost design aims to discourage unnecessary insertions and preserve the structure of the graph.

For a node deletion, ($\tilde{w} \to \varnothing$), the norm of the respective descriptor information is incorporated into the cost design by

$$c_{\tilde{u},\varnothing} = \|\boldsymbol{d}_{\text{struc},\tilde{w}}\| + \|\boldsymbol{d}_{\text{rec},\tilde{w}}\| \ . \tag{6.15c}$$

The norm of the descriptor takes into account the structural constraints and their impact on structural changes induced by various actions. It assigns a heavier penalty to nodes with higher degrees in the graph and their first-order neighborhoods. Moreover, the norm of the respective recurrence rate is considered to account for the deletion of reliably detected graph structures by deleting the respective graph node w.

Finally, the optimal graph transition from $\mathcal{G}_{\mathsf{G}}^{\ominus}$ to $\mathcal{G}$ is formally specified by the problem description

$$\pi^\star(\mathcal{G}_{\mathsf{G}}^{\ominus}, \mathcal{G}) \to \mathcal{G} = \min_{(\mathsf{a}_1,\dots,\mathsf{a}_k)\in\Gamma\left(\mathcal{G}_{\mathsf{G}}^{\ominus},\mathcal{G}\right)} \sum c\left(\mathsf{a}\left(\mathcal{G}_{\mathsf{G}}^{\ominus}\right)\right) \ , \tag{6.16}$$

which minimizes the sum of the modification costs for a sequence of feasible adaptation strategies $(\mathsf{a}_1, \ldots, \mathsf{a}_k) \in \Gamma\left(\mathcal{G}_{\mathsf{G}}^{\ominus}, \mathcal{G}\right)$, respecting the cost design (6.15) with c, and including the respective individual weighting of the separate costs. The solution space is defined by the sequence of feasible adaptation strategies $(\mathsf{a}_1, .., \mathsf{a}_k) \in \Gamma\left(\mathcal{G}_{\mathsf{G}}^{\ominus}, \mathcal{G}\right)$. Consequently, the deduced objective is to solve problem (6.16) efficiently.

Algorithm 3: Graph editing problem based on the $A^\star$-algorithm

Data: Noisy input graph $\mathcal{G}_{\mathsf{G}}^{\ominus}$, target graph $\mathcal{G}$
Result: $\pi^\star(\mathcal{G}_{\mathsf{G}}^{\ominus}, \mathcal{G}) \rightarrow \mathcal{G}$;
e.g. $\pi^\star = \{\tilde{u}_1 \rightarrow u_4, \tilde{u}_8 \rightarrow \varnothing, ..., \varnothing \rightarrow u_5\}$;
; `/* Optimal editing policy to transfer the noisy input graph to given target graph. */`
Initialize the open action set $\mathtt{a}_{\mathrm{OPEN}}$;
For each node $v \in \mathcal{G}$, insert substitution $\{\tilde{u}_1 \rightarrow v\}$ in the open action set $\mathtt{a}_{\mathrm{OPEN}}$;
Add all deletions $\{\tilde{u}_1 \rightarrow \varnothing\}$ to the open action set $\mathtt{a}_{\mathrm{OPEN}}$;
for *match m in* **m do**
  $\pi^\star = \arg\min_{\pi \in \mathtt{a}_{\mathrm{OPEN}}} \sum_{\mathsf{a} \in \pi} c_{(a)} + l(c\,(a_{\mathrm{leave}})\,p)$;
  if $\pi^\star$ *is a complete editing policy* **then**
    Return $\pi^\star$ as the solution
  end
  $\pi^\star = \{\tilde{u}_1 \rightarrow u_{i1}, ..., \tilde{u}_k \rightarrow u_{ik}\}$;
  if $k \leq |\tilde{U}|$ **then**
    for each $u \in U \setminus \{u_{i1}, ..., u_{ik}\}$, insert $\pi^\star \cup \{\tilde{u}_{k+1} \rightarrow u\}$ into $\mathtt{a}_{\mathrm{OPEN}}$;
    add $\pi^\star \cup \{\tilde{u}_{k+1} \rightarrow \varnothing\}$ to $\mathtt{a}_{\mathrm{OPEN}}$;
  else
    add $\pi^\star \cup \bigcup_{v \in U \setminus \{u_{i1}, ..., u_{ik}\}} \{\varnothing \rightarrow v\}$ to $\mathtt{a}_{\mathrm{OPEN}}$;
  end
end

Solving the graph-editing problem using the A* algorithm. The graph transition problem (6.16) is a combinatorial optimization problem that is known to be NP-hard, making it computationally demanding to find an exact solution for large-scale graphs. In this work, an A* algorithm is used, which provides a best-first search algorithm commonly used in path finding and graph traversal. In the approach of this work, the graph nodes are defined as states and the costs as a heuristic function. This approach yields an efficient and effective solution for the graph editing problem when compared to other heuristic methods, such as Munger's algorithm and the Hungarian algorithm [120].

To solve the problem (6.16), the $A^\star$ algorithm, based on [37], is employed. This algorithm finds the optimal policy $\pi^\star(\mathcal{G}_{\mathsf{G}}^{\ominus}, \mathcal{G}) \rightarrow \mathcal{G}$ through a heuristic search of the

solution space. The method is based on the concept presented in [104] and utilizes a search tree that is constructed dynamically at run-time [148]. The overall solution process, adapted to problem (6.16), is outlined in Algorithm 3. During the process, nodes are processed in the given order, and the corresponding actions are applied cumulatively. Subsequent nodes are created during the processing and considered in the search tree.

The `OPEN` portion of the algorithm refers to the set of nodes that have not yet been fully processed, while the search tree expands by evaluating subsequent nodes. In this approach, the cumulative costs from the initial a_0 to the current modification a_t are represented as $g_{\mathrm{A}\star}(\mathsf{a})$, while $h_{\mathrm{A}\star}(\mathsf{a})$ is used to estimate the costs from a_t to the final state, where $\mathcal{G}_{\mathsf{G}}^{\ominus} \equiv \mathcal{G}$. During the iterative solving process, the states are classified into one of the following three categories:

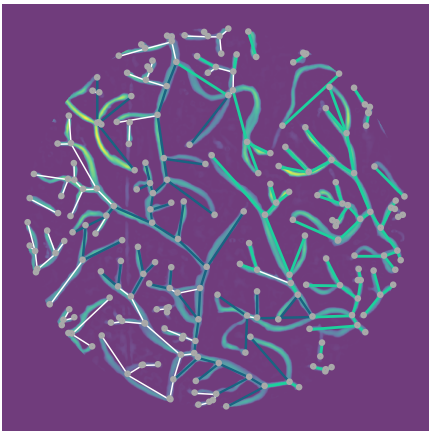
- *unknown states*; which are not yet encountered during the search, and as such, no path is known to the corresponding nodes. Initially, every node except the starting node is classified as an unknown state.
- *known states*; to which a (suboptimal) path is known. The known nodes are stored along with the respective cost value $g_{\mathrm{A}\star}(\mathsf{a}) + h_{\mathrm{A}\star}(\mathsf{a}))$ in the so-called `OPEN` List. The most promising node is selected and explored from this list for the next iteration.
- *checked states*; to which the shortest path is known. The finally examined nodes are registered in the so-called `CLOSED` list to avoid repetitive node inspections.

The algorithm terminates upon the final examination of the target state, and the found path is reconstructed using the predecessor output state. If the `OPEN` list is empty, the algorithm terminates immediately, indicating that no solution was found. Therefore, the A* algorithm aims to find the lowest cost solution for $g_{\mathrm{A}\star}(\mathsf{a}) + h_{\mathrm{A}\star}(\mathsf{a})$ in the heuristically guided search.

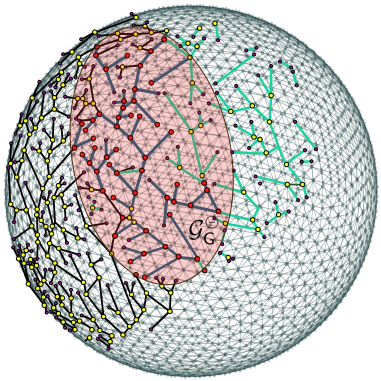
The resulting graph editing policy is depicted in Figure 6.14, where all nodes labeled in ● red represent descriptor-based matches and thus are excluded from the editing search. Figure 6.15 illustrates the resulting optimal editing policy for transferring the illustrated graphs. The nodes and edges information labeled in ● blue, ● red, and ● purple represent substitution, deletion, and insertion operations, respectively. In conclusion, the update of the global graph based on the established editing policy adheres to the distinct editing policy $\pi^\star(\mathcal{G}_{\mathsf{G}}^{\ominus}) \rightarrow \mathcal{G}_i$. Any insertion and deletion operations can be directly transferred from editing $\mathcal{G}_{\mathsf{G}}^{\ominus}$ to $\mathcal{G}_{\mathsf{G}}$. This ultimately leads to the updating of $\mathcal{G}_{\mathsf{G}}$ by the current observation.

6.3.3 Global Graph Update

The resulting optimal graph policy $\pi^\star(\mathcal{G}_{\mathsf{G}}^{\ominus}) \rightarrow \mathcal{G}_i$ yields an exact copy of the target graph. Indiscriminately adding or deleting information from the global graph can



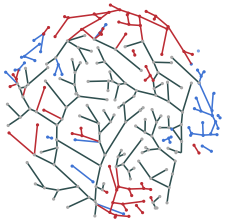
(a) Current image observation $\mathcal{G}_i$.



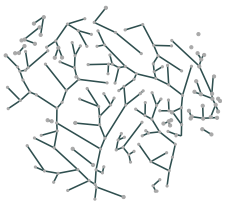
(b) Updated global graph $\mathcal{G}_{\mathsf{G}}$. The truncated graph $\mathcal{G}_{\mathsf{G}}^{\ominus}$ is highlighted in the red oval area.

Figure 6.14: The global graph $\mathcal{G}_{\mathsf{G}}$ is updated with newly observed graph structures $\mathcal{G}_i$. Nodes with descriptor-based matches define the search space for editing the world graph, and are highlighted in red. The truncated graph $\mathcal{G}_{\mathsf{G}}^{\ominus}$ defines this search space. Common edge structures recognized in both $\mathcal{G}_i$ and $\mathcal{G}_{\mathsf{G}}^{\ominus}$ are highlighted in blue, while newly added structures are highlighted in green.

(a) Initial graph $\mathcal{G}_{\mathsf{G}}^{\ominus}$.



(b) Graph transition $\pi^{\star}(\mathcal{G}_{\mathsf{G}}^{\ominus}) \to \mathcal{G}_i$.



(c) Observed target graph $\mathcal{G}$.

Figure 6.15: Graph editing: In (a) and (c) two different graph observations $\mathcal{G}_{\mathsf{G}}^{\ominus}$ and $\mathcal{G}$ are given. The corresponding graph transition is shown in (b), with deletions highlighted in red, insertions in blue, and node and edge substitutions in grey.

lead to significant robustness issues, particularly in situations where data uncertainty is present. To address this challenge, the recurrence rate as defined in (6.14) is deployed for the global graph update. Rather than naively overriding the graph representation with the determined policy, the global graph is updated by adjusting the respective recurrence rates in $\mathcal{G}_{\mathsf{G}}$ based on $\pi^{\star}$.

The recurrence rate n_{view} += 1 for the viewed pattern is incremented by one for all edges whose corresponding nodes are visible within the current view specified by the mask (6.13). The recurrence rates specific for re-detected patterns are adjusted for each edge individually. When an edge is re-detected based on a substitution operation, the respective n_{detect} += 1 is increased. Conversely, if the edge is not re-detected but a deletion is considered, the corresponding recurrence value n_{detect} −= 1 is decreased. For the initial pattern observation, the respective detection number $n_{\text{detect}} \equiv 1$ is initialized within the graph. To avoid relying solely on a single observation in the initial mapping process, the recent correspondence of the information of the nodes on the mesh is mapped on the 3D surface for each new observation, including recurrently detected node patterns. Then, the node position is updated through the weighted average

$$\boldsymbol{P}^{\star}_{\mathcal{G}_i} = a_{\text{upd,rec}} \boldsymbol{P}_{\mathcal{G}_i} + (1 - a_{\text{upd,rec}}) \boldsymbol{P}_{\mathcal{G}_{\mathsf{G}}} \ , \tag{6.17}$$

where based on the update rate $0 \leq a_{\text{upd,rec}} \leq 1$, the node positions $\boldsymbol{P}_{\mathcal{G}_{\mathsf{G}}}$ of the global graph $\mathcal{G}_{\mathsf{G}}$ are iteratively updated by the corresponding latest observation given by $\boldsymbol{P}_{\mathcal{G}_i}$. This approach ensures a reliable graph representation that gradually becomes more accurate over time.

6.4 Summary & Conclusion

The central research objectives of this chapter is summarized as follows: registration of corresponding graph features while addressing intraoperative constraints, such as a deformable scene and temporarily blocked pattern observations. The proposed descriptor design offers a quasi-deformation-invariant similarity measure for structural graph information, facilitating an efficient matching procedure. To realize a diversification of the solution space and to avoid a high dependence on individual matches, each descriptor is matched individually in the proposed descriptor-based matching procedure.

Due to mutually independently determined matches, many outliers may be present in the resulting solution set. However, conventional outlier classification methods found in the literature may not be sufficient in the presence of deformation, resulting in all descriptor-based matches being classified as outliers. To address this issue, a new outlier classification concept called SbOR was introduced. This concept utilizes pathological structures as a reference point, and any match that requires resectioning visible vascular structures is classified as an outlier. The efficacy of this method was demonstrated using a synthetically generated and distorted dataset. The proposed descriptor-based matching and SbOR outlier removal technique are generalizable to any pathological domain and provide a significant contribution to landmark-based orientation concepts for intraoperative conditions.

A 3D model embedding has been proposed to improve the field of intraoperative orientation. The embedding integrates subsequently observed patterns into a global

representation, allowing a current pattern observation to be matched with all previously observed patterns in a single query. This approach is particularly useful in scenarios, where vision is temporarily impaired, as it enables comparison with all previous observations when vision is restored, rather than conducting sequential and time-consuming pattern matching across recent observations.

The embedded model representation can update the model geometry to the current conditions, which helps ensure robust pattern matching for future observations. Additionally, the geometry reconstruction adaptation ensures feasible relations that represent the deformed scene by uniformly entraining the non-matched nodes along the deformed surface mesh.

To update the global graph representation based on the latest pattern observations, a graph editing technique is applied to accurately identify differences between the new graph and the global graph. This approach focuses on updating only the newly added elements, which helps to avoid introducing ambiguities into the global graph. As a result, the information content of the global graph is preserved, which is crucial for ensuring a continuous progression. Although the editing algorithm used in this procedure is well-established in the literature, it is often too complex to be readily applied. Nevertheless, in this work, the algorithm has been successfully adapted to the requirements of the task by limiting its complexity. This has been achieved by utilizing descriptor-based matches as a predefined editing policy and by efficiently restricting the editing space.

In summary, this chapter offers the following contributions: a tailored descriptor design for handling intraoperative challenges, a related outlier classification utilizing pathological constraints, and an inclusive view of pattern representation in a global model to update geometry based on new observations. A 3D model representation was developed to register the associated global graph information. This 3D model maintains Euclidean ratios for all spatial graph information, which is updated by adjusting the model geometry as needed.

7 Intraoperative Navigation and Scene Reconstruction

Intraoperative navigation and scene reconstruction are subject to enormous challenges under real-life conditions. Despite this, there remains a lack of comprehensive solutions in the literature. Thus, new concepts are required to establish a robust workflow for reconstruction purposes. In the previous chapters, new methodological solutions for specific problems were developed. Building on that, this chapter aims to holistically establish a comprehensive localization and reconstruction process that addresses real-world complexity for intraoperative navigation and scene reconstruction.

By integrating and combining all methods presented in this thesis so far, a comprehensive approach strategy is formulated, leveraging the unique strengths of each technique to address the complex intraoperative scene reconstruction problem holistically. The graph-based landmark orientation provides robust orientation through generalized pattern matching, regardless of initial location. Building on this, the gradient-based reconstruction approach improves pose accuracy and reconstructs geometry and texture information. By utilizing the graph-based orientation as a robust initial pose for the gradient-based reconstruction strategy, the resulting method compensates for the gradient-based method's high sensitivity to initial conditions. In this way, the proposed combination leverages the robustness of the matching method and the accuracy of the reconstruction method, providing a more reliable and accurate approach to the problem at hand.

The proposed concept integrates individual processes to capture ambiguities and enable the reconstruction of pose, geometry, and texture under intraoperative conditions. However, the graph-based landmark correspondences are insufficient for a holistic reconstruction process to achieve the desired accuracy since the corresponding node space might be distributed sparsely, and some node matches might even be obsolete. However, a pure image comparison may lead to complex stability problems, as described in Chapter 3. For this purpose, the extracted skeletonized pattern information is exploited, providing increased information gain for reconstruction purposes while ensuring high robustness given an appropriate loss formulation.

To address a holistic intraoperative reconstruction problem, this chapter is organized as follows. In Section 7.1, the overall reconstruction pipeline is presented, and

the proposed concept is validated through two successive experiments. The first experiment, discussed in Section 7.2, involves reconstructing the camera's pose and comparing it with indoor localization measurements by attaching markers to the endoscopic camera. In the second experiment, outlined in Section 7.2.2, the focus is on reconstructing the object's geometry using a fringe projection sensor. The experiment involves manipulating the volume of a balloon by inflating and deflating it with water, capturing changes in the object's geometry, and then comparing the image-based geometry reconstruction to the measured geometry data.

Finally, in Section 7.3, the reconstruction concept is extended by incorporating external depth map measurements into the reconstruction objective. The simultaneous integration of depth map data and image data demonstrates the comprehensive understanding and generalizability of the proposed reconstruction approach. From a methodological perspective, this concept provides a solution for the analysis of in-plane deformation and strain ratios. Moreover, the proposed reconstruction concept of in-plane deformation presents a high potential for the medical field of applications for multi-sensory data classification.

7.1 Holistic Reconstruction Pipeline

The initialization and update steps are presented separately in the following subsections. Therefore, a graphical flowchart representation has already been presented for the outline of the work in Chapter 1.4, for initialization and consecutive update steps in Figure 1.7.

7.1.1 Initialization

In the initial step, graph $\mathcal{G}_i$ is extracted from the current image observation $\boldsymbol{\mathcal{I}}_i \mapsto \mathcal{G}_i$, as discussed in Chapter 5. The inverse rendering (4.8) facilitates the remapping of the graph nodes onto the geometry mesh $\mathbf{M}_\mathrm{G}$. The graph is re-projected from an initial camera location $\boldsymbol{\phi}_{\mathrm{cam},0}$, with the assumption that the perspective captures the entire model, such that all back projections have feasible intersection points for all pixels with the geometry mesh $\mathbf{M}_\mathrm{G}$. Then, consecutive observations can be aligned relative to the initial perspective. In addition, the image information $\boldsymbol{\mathcal{I}}_i$ can be mapped to the texture model $\mathbf{M}_\mathrm{T}$ based on the forward rendering process outlined in (2.26). This results in a texture-based optimization problem, as formulated in (3.14).

However, as discussed in Section 3.3, relying solely on monitoring the intensity of the rasterized image data can lead to significant robustness issues during the reconstruction of a synthetic bladder model. This can result the convergence to a local minimum, which is exacerbated by factors such as gloss effects, image noise, and distortion effects in real-world images, particularly for intraoperative data. To enhance

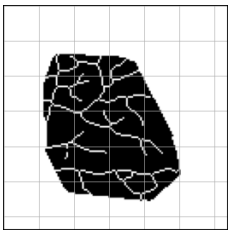
the information content of the texture image for intraoperative scene reconstruction, the skeletonized vascular image information $\boldsymbol{\mathcal{I}}_{\text{skel}}$ is incorporated. This approach incorporates skeletonized vascular image information to facilitate pattern-based similarity alignment between real-world image data and the model representation, while suppressing noise effects and allowing for the use of various loss formulations.

The image pattern can be reconstructed in a texture model by means of an image comparison following (3.15). However, unfavorably recorded pattern information, such as blurred image capture, can result in the omission of critical information from the global texture model and the inclusion of disturbed texture information. Additionally, if the previous pose or geometry reconstruction deviates slightly from the actual position, a mismatch can be quickly induced into the texture model, causing robustness issues to the overall reconstruction process.

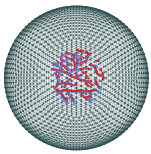
Therefore, the texture is not updated based on an image comparison according to (3.15), but rather, the skeletonized pixel set $\boldsymbol{p}_e$ is assigned to the texture mesh $\mathbf{M}_{\text{T}}$ directly through the inverse rendering process outlined in (4.8), separately for each extracted graph edge e. The corresponding skeleton pixel set $\boldsymbol{p}_e$ is extracted as part of the graph extraction process, which was introduced in Section 5.3.2. This combination of the global graph representation $\mathcal{G}_{\mathsf{G}}$ alongside the skeletonized structure representation $\mathbf{M}_{\text{T}}$ allows for a lightweight pattern analysis. If needed, the corresponding pixel set $\boldsymbol{p}_e$ linked to the edge facilitates a further in-depth analysis of the structural information.

The skeletonized pattern information is represented in the respective vertex grey feature space $\mathbf{C}_{\text{skel}}$ of $\mathbf{M}_{\text{T}}$. Furthermore, the geometry of the texture model $\mathbf{M}_{\text{T}}$ is driven by the geometry mesh $\mathbf{M}_{\text{G}}$ and determined by the mesh subdivision (3.13) as introduced in Section 3.2.1. This allows for the representation of texture with high resolution while simultaneously ensuring manageable complexity. As a result, $\mathbf{M}_{\text{T}}$ serves as a repository of the observed and skeletonized structures, providing the necessary information for scene reconstruction.

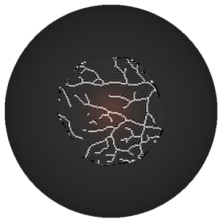
Without prior model knowledge, a unit sphere can be used to initialize the geometry model $\mathbf{M}_{\text{G}}$. However, if prior knowledge is available, for example from MRI geometry reconstruction, it can be utilized to initialize the geometry representation. Despite this approach, challenges may arise, particularly regarding the registration of the preoperative model with intraoperative image data, which can be especially challenging for soft organs. Therefore, the non-trivial task of preoperative data registration is not considered in the scope of this work. Instead, it is proposed that the geometry parameterization $\mathbf{M}_{\text{G}}$ can be initiated from any initial state. Thus, a unit sphere is chosen as an initial template mesh without imposing any constraints on the generality. Figure 7.1a shows an initial texture observation $\boldsymbol{\mathcal{I}}_{\text{skel}}$, while Figure 7.1b presents the corresponding initialized global graph $\mathcal{G}_{\mathsf{G}}$, and Figure 7.1c displays the respective initialized texture model $\mathbf{M}_{\text{T}}$.



(a) Skeletonized image observation $\boldsymbol{\mathcal{I}}_{\text{skel}}$ masked by $\mathbb{M}_{\text{matched}}$ according to (6.13).



(b) Mapped pixel paths from image planes to model surface $\mathbf{M}_{\text{G}}$.



(c) Initialized texture model $\mathbf{M}_{\text{T}}$.

Figure 7.1: Based on a predefined initial camera perspective $\boldsymbol{\phi}_{\text{cam},0}$, the model representation of the global graph model $\mathcal{G}_{\mathsf{G}}$ on the geometry mesh $\mathbf{M}_{\text{G}}$ and the texture model $\mathbf{M}_{\text{T}}$ are initialized by the skeletonized image structure $\boldsymbol{\mathcal{I}}_{\text{skel}}$.

7.1.2 Iterative Model Update

Once the model representation covers initial observations, any further observations can be aligned with respect to the initial camera perspective $\boldsymbol{\phi}_{\text{cam},0}$. Therefore, the pattern in the global graph $\mathcal{G}_{\mathsf{G}}$ and in the texture model $\mathbf{M}_{\text{T}}$ are sequentially exploited for pose reconstruction.

The graph extraction $\mathcal{G}_i$ for the current image observation $\boldsymbol{\mathcal{I}}_i$ is compared to the global graph representation $\mathcal{G}_{\mathsf{G}}$ using the descriptor-based graph matching procedure described in Section 6.1. This process enables a pattern alignment, which is independent of the initial reconstruction guess. Additionally, the descriptor-based graph matching procedure is coupled with deformation-invariant outlier removal. This process produces a set of matches $\mathbf{m}_{\mathcal{G}_i \leftrightarrow \mathcal{G}_{\mathsf{G}}}$ that provides the necessary pattern registration for pose reconstruction. Through the inverse rendering mapping (4.8), the extracted node positions $\boldsymbol{p} \in \mathcal{G}_i$ can be transferred from the image plane to the model surface of $\mathbf{M}_{\text{G}}$. This information is used to formulate the reconstruction objective in terms of 3D Euclidean distances and normal similarities, as shown in (4.10), while leveraging the given match correspondences $\mathbf{m}_{\mathcal{G}_i \leftrightarrow \mathcal{G}_{\mathsf{G}}}$. The deduced optimization objective is formulated analogously to (4.9). This objective is demonstrated to exhibit well-conditioned convergence behavior in Section 4.2.2. However, due to the sparse and error-prone nature of the graph-based landmark representation, the resulting perspective reconstruction $\tilde{\boldsymbol{\phi}}_{\text{cam}}$ must be considered as a preliminary and approximate solution. Thus, while the deduced pose optimization of the form (4.9) is well solvable, inaccuracies in the assignments may result in the corruption of the loss formulation itself, either through incorrect node matches or an erroneous graph extraction process.

To improve the accuracy of the graph-based pose reconstruction and compensate for any associated uncertainties, the pose reconstruction is repeated by exploiting the skeletonized structures provided through the texture model $\mathbf{M}_{\mathrm{T}}$. While the optimization problem can be formulated directly on the image plane by supervising the differences of the pixel intensities $\|\boldsymbol{I}_{\mathrm{skel}} - \mathbf{I}\|$, as shown in (3.16), the convergence properties of this pattern-matching formulation are inferior, as discussed in Section 3.3. When corresponding patterns differ slightly in the initial stages, the optimization can quickly converge to a local minimum in the subsequent iterations. To address this issue, a point-cloud-based pose reconstruction similar to the graph-based pose reconstruction (4.9) is followed. This procedure has been shown to exhibit improved convergence behavior, as demonstrated in Section 3.3. However, unlike the graph-based pose reconstruction, no predefined point registrations are available.

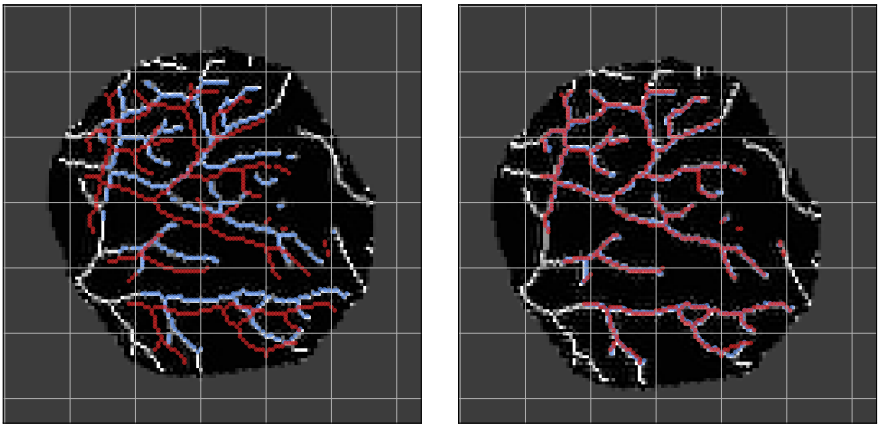
The graph extraction process provides the corresponding pattern curves of all matching edges $\boldsymbol{p}_e$ based on the skeletonized image $\boldsymbol{I}_{\mathrm{skel}}$. Additionally, the established matches $\mathbf{m}_{\mathcal{G}_i \leftrightarrow \mathcal{G}_{\mathsf{G}}}$ enable the reduction of pattern areas to the potentially matchable pixel locations $\boldsymbol{p}_{I\mathbb{M}}$ in the image plane $\boldsymbol{I}_{\mathrm{skel}}$ that fall within the given mask $\mathbb{M}_{\mathrm{matched}}$ as specified in (6.13). To compare the inverse rendered spatial structures of $\boldsymbol{p}_{I\mathbb{M}}$ depicted in $\mathbf{M}_{\mathrm{T}}$, the resulting vertices $\boldsymbol{P}_{I\mathbb{M}}$ and normal data $\mathbf{N}_{I\mathbb{M}}$ must be aligned to the corresponding structure representation, which is represented in the model by $\boldsymbol{P}_{\mathsf{vas}}$ and $\mathbf{N}_{\mathsf{vas}}$.

Thus, for camera reconstruction, the deduced objective is to adjust the camera position $\boldsymbol{\phi}_{\mathrm{cam},i}$ to achieve spatial coincidence between the patterns in the point cloud $\boldsymbol{P}_{I\mathbb{M}}$ and the vascular structure $\boldsymbol{P}_{\mathsf{vas}}$. Unlike the graph-based node correspondences $\mathbf{m}_{\mathcal{G}_i \leftrightarrow \mathcal{G}_{\mathsf{G}}}$, the point clouds $\boldsymbol{P}_{I\mathbb{M}} \leftrightarrow \boldsymbol{P}_{\mathsf{vas}}$ and corresponding normals $\mathbf{N}_{I\mathbb{M}} \leftrightarrow \mathbf{N}_{\mathsf{vas}}$ do not have predetermined point registrations. Therefore, a differentiable error function is required to formulate an optimization objective that expresses the difference between the listed point clouds.

The Chamfer distance is a suitable loss measure that expresses the difference between two point clouds regardless of their relative size. It is defined as

$$\begin{aligned} \mathcal{L}_{\mathrm{chf}}(\boldsymbol{J}, \boldsymbol{Q}) = |\boldsymbol{J}|^{-1} \sum_{(J,Q)\in\Omega_{\boldsymbol{J},\boldsymbol{Q}}} \|J - Q\|^2 \\ + |\boldsymbol{Q}|^{-1} \sum_{(Q,J)\in\Omega_{\boldsymbol{Q},\boldsymbol{P}}} \|Q - J\|^2 \\ \text{with } \Omega_{\boldsymbol{Q},\boldsymbol{J}} = \left\{ \left(J, \arg\min_{Q} \|Q - J\| \right) : J \in \boldsymbol{J} \right\} , \end{aligned} \tag{7.1}$$

where $\Omega_{\boldsymbol{Q},\boldsymbol{J}}$ is the set of node pair combinations (J, Q) where $Q \in \boldsymbol{Q}$ is the closest point to a corresponding point $J \in \boldsymbol{J}$ [101]. The brute force implementation of (7.1) requires up to $|\boldsymbol{J}| \times |\boldsymbol{Q}|$ individual evaluations of point pair combinations. A more efficient implementation of (7.1) can be achieved by using the *Cuda*-based implementation of a *k*-nearest neighbor search [101] in a *kd* tree query. This approach



(a) Model rendering with initial camera pose $\tilde{\boldsymbol{\phi}}_{\text{cam}}$.

(b) Model rendering with adjusted camera pose $\boldsymbol{\phi}^{\star}_{\text{cam}}$.

Figure 7.2: Rendered image with extracted structure based on model texture in ● blue and extracted point cloud based on the structure of the target image in ● red.

reduces computational cost by over an order of magnitude [90]. Additionally, this implementation enables the Chamfer distance to be differentiable with respect to the Euclidean distances of the data points, making it suitable for gradient-based optimization. Although the point assignments $\boldsymbol{\Omega}_{\boldsymbol{Q},\boldsymbol{J}}$ must be recomputed at each iteration, the gradient is not compromised as differentiability is only required within one iteration. Thus, the Chamfer distance (7.1) enables the formulation of the deduced optimization problem

$$\boldsymbol{\phi}^{\star}_{\text{cam}} = \arg\min_{\boldsymbol{\phi}} \sum_{i=t-h}^{t} \mathcal{L}_{\text{c}}\left(\begin{bmatrix}\boldsymbol{P}_{\text{vas}}\\ \boldsymbol{N}_{\text{vas}}\end{bmatrix}, \begin{bmatrix}{}^{\odot}\mathscr{R}^{\boldsymbol{P}}_{\boldsymbol{\phi}}\left(\mathbf{M}, \boldsymbol{p}_{\mathcal{I}\mathbb{M},i}\right)\\ {}^{\odot}\mathscr{R}^{\mathbf{N}}_{\boldsymbol{\phi}}\left(\mathbf{M}, \boldsymbol{p}_{\mathcal{I}\mathbb{M},i}\right)\end{bmatrix}\right). \tag{7.2}$$

The iterative search for the solution of (7.2) starts by initializing it with the pose reconstruction obtained from the initial sparse graph-based pose reconstruction given by the initial graph-based pose reconstruction $\tilde{\boldsymbol{\phi}}_{\text{cam}}$. The accuracy of the point cloud-based pose reconstruction is demonstrated in Figure 7.2b, which shows the coincident patterns in high detail compared to the initial condition in Figure 7.2a.

7.1.2.1 Deformation

Subsequently, the camera perspective and deformation of the observed object are reconstructed from the monocular camera image. However, distinguishing between visual changes caused by changes in the camera's position and those caused by object

deformation is ambiguous without using additional sensors to capture the camera trajectory. The analytical camera model (2.5) only applies to visual changes caused by perspective changes. Object deformations, on the other hand, are generally unconstrained and do not follow any specific model. For example, changes in the size of patterns in the image plane could be due to a change in camera perspective or a self-scaling deformation. Either way, camera movements generally proceed much faster than deformation. As a result, the following empirical approach can be employed to separate the influence of each on the observed image: First, changes in the observed patterns on the image plane are resolved by adjusting the camera perspective. Then, any remaining inconsistencies not accounted for by the camera model are resolved by adjusting the model geometry.

In this manner, the discrepancy between observed and target patterns is attributed to deformation effects. The spatial graph model is adjusted in advance using node-based assignment reconstructions, as opposed to the geometry update covered in Section 6.3. However, relying on a limited number of point assignments for the similarity objective is insufficient, and outliers may compromise the overall reconstruction. To enhance information quality and detail, the main vessel structures are utilized, similar to the approach used in previous perspective reconstructions. Skeletonized point clouds are processed based on their Chamfer distances to achieve complete structural alignment, as outlined in problem formulation (7.2). Although the vascular structures in the model $\mathbf{M}_\mathrm{T}$, represented by $\boldsymbol{P}_\mathrm{vas}$, are exploited for orientation, they cannot be directly guided by 3D spatial coordinates and associated normals. Instead, the structures must be made coincident on the image plane by adjusting the model's geometry to match the patterns observed from the associated camera perspective. However, the surface information cannot be directly transferred to the image plane by the proposed rendering function, as the resulting point clouds would be discretized in the image matrix, making gradients infeasible across pixel locations.

The analytical camera model is used to map the structure-forming point cloud $\boldsymbol{P}_\mathsf{vas}$ to the image plane, similar to the global graph adaptation. However, analogously to the camera reconstruction (7.2) there are no predefined point alignments of the corresponding pattern. As such, the structure-forming points must be checked for visibility before proceeding with any subsequent pattern correlation. Therefore, the structure-forming point cloud $\boldsymbol{P}_\mathrm{vas}$ is sorted depending on the distances to the image plane while discarding obscured and unobservable structures. Once the visibility check is completed, which in principle follows the Z-buffering procedure (2.13), the checked points $\boldsymbol{P}^\star_\mathrm{vas}$ are projected onto the image plane according to the analytical camera model

$$\boldsymbol{p}^\star_\mathrm{vas} = \mathtt{M}(\boldsymbol{\phi}_{\mathrm{cam},i})\boldsymbol{P}^\star_\mathrm{vas}, \tag{7.3}$$

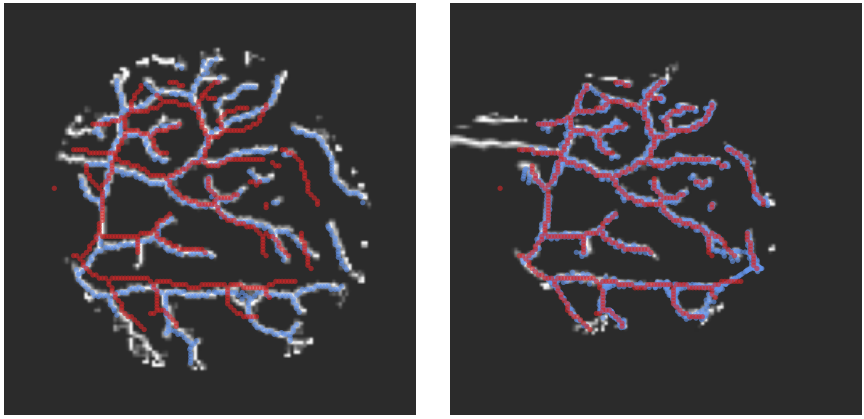
according to (2.5), to ensure differentiability. The Chamfer distance (7.1) is then used to compare the distances between the structures represented by the point clouds

$\boldsymbol{p}^{\star}_{\text{vas}}$ and $\boldsymbol{p}_{\mathcal{I}\mathbb{M}}$. This ultimately leads to the formulation of the optimization problem

$$\mathbf{V}^{\star}_{\text{G}} = \mathbf{V}^{0}_{\text{G}} + \arg\min_{\Delta\mathbf{V}} \sum_{i=t-h}^{t} \mathcal{L}_{\text{chf}}\left(\boldsymbol{p}^{\star}_{\text{vas}}, \boldsymbol{p}_{\mathcal{I}\mathbb{M}}\right) + \mathcal{L}^{\diamond}_{\text{nor}}(\mathbf{M}, \mathbf{M}^{\diamond}) + \mathcal{L}_{\text{edg}}(\mathbf{M}) + \mathcal{L}_{\text{lap}}(\mathbf{M}), \tag{7.4}$$

where $\mathbf{V}_{\text{G}}$ co-determines the vertices positions associated with $\mathbf{M}_{\text{T}}$, as discussed in Section 3.2.1.

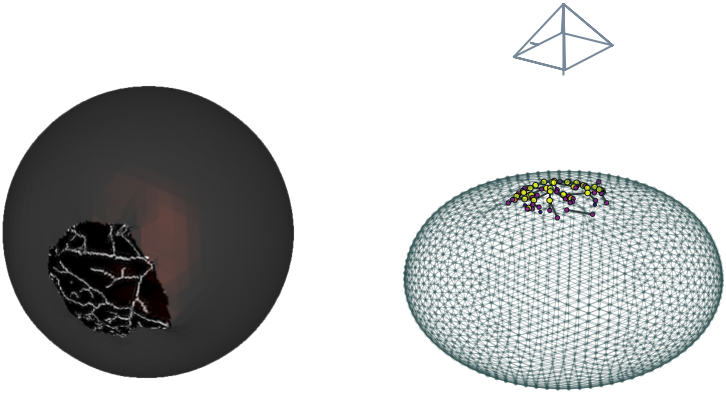
The geometry adaptation, as discussed in Section 3.1.2, is subject to regularization terms to prevent the optimization from becoming ill-posed. This robust loss design applies to all vertices that are not covered by the primary supervision of the structural adaptation. The template mesh, $\mathbf{M}^{\diamond}$, guides the regularization of areas not monitored by the primary pattern, ensuring that unsupervised regions of the model do not deviate from the previous reconstruction. Moreover, the regularization losses are scale-invariant, which allows any observed pattern scaling to be uniformly propagated across the entire mesh. For further details, refer to Section 3.1.3.4, where the inherent scaling is tested and discussed within the context of scaled silhouetted image observations.



(a) Rendering with initial model geometry. **(b)** Rendering with adjusted model geometry.

Figure 7.3: Model renderings with extracted structures based on represented texture in ● blue and extracted point cloud based on the structure of the target image in ● red. The model geometry is adjusted to align with the respective pattern depicted on the image plane.

To test the geometry adaptation, a synthetic distortion is imposed upon the structural discrepancy in accordance with the image distortion model, as depicted in Figure 7.3a. As there is no remaining residual error after pose reconstruction for the example shown in Figure (7.2b), a comparison is made between the observed



(a) Texture model $\mathbf{M}_\mathrm{T}$ with skeletonized pattern.

(b) Geometry model $\mathbf{V}_\mathrm{G}$ with graph pattern.

Figure 7.4: Model geometry resulting from the definition of texture-driven reconstruction objective in respective model representations.

target structures and the undistorted structures within the model $\mathbf{M}_\mathrm{T}$, as shown in Figure 7.3a. The figure showcases the undistorted structures ($\boldsymbol{p}^\star_\mathrm{vas}$) and the target structures subjected to the synthetic distortion ($\boldsymbol{p}_{\mathcal{I}\mathbb{M}}$). Figure 7.3b presents the $\boldsymbol{p}^\star_\mathrm{vas}$ and $\boldsymbol{p}_{\mathcal{I}\mathbb{M}}$ point cloud patterns arranged in a unified manner, demonstrating the effective adaptation of the corresponding pattern courses. The error trajectories depicted in Figure 7.5 demonstrate a fast decrease of the Chamfer distance and convergence behavior during the initial iterations.

However, it should be noted that there is a small increase in regularization losses, as the Chamfer distance enforces a non-uniform distribution of the meshes, preventing any invariant changes as outlined in the design of the regularization losses. The final geometry adaptation is illustrated in Figure 7.4a for the skeletonized mesh representation $\mathbf{M}_\mathrm{T}$, whereas the corresponding geometry model $\mathbf{M}_\mathrm{G}$ is shown in the mesh plot in Figure 7.4b.

It is worth noting that, in general, it is technically infeasible to conclusively verify whether the reconstructed geometry represents the correct geometry. The geometric triangulation principle for monocular image

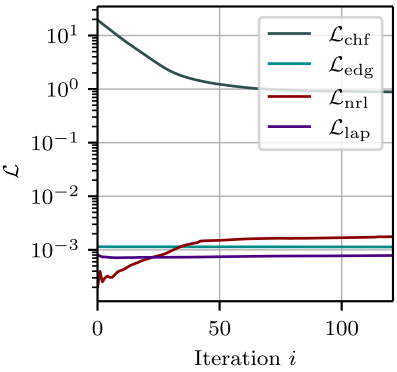


Figure 7.5: Loss trajectories of pattern-based geometry reconstruction.

data necessitates at least two images taken from two distinct perspectives. While this requirement may be met in specific formulations, it is not guaranteed in reality, as ongoing deformation between images would invalidate the reconstruction. In its most general formulation, the proposed reconstruction pipeline processes only one observation per iteration to be able to respect deformation in between multiple observations. Despite the limitations imposed by using only one image observation for geometry reconstruction, the resulting reconstruction is deemed sufficiently accurate, as it is demonstrated in Section 7.2.2 through the evaluation with additional depth measurements.

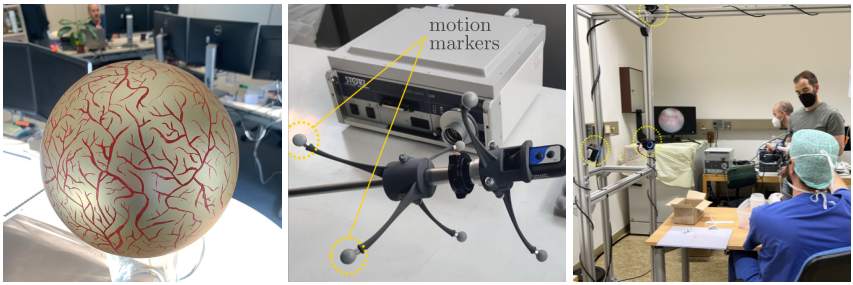
Update Global Graph Model. Once the camera position is reconstructed and the geometry is matched to the currently observed pattern, the remaining task is to update the global graph model $\mathcal{G}_\mathsf{G}$ with the patterns from the current image observation that are not included in the current representation of $\mathcal{G}_\mathsf{G}$. The graph editing policy, as presented in Section 6.3.2, is employed for the respective model update of $\mathcal{G}_\mathsf{G}$ for $\mathcal{G}_i$. The resulting editing policy $\pi^\star(\mathcal{G}_\mathsf{G}^{\circledcirc}) \rightarrow \mathcal{G}_i$ is used for the pattern update according to (6.16). The update for $\mathcal{G}_\mathsf{G}$ is given by the insertion and respective deletion operations resulting from the specified editing policy $\pi^\star$.

As new edges are inserted and existing edges are re-detected during the update process, the corresponding re-occurrence rate a_rec for each edge is updated in accordance with (6.17). The re-occurrence rate, defined for all edges in (6.14), determines which structures can be regarded as significant and reliably re-detectable, thereby enhancing the matching procedure. Consequently, only structures that have achieved at least the required re-occurrence rate $a_\mathrm{rec}^\star$ are deemed reliable. Simultaneously, patterns failing to meet this criterion are eliminated as potential matching candidates. This helps to prevent the graph from becoming overfilled, as outlined in Section 6.3.2.

Update Texture Model. The established graph editing policy $\pi^\star$ facilitates the update of the individual texture pattern $\boldsymbol{p}_\mathbf{E}$ analogously to the corresponding graph edges $\mathbf{E}$. This process involves removing prior pattern information that corresponds to the updated structure from the texture model $\mathbf{M}_\mathrm{T}$. In addition, through the edge extraction process for each edge e, the corresponding pixel set $\boldsymbol{p}_e$ is linked, which encodes the respective pattern course. This enables the reassignment of feature information, represented by $\boldsymbol{c}_e$, related to $\boldsymbol{p}_e$ to the feature space $\mathbf{C}_{\mathbf{M}_\mathrm{T}}$ according to

$$^{\circledcirc}\mathscr{R}^{\mathbf{C}}(\mathbf{M}_\mathrm{T}, \boldsymbol{p}_e, \boldsymbol{c}_e) \mapsto \mathbf{C}_{\mathbf{M}_\mathrm{T}} \ . \tag{7.5}$$

In analogy to the proposed pattern update (7.5), respective pattern information can be inserted or removed in $\mathbf{M}_\mathrm{T}$ for a given edge e if a repeated recognition or non-recognition of the individual edge information is detected for a new pattern structure that does not correspond to the model representation. Therefore, a binary representation is used to represent either the existence or non-existence of the structure.



(a) Sphere with painted structure. **(b)** Cystoscope with motion markers. **(c)** Work space of the optitrack system.

Figure 7.6: Experimental test set-up to validate pose and texture reconstruction.

Thus, edge-specific information is added or removed from the texture representation $\mathbf{C}_{\mathbf{M}_\mathrm{T}}$ if the respective edge information is detected or not detected within the required recurrence rate $a^{\star}_{\mathrm{rec}}$.

Accordingly, a given structure $\boldsymbol{p}_e$ observed in the image plane can be deleted in the model representation by the feature update by setting $\boldsymbol{c}_e \equiv 0$ for non-existence and $\boldsymbol{c}_e \equiv 1$ for existence, following (7.5). In this way, the texture model can be continuously updated without inadvertently deleting reliable structures or unintentionally adding unclear structural observations during poor image acquisition. As a result, the information value of the texture model is preserved, providing reliable pattern information for an accurate scene reconstruction while overcoming intraoperative challenges.

7.2 Experimental Validation of the Rendering-based Scene Reconstruction

The challenge of simultaneously validating pose and geometry variations on the image observation highlights their interdependent impact. To evaluate the performance, the problem is separated, and the camera reconstruction and geometry matching are independently validated in separate experiments. In Section 7.2.1, image observations are used to reconstruct the pose and texture of a rigid environment model. In Section 7.2.2, the texture and geometry adaptations are compared with depth measurements obtained from a fringe projection sensor for a given camera pose.

7.2.1 Validation of the Camera Pose Reconstruction

The objective of this experiment is to reconstruct the perspective from a captured image sequence and its corresponding pattern representations, as described in the

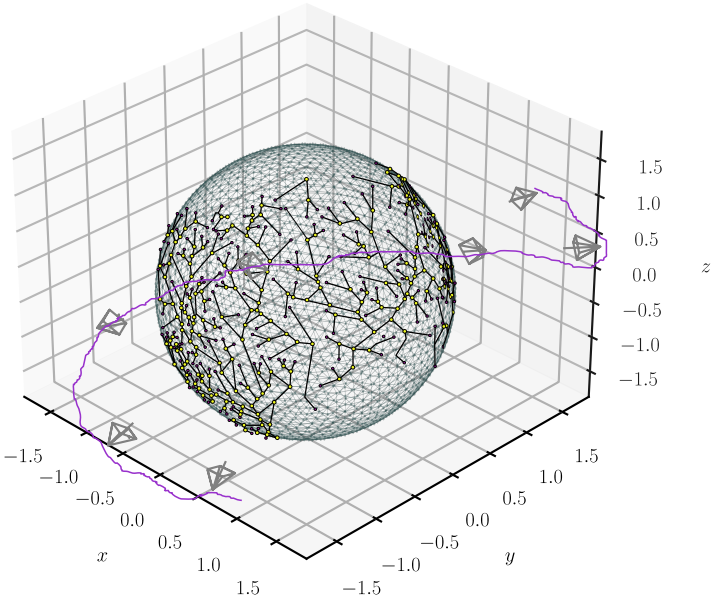


Figure 7.7: Reconstructed graph patterns in the form of global graphs $\mathcal{G}_{\mathrm{G}}$ are shown on the model surface $\mathbf{M}_{\mathrm{G}}$. The corresponding reconstructed camera position for the observation is shown in ● grey, and the measured trajectory by the Optitrak system is shown in ● purple.

proposed reconstruction pipeline. Therefore, an 18 cm diameter sphere with artificial vascular texture on its surface is considered as a test object. The sphere is shown in Figure 7.6a.

The proposed reconstruction pipeline employs sparse image data for the entire reconstruction process. Specifically, the raw video data is sampled at 2.5 s intervals. This results in a sparse dataset of 25 images that is used for the validation of the proposed pose reconstruction. In Section 6.2, Figure 6.5b illustrates a pair of subsequent images of the spherical model, used as an example in the proposed graph matching procedure for outlier removal.

To assess the accuracy of the reconstructed camera poses, a comparison is made with externally tracked pose positions obtained using an OptiTrack system. The OptiTrack system employs infrared cameras and reflective markers to precisely track objects in 3D space. As illustrated in Figure 7.6b, markers are mounted on the endoscopic camera to make the stable detection of reference points on the tracked endoscope possible. The system consists of six independent cameras that triangulate

the redundant detections in the camera recordings to determine the 3D position of a marker. The overall setup is shown in Figure 7.6c and enables a tracking accuracy for each of the mounted markers within 0.1 mm. The kinematic camera model, as described in Section 2.1.8, is utilized to compute the respective endoscope pose using the redundant marker positions. Figure 7.7 displays the measured course of an manually followed arbitrary trajectory along with the camera poses reconstructed from the observed camera images.

The experiment aimed to evaluate the accuracy of reconstructing the perspective of an image sequence using a synthetic model with a working range of $85\,\text{cm} \times 85\,\text{cm}$. The total accuracy of the absolute position is satisfactory, with an average error of 0.65 cm. However, the rotational angles show deviations in the range of $6°$, indicating a lower level of success. Nonetheless, the method validates its principle despite the relatively high level of reconstruction error.

Increasing the image resolution is presumed to improve the precision of pose reconstruction. With higher resolution, the landmark information is more accurate, leading to increased precision. It should therefore be noted that the current resolution of 256×256 pixels significantly limits the overall reconstruction precision.

7.2.2 Experimental Validation of the Geometry Reconstruction

To validate the proposed geometry reconstruction for a real-world application, the monocular image-based geometry reconstruction is compared to an externally measured depth map. A balloon is used as a deformable test object whose volume can be manipulated by inflating or deflating it with water. A water pump is implemented to control the flow rate to test the geometry reconstruction for different levels of deformation. The setup shown in Figure 7.8 captures the balloon with a tested maximum volume of 1.6 L. Additionally, the homogeneous balloon surface is textured to approximate vascular structures and to provide unique landmark information for the reconstruction targets.

To evaluate against the proposed monocular image-based geometry reconstruction, a fringe projection sensor is used to measure the deformation. The sensor setup consists of an endoscopic camera pointed at the object, with the fringe projector installed at an angle to the camera. The fringe projector illuminates the object's surface with a series of regularly spaced stripes, which the endoscopic camera records from a different viewpoint. Depth information is calculated from the captured image by applying the triangulation principle to the predefined light pattern and the observed pattern. The stripe patterns projected onto the object surface intersect with those observed by the camera, enabling the determination of corresponding depth information for each pixel in the camera plane. It is worth noting that the fringe projection sensor employed in this setup is specifically designed for non-contact depth scanning in intraoperative applications, as outlined in [133].

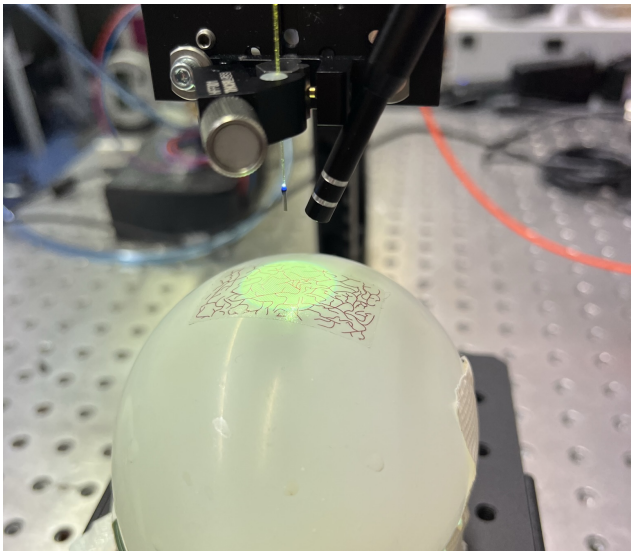
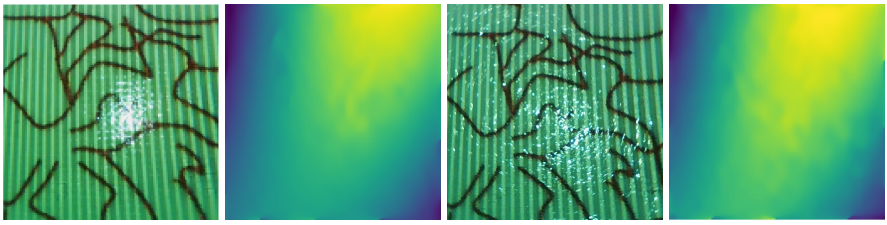


Figure 7.8: The experimental setup involves a camera, a rigid projector, and a balloon used as test object. The object's surface is illuminated by a small projector installed at an inclined angle to the camera, projecting a fringe pattern that encodes the 3D shape into the image data. [113]

A measurement observation, denoted as O, consists of a data pair $O = \{\boldsymbol{I}, \boldsymbol{D}\}$, where the color image $\boldsymbol{I}$ and the corresponding depth map $\boldsymbol{D}$ are at the same resolution and depict the same scene and perspective at the same time.

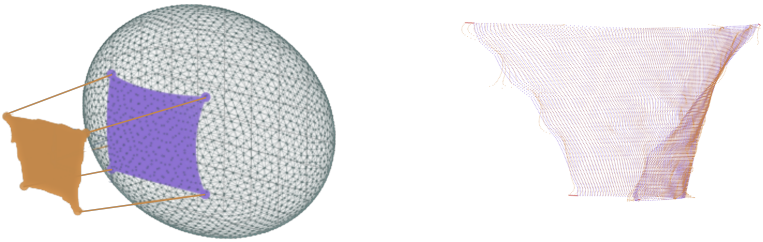
Figure 7.9a displays the observation of the balloon in its undeformed state as $O_{\mathrm{ud}} = \{\boldsymbol{I}_{\mathrm{ud}}, \boldsymbol{D}_{\mathrm{ud}}\}$. The respective image observation $\boldsymbol{I}_{\mathrm{ud}}$ is shown in Figure 7.9a, which displays the projection patterns on the surface of the balloon captured in the camera image. The corresponding reconstructed depth map $\boldsymbol{D}_{\mathrm{ud}}$ is shown in Figure 7.9b. Furthermore, the experiment involves deflating the balloon to a specific volume, resulting in a decrease in volume and a corresponding deformed observation tuple $O_{\mathrm{def}} = \{\boldsymbol{I}_{\mathrm{def}}, \boldsymbol{D}_{\mathrm{def}}\}$. The image data $\boldsymbol{I}_{\mathrm{def}}$ is displayed in Figure 7.9c, which shows the projection patterns on the surface of the deformed balloon captured in the camera image. The corresponding depth map measurement $\boldsymbol{D}_{\mathrm{def}}$ is shown in Figure 7.9d.

As the camera perspective is fixed, any observed variation in the image between pre- and post-deformation observations can be attributed solely to the object's deformation. Thus, in the proposed reconstruction procedure, it is assumed that the camera perspective remains constant, and the sphere geometry is initially aligned with the depth measurements to facilitate the comparison of the reconstructed geometry with the measured depth-map data. Therefore, the depth map is transformed into a spa-



(a) $\boldsymbol{\mathcal{I}}_{\mathrm{ud}}$ (b) $\boldsymbol{D}_{\mathrm{ud}}$ (c) $\boldsymbol{\mathcal{I}}_{\mathrm{def}}$ (d) $\boldsymbol{D}_{\mathrm{def}}$

Figure 7.9: Observation of experimental data, including balloon surface in undeformed state $\mathcal{O}_{\mathrm{ud}} = \{\boldsymbol{\mathcal{I}}_{\mathrm{ud}}, \boldsymbol{D}_{\mathrm{ud}}\}$ and in the deformed state $\mathcal{O}_{\mathrm{def}} = \{\boldsymbol{\mathcal{I}}_{\mathrm{def}}, \boldsymbol{D}_{\mathrm{def}}\}$, where each observation contains the corresponding image observation $\boldsymbol{\mathcal{I}}$ and depth map data $\boldsymbol{D}$.



(a) Depth map measurement vs. model correspondence.

(b) Point cloud differences observed before and after geometry adaptation.

Figure 7.10: A geometry model is fitted to the measured surface point cloud, where the corresponding depth information from the depth map is referred to the 3D space based on the given observation perspective.

tial representation where each pixel in the depth map and its corresponding image have a corresponding spatial point cloud representation, as respectively shown in Figure 7.10b and Figure 7.10a.

Each pixel in the image $\boldsymbol{\mathcal{I}}$ or value in the depth map $\boldsymbol{D}$ is mapped to a corresponding point cloud representation $\boldsymbol{P}$ on the mesh surface of $\mathbf{M}_{\mathrm{G}}$ by ${}^{\circ}\mathscr{R}(\mathbf{M}_{\mathrm{G}}, \boldsymbol{\phi}_{\mathrm{cam}}, \boldsymbol{\mathcal{I}}_{\mathrm{ud}}) \mapsto \boldsymbol{P}_{\mathrm{ud}}$, allowing for a direct comparison between the measured depth information $\boldsymbol{D}_{\mathrm{ud}}$ and the given surface geometry of $\mathbf{M}_{\mathrm{G}}$. This leads to the objective of adapting the mesh vertices to minimize the deviation between the mesh geometry and the corresponding 3D positions of the depth map. In Figure 7.2a, the depth map is depicted in relation to the camera's position in 3D space, along with the mesh geometry adapted by the reconstruction, as described by (7.4). Once the initial proportions of the geometry are reconstructed, the initial texture observation is applied to the initial mesh geometry following (7.5).

As outlined in Section 7.1.2.1, the geometry reconstruction adjusts the model's geometry to match the pattern of the new image observation given an initial model representation. The goal of the geometry reconstruction, as defined by the objective (7.4), is to establish a correspondence between the observed patterns of the model representation and the structures observed on the image plane.

The initialized model $\mathbf{M}_\mathrm{T}$ is rendered with the initial image observation, shown in Figure 7.12a, where extracted structures from the second observation, $\boldsymbol{O}_\mathrm{def} = \{\boldsymbol{\mathcal{I}}_\mathrm{def}, \boldsymbol{D}_\mathrm{def}\}$ are overlaid in red for comparison. The respective image $\boldsymbol{\mathcal{I}}_\mathrm{def}$ of the deflated observation is shown in Figure 7.12b. The adapted model's rendering in the adapted geometry is depicted in Figure 7.12c. The resulting corresponding pattern of the model structure and the observed target structure of the deformed observation demonstrates consistent and satisfactory results. The resulting overall geometry adaptation in 3D is depicted in Figure 7.11. The structures provide reliable monitoring of the reconstruction problem for the comparatively large geometry deformation. Remarkably, the image pattern provides reliable landmark information, even in the presence of interferences caused by observed light patterns.

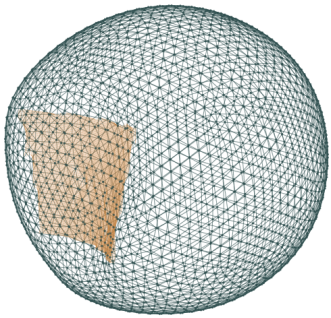


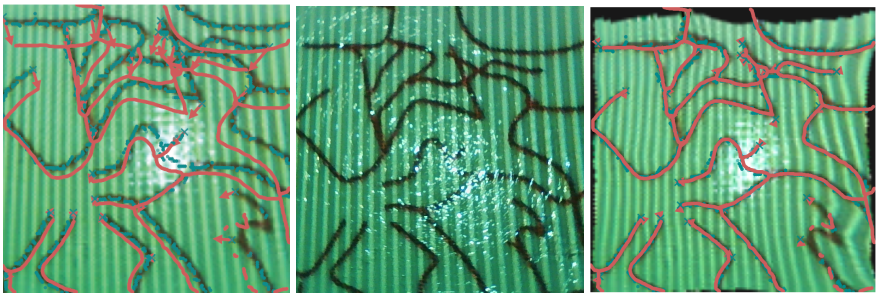
Figure 7.11: Geometry adaption of $\mathbf{M}_\mathrm{G}$ to the initial observation $\boldsymbol{D}_\mathrm{def}$.

The objective of evaluating the geometry adaption is to assess the accuracy of the mesh geometry beyond a qualitative assessment given by the coincident structure pattern observed on the image plane. Therefore, the inverse rendering function is used to sample the mesh geometry, assigning a geometry point from the mesh to each measurement point, as shown in Figure 7.10b. This allows for the calculation of the corresponding error measure. Furthermore, the differences between the point clouds of the depth map measurement and the sampled model geometry are depicted in Figure 7.10b. The precision of the reconstruction of the depth information compared to the measured depth information lies within $\pm 1.1\,\mathrm{mm}$.

The effectiveness of the scale-invariant regularization design can be evaluated by measuring the deflated fluid volume, which allows for the determination of how well the observed deformation in the image data is represented in the overall model. In the case presented, the drained fluid volume was found to be $400.28\,\mathrm{mL}$ using a weight scale corresponding to a volume change of 9%. When comparing the mesh volume before and after deformation, a change of 7% was observed. Although this yields only a rough estimate, it serves to validate the practicality of the regularization design in real-world scenarios, particularly in cystoscopic applications that have inspired its specific design.

In conclusion, the proposed monocular geometry reconstruction technique yields

satisfactory results for the tested use case, although accurately reconstructing geometry ratios from a single image observation is in general an ambiguous problem. It is important to note that the validity of the experiment is limited to the specific experimental setup, and the determined accuracy cannot be generalized to other observation perspectives. However, the proposed approach allows for the reconstruction of precise geometry ratios from multiple consecutive images in a partially rigid environment. This is accomplished by integrating additional sensor measurements, assuming that no deformation occurs between observations.



(a) unadapted 2D model observation. **(b)** deformed pattern observation **(c)** adapted 2D model observation.

Figure 7.12: The geometry of the mesh geometry model $\mathbf{M}_{\mathrm{G}}$ is adjusted to match the structure patterns of the deformed observation $\boldsymbol{\mathcal{I}}_{\mathrm{def}}$ on the image plane, from the given observation perspective, exploiting the structures contained within the texture model $\mathbf{M}_{\mathrm{T}}$

7.3 In-Plane Deformation

In the following, the flexibility of the framework is demonstrated by incorporating additional external measurement data for multi-sensory reconstruction of deformation effects. Specifically, depth information, previously used for validation in the preceding section, is combined with visual camera image data to achieve a unified geometry reconstruction. This approach ensures an unambiguous geometry reconstruction and requires data acquisition at only one observation time. Furthermore, the concept of in-plane reconstruction is introduced, which holds significant value in the context of intraoperative classification and highlights the potential of the proposed methods for real-world applications.

To provide a brief background on the problem description and in-plane reconstruction, reference is made to the main approaches discussed in Section 7.3.2. An experimental evaluation and discussion are presented in Section 7.3.3.

7.3.1 Motivation for the Field of Application

It is well-established that tumors tend to be stiffer than healthy tissue due to various factors, such as an enhanced cross-linking of the extracellular matrix. However, surgical palpation during minimally invasive surgery is constrained by limited intervention space and mechanical leverage effects, as discussed in the limitations of endoscopy in Section 1.1.2. To overcome this constraint, specialized sensors have been developed to measure tissue stiffness and offer the surgeon a more comprehensive understanding of tissue abnormalities. One such sensor concept [133] is developed at the Institute of Applied Optics [44]. This sensor principle [133] involves applying a predetermined force to the tissue and measuring the resulting deformation with a fringe projection sensor to ascertain the stiffness of the material. In this manner, a strain matrix can be derived by comparing the measured depth map distribution between successive observations. This strain matrix encodes the distribution of tissue stiffness and holds the potential to facilitate intraoperative tissue classification and diagnosis.

By comparing successive depth map measurements, the fringe projection sensor measurement facilitates the evaluation of the material's deformation in the longitudinal direction, which is in reference to the camera's principal axis. However, in-plane deformation, which refers to the change in the dimension in a plane that is perpendicular to its normal axis, has not received much attention for intraoperative scenarios. For instance, evaluating the depth map measurements of an object surface with changed geometry rations may appear in the same shape. This similarity can make it difficult to detect in-plane deformation using the majority of existing sensor principles, as illustrated in Figure 7.13.

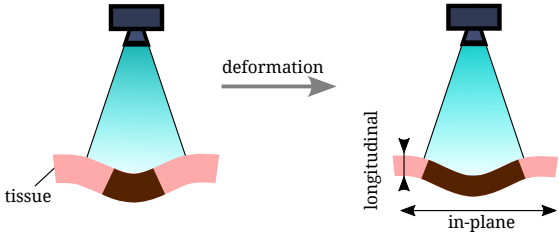


Figure 7.13: Problem description of observability of in-plane strain. The shape of the tissue can be observed equally for different deformations in-plane. In plane deformation is not observable by evaluating the depth profile. Confer with [113].

This work presents a novel method for detecting in-plane deformations by fusing depth-map sensor measurements with observed image data. The approach involves a gradient-based reconstruction formulation that utilizes depth information and visual data simultaneously to identify and quantify in-plane deformation. The depth

measurement supervises the geometry adaptation, while the visual data matches corresponding landmark features to infer in-plane deformation. By comparing reconstructed geometries in different states, the distribution of in-plane strain can be determined. This presents a promising technique that offers a new way of identifying tissue abnormalities during surgery.

7.3.2 In-Plane Reconstruction Scheme

The author has partially published the proposed approach for in-plane strain reconstruction using the methods discussed and developed in this work in [113]. The proposed concept focuses on assessing the geometric changes between two consecutive observations, specifically $O_{\mathrm{ud}} = \{\boldsymbol{I}_{\mathrm{ud}}, \boldsymbol{D}_{\mathrm{ud}}\}$ and $O_{\mathrm{def}} = \{\boldsymbol{I}_{\mathrm{def}}, \boldsymbol{D}_{\mathrm{def}}\}$, which represent the states before and after deformation, respectively. The reconstruction process consists of three primary steps, as illustrated in Figure 7.14:

1) **Initialization:** The model's shape and texture are adjusted to correspond with the measurements of the initial observation O_{ud} in the undeformed state.
2) **Adaptation to Deformed State:** The image and shape information are aligned to match the deformed observation state O_{def}. During this model adaptation, only the geometry parameters are taken into account.
3) **Decoding In-Plane Deformation:** The in-plane deformation is decoded by comparing the model's geometry before and after the adjustment and by evaluating the relative change in length for the corresponding mesh edges.

For scene reconstructions, the camera position is considered fixed in this set-up to limit the complexity. Thus, the multi-criteria reconstruction problem is formulated as

$$\mathbf{M}^{\star} = \arg\min_{\mathbf{M}} \underbrace{\|\boldsymbol{I}_{\mathbf{M}}(\mathbf{M}) - \boldsymbol{I}_{D}\|^2}_{\mathcal{L}_{\mathrm{I}}(\mathbf{M})} + \underbrace{\|\boldsymbol{P}_{\mathbf{M}}(\mathbf{M}) - \boldsymbol{P}_{D}\|^2}_{\mathcal{L}_{\mathrm{G}}(\mathbf{M})} + \mathcal{L}_{\mathrm{reg}}(\mathbf{M}) \ . \tag{7.6}$$

This formulation follows the general form of geometry adaptation in (7.4), and it incorporates the depth map data $\boldsymbol{D}$ to supervise the geometry adaptation between the model and the depth map measurement. The image similarity loss, $\mathcal{L}_{\mathrm{I}}(\mathbf{M})$, is based on the Euclidean distances between each pixel's intensities, while the geometry loss, $\mathcal{L}_{\mathrm{G}}(\mathbf{M})$, quantifies the deviation between the model representation $\mathbf{M}$ and the depth map measurements $\boldsymbol{D}$. The geometry regularization loss, $\mathcal{L}_{\mathrm{reg}}(\mathbf{M})$, is a combination of the regularization designs from (3.5), (3.6), and (3.7).

To guide the geometry adaptation, for every pixel on the image plane, intersection points on the mesh surface, denoted as $\boldsymbol{P}_{\mathrm{m}}$, are determined by utilizing the inverse rendering function represented by (4.8). This helps to depict the current shape of the mesh. By using the camera's calibration matrix and the evaluated pose θ_{cam}, the captured depth map, labeled as $\boldsymbol{D}_{\mathrm{s}}$, can be transformed into a 3D point cloud, termed as $\boldsymbol{P}_{\mathrm{s}}$. The geometry loss $\mathcal{L}_{\mathrm{G}}$, quantifies the disparity between these two point

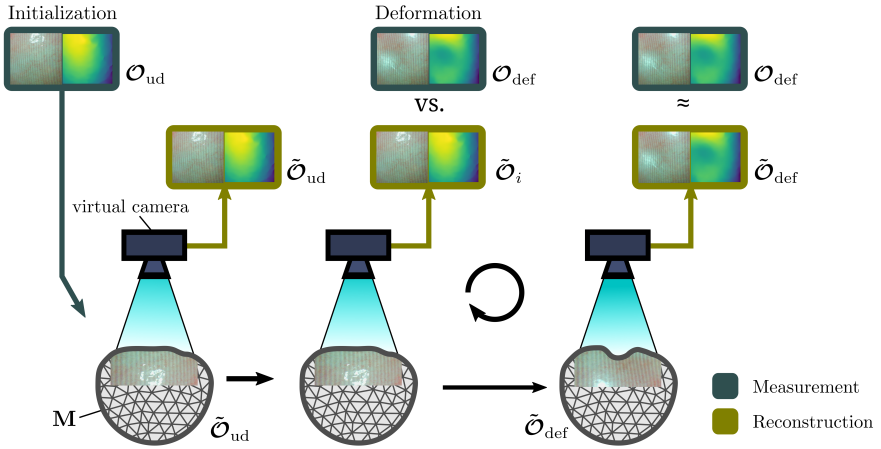


Figure 7.14: Proposed in-plane strain reconstruction concept. First, the model $\mathbf{M}$ is fitted to observation $\mathcal{O}_{\text{ud}} = \{\boldsymbol{\mathcal{I}}_{\text{ud}}, \boldsymbol{D}_{\text{ud}}\}$, encompassing both shape and texture. Post deformation, this observation $\mathcal{O}_{\text{def}} = \{\boldsymbol{\mathcal{I}}_{\text{def}}, \boldsymbol{D}_{\text{def}}\}$ is compared with the synthetic model data $\tilde{\boldsymbol{\mathcal{O}}}$. The goal is to modify the model's geometry to match the synthetic model $\tilde{\boldsymbol{\mathcal{O}}}$ after deformation. The model's reconstruction before deformation is denoted as $\tilde{\boldsymbol{\mathcal{O}}}_{\text{ud}}$, and the one after deformation is represented as $\tilde{\boldsymbol{\mathcal{O}}}_{\text{def}}$. Confer with [113].

clouds, $\boldsymbol{P}_{\text{s}}(\boldsymbol{D}_{\text{s}})$ and $\boldsymbol{P}_{\text{m}}$, and is calculated based on the spatial distance between both sets of points.

Through the loss design $\mathcal{L}_{\text{G}}(\mathbf{M})$, the geometry adaptation is supervised directly on the mesh surface. This approach contrasts with conventional methods that rely on depth map comparisons in the image plane. Further, the inverse rendered point correspondences $\boldsymbol{P}(f_{\text{ID}}, \text{uv})$ can be parameterized by respective face IDs f_{ID} and barycentric uv values. This inherently allows for the spatial point cloud update $\boldsymbol{P}^{\star}\left(\mathbf{M}_{\text{G}}^{\star}, \{f_{\text{ID,init}}, \text{uv}_{\text{init}}\}\right)$, which remains consistent with the mesh surface for any geometry adaption of $\mathbf{M}_{\text{G}}^{\star}$. The geometry invariant surface parameterization is crucial for the initialization of the model for the first observation data. It allows for the simultaneous adaptation of the model's texture and geometry in the first observation with fixed correspondences for the mesh $\mathbf{M}_{\text{G}}$ to the depth map data $\boldsymbol{D}_{\text{ud}}$. As previously discussed in Section 3.3, the simultaneous supervision of texture and geometry on the image plane reveals an ambiguous reconstruction problem. In the first iteration, the inverse rendering function (4.8) is applied to determine the respective face IDs $f_{\text{ID,init}}$ and uv_{init} values of the intersection. Then, in any subsequent iterations, the determined surface point cloud $\boldsymbol{P}$ can be updated by adapted geometry $\mathbf{M}_{\text{G}}^{\star}$. This is possible as the correspondences of the respective $f_{\text{ID,init}}$ and uv_{init} values are consistent. Therefore, the 3D point cloud information $\boldsymbol{P}(f_{\text{ID,init}}, \text{uv}_{\text{init}})$ is encoded by

deformation invariant mesh parameters $f_{\text{ID,init}}$ and $\mathbf{uv}_{\text{init}}$, making the simultaneous geometry and texture adaption well-defined.

In the second adaptation, the texture of the previously initialized model is held constant and not part of the optimization process. Instead, only the geometry parameterization of $\mathbf{M}_{\text{G}}$ is considered as the variable parameter space and optimized to simultaneously match the image and geometry data of the post-deformation observation $O_{\text{def}} = \{\boldsymbol{I}_{\text{def}}, \boldsymbol{D}_{\text{def}}\}$. During optimization, the inverse rendering mapping is applied at each iteration to recalculate the surface point cloud. This step provides the necessary flexibility to accurately represent in-plane deformation in the reconstruction of $\mathbf{M}_{\text{G}}$ for the observation O_{def}.

7.3.3 Experimental Evaluation

The experimental discussion of the reconstruction process is based on the balloon experiment, which utilizes the same setup as the validation in Section 7.2.2. The experiment simulates large volume changes by inflating and deflating the balloon with water. Additionally, the concept is applied to pig bladder tissue to demonstrate its real-world potential and the challenges it poses.

7.3.3.1 Geometry Reconstruction based on a Balloon Deformation

As a recapitulation of the test experiment introduced in Section 7.2.2, the balloon is observed in its undeformed state at 400.28 mL. The volume is subsequently reduced by 9% to a final deflated volume of 372.28 mL. The initial volume of 400.28 mL serves as the undeformed reference observation O_{ud}, while the deflated volume presents the corresponding deformed observation O_{def} of the object surface. [113]

The camera and depth map measurements used in this experiment are consistent with those considered in Section 7.2.2. Figure 7.9 shows the corresponding experimental set-up. The observation of the supervised pattern correspondences on the image plane are observed similarly as the pattern observation based on a purely image-based geometry reconstruction. As a result, the respective outcomes are essentially observed as the same, making it necessary to refer again to Figure 7.12c.

The outlined reconstruction process is applied to the undeformed observation O_{ud}, where the resulting model reconstruction $\mathbf{M}_{\text{ud}}$ is subsequently adapted to the deformed observation O_{def}, which is represented in the geometry model $\mathbf{M}_{\text{def}}$. Based on the respective model reconstructions, the in-plane strain distribution $\epsilon_{\mathbf{M}} = \frac{\mathbf{M}_{\text{def}} - \mathbf{M}_{\text{ud}}}{\mathbf{M}_{\text{ud}}}$ is calculated as the ratio of the difference between the respective model reconstructions $\mathbf{M}_{\text{ud}}$ and $\mathbf{M}_{\text{def}}$. The resulting strain distribution $\epsilon_{\mathbf{M}}$ is shown in Figure 7.15. The visualization indicates that the majority of the mesh undergoes contraction, which is consistent with the characteristics of the homogeneous latex material of the air balloon. Furthermore, the deformation extends to areas outside the observed

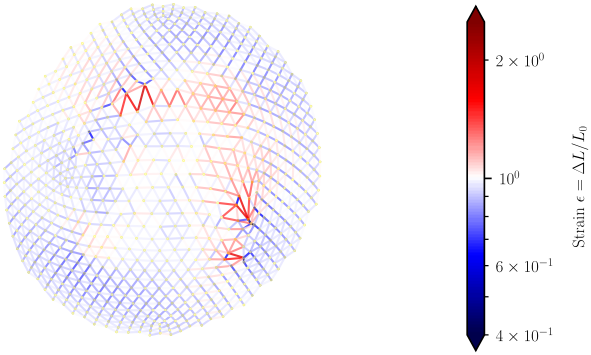


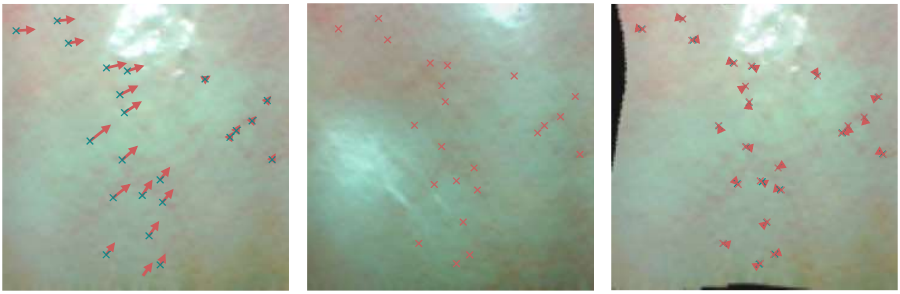
Figure 7.15: Reconstructed geometry $\mathbf{M}_{\text{ud}}$ for the balloon experiment is shown after deformation in observation O_{def}. To encode the relative in-plane strain distribution, the edges of the reconstructed geometry are color-coded: red for expansion and blue for contraction. This color-coding is based on the change in length of the edges relative to the geometry reconstruction $\mathbf{M}_{\text{ud}}$ in the undeformed stage. The in-plane strain distribution is then evaluated as $\epsilon = \frac{\mathbf{M}_{\text{def}} - \mathbf{M}_{\text{ud}}}{\mathbf{M}_{\text{ud}}}$. [113]

region due to the regularization design, which promotes uniform edge lengths. On average, the edges exhibit a 12% change in length. The change in volume can confirm the corresponding change in strain if the balloon is approximated as a hollow sphere. Therefore, the circumferential strain $\epsilon_C = \frac{\Delta C}{C_{\text{ud}}}$ is given by the relative change of the circumference $\Delta C = \|C_{\text{ud}} - C_{\text{def}}\|$ with respect to the initial circumference C_{ud}, or correspondingly, by the relative change of the radius $\Delta r = \|r_{\text{ud}} - r_{\text{def}}\|$ with respect to the initial radius r_{ud} as the relation between the circumference C and the radius r are given by $C = 2\pi r$. Moreover, the relation between the circumference C and volume V of a sphere is $C = 2\pi((\frac{3V}{4\pi})^{1/3})$. This leads to a relative change of circumference $\epsilon_C = -23\%$ for a -9% change in volume. The reconstructed in-plane distribution with an average of $\epsilon_{\text{av}} = 0.18$ is comparable in magnitude. Supplementary techniques such as landmark registration and increasing the mesh and image resolution are expected to further improve the accuracy of the method. Nevertheless, the experimental setup is considered promising and demonstrates the in-plane reconstruction concept.

7.3.3.2 Geometry Reconstruction for Deformed Tissue of a Pig Bladder Sample

The reconstruction approach is based on identifying corresponding landmark features, which can be challenging in real-world scenarios. For instance, determining and assigning intraoperative landmarks for ex-vivo bladder tissue samples, as shown in Figure 7.16a, can be difficult due to the tissue's hardly visible vascular structure and shiny surface texture. To overcome this limitation during testing, visible

landmarks are manually identified in subsequent tissue recordings. Although this approach acknowledges the limitations of manual determination of associated landmarks, it provides a solution and highlights the potential for more realistic data. The method involves using an air jet to induce deformation in a pig bladder's flexible tissue, and the object's texture can be observed in both configurations $\mathcal{O}_{\text{ud}} = \{\boldsymbol{\mathcal{I}}_{\text{ud}}, \boldsymbol{D}_{\text{ud}}\}$ and $\mathcal{O}_{\text{def}} = \{\boldsymbol{\mathcal{I}}_{\text{def}}, \boldsymbol{D}_{\text{def}}\}$, as seen in Figure 7.16. The dent created by the air-jet is located on the left part of the deformed observation in Figure 7.16b.



(a) unadapted 2D model observation.

(b) deformed pattern observation

(c) adapted 2D model observation.

Figure 7.16: Geometry reconstruction $\mathbf{M}_B$ for the balloon experiment at stage B after deformation. The edges are shaded based on the variation in length when compared to the geometry during phase A. This illustrates the relative distribution of in-plane strain, calculated as $\epsilon = \frac{\mathbf{M}B - \mathbf{M}A}{\mathbf{M}A}$. Confer with [113]

Figure 7.16c shows how the model adaptation matches the landmark information to the target localities on the image plane. In certain areas, the model surface lacked identifiable features because of bright light reflections seen in the images. Consequently, the in-plane reconstruction in these areas remains ambiguous. This study presents valuable findings on the proposed in-plane strain reconstruction methods for real-world applications, while also highlighting their current limitations and future potential. The findings emphasize the importance of reliable intraoperative landmark identification and suggest promising research directions for future work in this area. Additionally, the results of this study demonstrate the broader potential of the proposed methods for a range of other applications beyond in-plane strain reconstruction.

7.4 Summary & Conclusion

In this chapter, the various methods discussed and introduced in this thesis were integrated into a comprehensive reconstruction pipeline. This pipeline addresses the intraoperative challenge of localization and scene reconstruction in the context of

deformable environments. In its most general form, the reconstruction pipeline was formulated on the premise of requiring only a single monocular image observation. To this end, a separation approach is followed in which any changes in the image that cannot be attributed to camera position are attributed to changes in geometry. The initial step of the proposed reconstruction pipeline involves reconstructing the pose through landmark detection, utilizing graph-based methods for robust and efficient solutions. Subsequently, accuracy is enhanced by incorporating visible pattern matching, leveraging the strengths of each method to compensate for their respective limitations. The graph-based reconstruction provides a lightweight, robust solution that is independent of the initial camera location, enabling the observation of all patterns encountered so far. To account for potential inaccuracies, the Chamfer distance is used in matching all patterns at each iteration step, eliminating the need for pre-determined landmark correspondences. The method also leverages the entire pattern course to increase accuracy through gradient-based methods. Then, the model geometry is adapted to resolve discrepancies in previously measured structures that could not be resolved solely through adjustments in camera pose reconstruction. New structures observed in the image acquisition that are not yet represented in the model are incorporated into the model through the editing procedure that avoids overwriting any previously established patterns. Reliable visible patterns are strengthened through appropriate weighting, thus reinforcing the overall reliability of the solution. For validation, the proposed reconstruction framework was tested based on two separate setups to evaluate the effectiveness of camera pose reconstruction and geometry reconstruction independently. Camera pose reconstruction was validated by measuring the kinematics of a camera using an external optical system. A rigid sphere with visible structures was used as test object. The reconstructed camera pose was compared to the OptiTrack measurements. The respective pose reconstructions exhibited an average deviation of 10 mm to the measurement, which can be attributed to the resolution of the image data and the quality of the landmark information. Geometry reconstruction was validated through external measurement of a balloon's geometry using a fringe projection sensor. The results of this experiment revealed a reconstruction error of 1.8 mm in depth, which is a highly satisfactory level for the targeted intraoperative application. Expanding on the validation of geometry reconstruction, an in-plane reconstruction concept was introduced, which embeds the measured depth map data into the reconstruction formulation. In addition to its methodological significance, this approach addresses the critical issue of in-plane deformation reconstruction. The effectiveness of this approach is evaluated by comparing the mesh length between two consecutive observations of an object's surface. The results show great promise for real-world applications, particularly in determining tissue stiffness distribution and identifying hardened tissue boundaries such as tumor margins.

8 Conclusion

In this thesis, a comprehensive strategy was introduced for reconstructing camera perspective, geometry, and texture information from a single image observation in dynamic intraoperative environments. The proposed processing pipeline was motivated by the specific challenges encountered during cystoscopic interventions. The reconstruction framework is based on gradient-based optimization, aiming to adapt the model representation to the most recent camera observation. Essentially, the concepts of differential rendering and graph-based landmark extraction and mapping techniques were employed and developed to address the complexity of the problem and the robustness requirements of intraoperative challenges.

8.1 Summary and Contribution to the State-of-the-Art

The main contributions of this work to the state-of-the-art arise from the methodological contributions around the problem of intraoperative scene reconstruction for deformable environments. In contrast to prevalent reconstruction concepts in the literature, a different perspective on the problem of localization and scene reconstruction has been taken. The design of a differential rendering mapping allowed for the general objective formulation: How does the model representation have to be adjusted in terms of geometry, texture, and viewing direction of the camera model such that the rendered image approximates the actual image observation? Nevertheless, the reconstruction of the geometry, pose, and texture information presents a significant challenge as the deduced optimization problem is commonly over-determined.

A significant contribution of this work is the reformulation and separation of the ill-posed optimization problem into well-defined reconstruction problems, allowing for model reconstruction from monocular image observations. A workflow has been proposed to reduce complexity by performing pose, geometry, and texture reconstructions sequentially. Therefore, a triangle mesh has been used as the initial model representation, leveraging a subdivision strategy to allow for high texture resolution while maintaining a geometry representation at a manageable complexity. Following the proposed separation strategy for reconstruction, the model is initialized by the image patterns of the first observation. For any subsequent observation, the camera perspective is adjusted such that the rendered structure matches the structures

of the current observation. In this manner, any discrepancies that could not be eliminated by adjusting the camera perspective are then attributed to deformation effects. Furthermore, by adjusting the model geometry, the synthetically rendered image patterns of the model in the image plane can be brought to coincide with the current pattern observation. Finally, with the adapted geometry ratios, patterns detected in the current observation that were not previously represented in the model can be appropriately updated. Based on the represented pattern, any subsequent observation can be aligned to the pattern courses represented in the model representation. However, it could be shown that the alignment of the corresponding pattern based on image intensities is likely to result in a sub-optimal solution.

To meet the real-world challenges, a two-stage pose reconstruction method was proposed. Specifically, to enhance the information content, the first processing stage of the pose reconstruction method relies on predetermined landmark point matches between the model pattern and the current image pattern. This ensures a fast and reliable, albeit approximate, solution for pose reconstruction. In the second stage, the accuracy of the reconstructed pose is improved by matching the entire pattern profiles instead of relying solely on a subset of predefined point matches. To identify the necessary point matches, the visible vessel structures are extracted as graphs to obtain robust landmark information.

A graph extraction method was presented to encode unique pattern descriptions for each image observation. Graph features were then matched according to their descriptor similarities. Consequently, a structure-preserving descriptor design was introduced to describe these patterns while considering their spatial dependencies robustly. Furthermore, the graph representation was incorporated into the overall model, which allowed for updates to the spatial similarity descriptions using the most recent observations. This approach ensures that corresponding features remain updated and applicable even under changing geometric constraints.

Despite the efforts that have been undertaken to make the descriptors as unambiguous as possible, outliers in the set of similarity matches are generally unavoidable. Therefore, a new - deformation invariant - SbOR outlier classification algorithm for intraoperative scenes was presented by employing the vascular structures. The proposed outlier detection is based on the assumption that blood vessels must maintain their structural interconnection regardless of deformation. Given this principle, blood vessels cannot suddenly re-intersect and change their connectivity due to deformation. To validate the proposed structure-based outlier detection, a synthetic dataset was generated where image distortions artificially induced the deformation effect. It could be shown that the proposed outlier method reliably detects outliers with an accuracy of 94% compared to the given ground truth, regardless of the severity of the deformation. In contrast, the conventional RANSAC outlier classification fails even for moderately distorted image pairs. Nevertheless, the matches that pass the SbOR outlier detection may still contain a small number of undetected false matches. Specifically, the proportion of outliers matches but legitimate alignments

according to the SbOR check was less than 5%. However, the pose reconstruction may also be disturbed by the landmarks' inaccuracy, originating from the graph extraction. As stated, the point matches have proven to result in well-conditioned reconstruction objectives. Nevertheless, due to the inherent uncertainties, the resulting pose reconstruction is only to be considered as an initial approximation of the respective pose.

The proposed reconstruction framework enhances approximation accuracy through subsequent pose reconstruction by integrating information from the entire pattern space. In this method, pattern courses are represented as point sets, allowing for determination of similarities between the model and current observation based on distance measures between given point cloud data, eliminating the need for prior registration. Additionally, pattern differences are defined directly on the model surface, considering both 3D positions and normal orientations.

Therefore, this work presented the concept of inverse rendering, which enables the back-projection of 2D observations onto the 3D model surface in a differentiable manner. While the conventional differentiable (forward) rendering approach facilitates differentiation of pixel intensities, the proposed inverse differential rendering process transfers 2D information onto the 3D scene model in reverse. Furthermore, utilizing spatial information derived from the inverse rendering process has been demonstrated to result in more robust pose reconstruction compared to relying solely on image-based landmark information associated with the image plane.

The proposed inverse rendering concept is designed by spatially aggregating multiple auxiliary points. The distributed auxiliary points are combined based on a weighted average design, which facilitates the differentiation of the aggregated surface information. Moreover, the spatial point distribution and weight design mainly determine the differentiability. Consequently, the control parameters were evaluated based on their influence and performance in pose reconstruction using the inverse rendering design. However, a high dispersion of the auxiliary point distribution can cause noise in the sharpness of the back-projection image while also accelerating the convergence. Nevertheless, it was observed that the proposed method exhibits saturation, indicating that increasing the variance of the distribution design indefinitely would not contribute to the overall solvability of the problem. For geometry reconstruction, the model is adapted to reduce the discrepancy between corresponding image patterns. However, the geometry adaption can be challenging as unobserved occluded faces can alter arbitrarily without affecting the image-based optimization objective. To address this issue, regularization losses were introduced to ensure the stability of the reconstruction process. Particularly for intraoperative applications, a template-based mesh regularization concept was proposed to guide the orientation of unobserved regions to previously observed patterns to preserve the overall shape of unsupervised areas. Furthermore, the proposed regularization is scaling-invariant, ensuring that an observed scaling ratio in the image plane propagates throughout the entire geometry reconstruction, even for unobserved surface areas.

For pattern representation, a combination of graph representation and skeletonized pattern structure was exploited to achieve robustness and accuracy simultaneously. The graph representation provides simplicity and robustness through graph matching, while the representation of entire pattern profiles ensures the necessary resolution. However, directly transferring the established matches of currently observed patterns to the model presentation may overwrite previously seen patterns in the texture representation. Moreover, unmapped structures may be transferred in the graph representation, potentially leading to duplication in the global graph representation in the case of unrecognized matches. To address this, a processing strategy for the graphs was proposed to identify differences in the structures beyond the detected matches. Based on this strategy, the graph and skeletonized structures belonging to a graph edge can be updated in the global graph and texture map. Moreover, additional detection features have been introduced to reduce the structures to the most reliable and recognizable components for matching. Within the overall reconstruction problem, robust and well-visible patterns are reinforced, ensuring that small, irregularly observed structures are suppressed and do not negatively impact the overall reconstruction process.

8.2 Discussion and Limitations

Distinguishing between changes in a monocular image resulting from a shift in camera perspective and those resulting from deformation effects is not feasible without supplementary sensor information. However, the approach adopted in this study of adjusting the camera pose first and then resolving any remaining error by adjusting the geometry is suitable for the given application. This is because camera movements typically occur more quickly than intraoperative deformation effects. Additionally, any change in image observation caused by a change in camera perspective must adhere to the camera model, while deformation effects can cause arbitrary nonlinear distortions in the image plane. Nevertheless, the reconstruction concept can be easily extended by adding external position information and is not limited to using monocular images alone.

It is important to emphasize that a single monocular image observation does not necessarily provide a veracious depth reconstruction due to inherent physical limitations. Nevertheless, by accumulating observations of the same scene from varying viewpoints, depth information can be calculated through the geometric triangulation of feature points. If deformation occurs between successive observations, this procedure can become problematic. For an automated documentation and registration algorithm, the actual spatial precision may not be decisive; however, it is crucial to ensure that the current model observation aligns with the current observation. This issue was addressed in this work by presenting a comprehensive reconstruction pipeline. The adaptive model representation is able to accurately reflect what is

visualized to correctly re-register the intraoperative observations and measurement data in the model with the appropriate surface positions.

The reconstructed geometry was compared to external depth map measurements obtained by a fringe projection sensor for experimental validation. The experiment demonstrated the effectiveness of reconstructing geometry ratios based on a single monocular image observation. Specifically, a monocular image observation was used to reconstruct the geometry of an air balloon with an 80% decrease in volume, achieving a geometric accuracy of 94% when compared to external depth measurements. However, it is essential to note that the validity of this accuracy is limited to the specific experiment and the image resolution of 256×256 pixels utilized. Nevertheless, the experiment highlights the concept's efficacy, which is considered sufficient for a wide range of applications.

In its most general formulation, the presented reconstruction pipeline relies solely on one monocular image. This eliminates the need for any technical modifications and makes the proposed image-based scene reconstruction universally applicable. However, supplementary information such as external position or depth measurements can be easily incorporated to enhance the optimization formulation. Therefore, versatility and universality are achieved through the proposed reconstruction formulation.

Furthermore, the proposed optimization objective indirectly facilitates a sensor fusion concept, where image data and depth-map measurements were considered in a combined geometry reconstruction. Apart from its methodological aspect as a multi-objective geometry reconstruction, a concept for in-plane reconstruction was proposed. The relative change of the mesh between successive observations quantifies the in-plane strain. This quantified in-plane deformation presents a valuable and promising technique for tissue classification, with methodological foundations provided in this work.

In-plane reconstruction enables the detection of hardened tissue areas for any uniform tissue excitation. The material stiffness can be estimated from the relative tissue deformation, as the deformation is relative to its material stiffness. The potential of this method to determine material stiffness and identify hardened tissue boundaries, including tumor borders, in intraoperative settings makes further advancement of this contribution highly valuable.

Appendix

A

A.1 Numeric Optimization Following the Gradient-Decent

In general, the objective of an optimization problem is to identify the set of parameters $\boldsymbol{\theta}^*$ that yields the minimum value $\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$ of a specified loss function $\mathcal{L}(\boldsymbol{\theta})$. In some cases, it may not be practical or even possible to find the exact solution to this optimization problem using analytical methods. Therefore, numerical optimization techniques are employed to approximate the optimal solution. These techniques involve iteratively updating the parameters to minimize the loss function over multiple iterations until the convergence criteria are met.

One such technique is the gradient descent algorithm, which involves updating the parameters in the direction of the steepest descent, as determined by the gradient of the loss function. The procedure can be generalized as iteratively updating the parameters in the direction of the negative gradient of the loss function, which points towards the minimum of the function. The updating rule for this procedure is given by

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \alpha_{t-1} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{t-1}) \,, \tag{A.1}$$

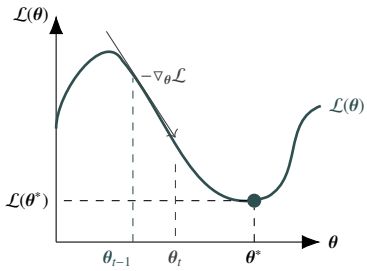


Figure A.1: Numerical minimization of the cost function $\mathcal{L}(\boldsymbol{\theta})$ by traversing the negative gradient.

where $\boldsymbol{\theta}_t$ is the current parameter set, $\boldsymbol{\theta}_{t-1}$ is the previous iteration's parameter set, α_{t-1} is the learning rate from the previous iteration, and $-\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_{t-1})$ is the negative gradient of the loss function from the previous iteration. The iterative parameter results in a minimal loss, e.g. as shown in Figure A.1.

The unmodified pure gradient descent algorithm is generally not the most effective choice for high-dimensional optimization problems due to its fixed learning rate, which can lead to numerical instability or being stuck in local minima. Therefore, two specific algorithms are presented in the following to facilitate robustness.

Stochastic gradient descent: Stochastic gradient descent (SGD) and its variants provide practical improvements over the standard gradient descent algorithm. Instead of computing gradients using the entire dataset, they randomly sample data at each iteration, resulting in faster and more efficient computations. This stochastic approach is widely used in machine learning due to its ability to handle large and complex datasets. Moreover, stochastic gradient descent enables progress in non-uniform parameter distributions, assists in escaping local minima, and prevents overfitting through regularization. Additionally, random sampling introduces noise into the optimization process, which can help avoid saddle points and other suboptimal solutions.

Adaptive Moment Estimation: The Adaptive Moment Estimation (Adam) optimizer is applied in this work to solve gradient-based optimization problems, owing to its demonstrated reliability in high-dimensional optimization problems [55]. The Adam algorithm, a variant of SGD, modifies the parameter update (A.1) to

$$\theta_t = \theta_{t-1} - \underbrace{\alpha_{t-1} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\theta_{t-1})}_{\text{orig. step size}} \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \right) \in \mathbb{R}^{\mathcal{P}}, \tag{A.2}$$

with the update operation performed for every parameter θ_p in $\boldsymbol{\theta} \in \mathbb{R}^{\mathcal{P}}$, $p = 1, \ldots, \mathcal{P}$. During the parameter update, the original step size in (A.1) is multiplied with the original step size from SGD to yield the final parameter update. The Adam solver uses first-order gradients and adapts the learning rate for each parameter based on estimates of the first and second moments of the gradients. The first moment m_t of the gradient is its mean, which is estimated using an exponential moving average with a decay rate of β_1. After updating m_t, a bias correction $m \to \hat{m}$ is applied

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\boldsymbol{\theta}} \mathcal{L}(\theta_t) \in \mathbb{R}^{p}, \qquad \beta_1 \in [0, 1] \tag{A.2a}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \ . \tag{A.2b}$$

The second moment v_t of the gradient is its uncentered variance, similarly estimated by an exponential moving average that decays at a rate of β_2. The respective estimates are calculated using exponential moving averages that decay at a rate of β_1 and β_2, respectively. An bias correction is applied by

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \nabla_{\boldsymbol{\theta}} \mathcal{L}^2(\theta_t) \in \mathbb{R}^{\mathcal{P}}, \qquad \beta_2 \in [0, 1] \tag{A.2c}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \ . \tag{A.2d}$$

to obtain the final estimates $\hat{m}_t$ and $\hat{v}_t$. The Adaptive Moment Estimation (Adam) optimizer incorporates the regularization term ε to prevent division by zero. This algorithm leverages first-order gradient information and an adaptive learning rate

for each parameter, making it effective for high-dimensional optimization problems, such as those addressed in this work.

The exponential moving averages and bias correction applied in Adam can effectively mitigate the impact of noisy gradients, which are a common issue in high-dimensional optimization problems, especially in ill-posed cases. As a result, the robustness of Adam to noisy gradients and the choice of an initial learning rate make it a strong choice for the problems covered in this work for adapting the parameterized mesh model and also for training the proposed data-driven deep learning concepts.

A.2 U-Net Network Architecture

The U-Net derives its name from the U-shaped pattern formed by its contracting and expanding convolution pathways, as illustrated in Figure A.2. This network architecture was originally developed for biomedical segmentation applications [107].

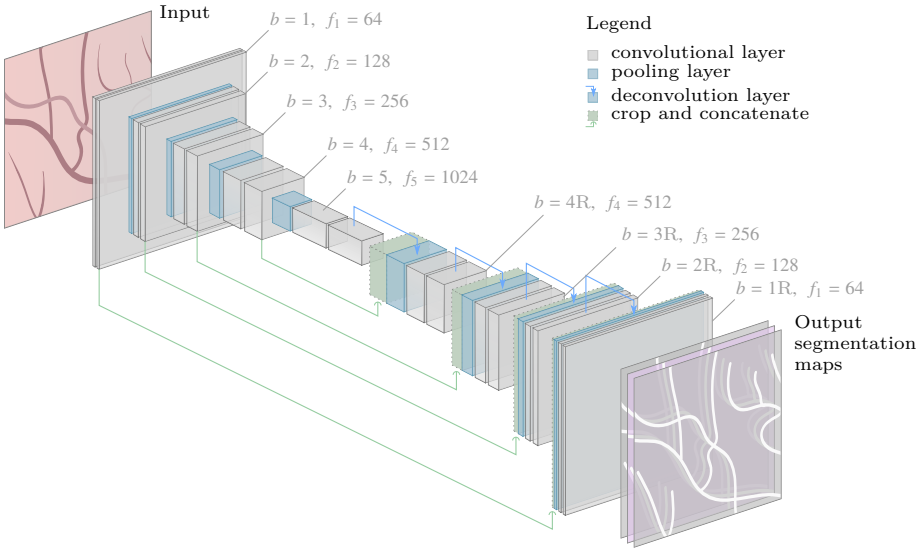


Figure A.2: U-Net architecture as described in [107]. The network is composed of a contracting path that starts at level $l = 1$ (input level), progresses to $l = 5$ (model's internal representation), and returns to level $l = 1$ (output level) via an expanding path. Skip connections (in green) connect the output feature map of the contracting path at level l to the up-sampled output at level l of the expanding path.

The U-Net processes a square input image through two consecutive convolution operations, each followed by a nonlinear activation. The input block $b = 1$ utilizes $f_1 = 64$ convolution filters. Downsampling of the intermediate output is performed by a pooling layer. The contracting path involves repeated double convolutions and downsampling until the dimensions $32 \times 32 \times 512$ are attained. Each block of consecutive double convolutions is assigned an id b, with the associated number of convolution filters being $f_b = f_1 \cdot 2^{b-1}$.

In the expanding path, two consecutive 2D convolutions are carried out, each followed by a ReLU activation. The output is then upsampled using a 2×2 deconvolution operation. This sequence of double convolutions and deconvolution is executed the same number of times as the corresponding downsampling sequence in the contracting path [107].

A crucial aspect of the U-Net architecture involves the integration of skip connections. These connections are established by combining output feature maps from the contracting pathway, prior to down-sampling, with those generated by the upsampling process in the expanding pathway. To concatenate these feature maps, they must possess identical width and height dimensions, necessitating cropping of the contracting pathway's feature map if required. The incorporation of skip connections facilitates localization of image patches throughout the entire architectural framework [107].

It is worth mentioning that in the original U-Net, as proposed in [107], output feature maps from each convolution operation are smaller than their respective input feature maps before convolution. This is because the objective in [107] focused on classifying image patches rather than conducting pixel-wise classification. Consequently, maintaining consistent spatial dimensions was not a requirement in the original U-Net. At the concluding stage, a 1x1 convolution is executed on the final output feature map for each class predicted [1]. For a more in-depth examination of the network architecture design, please refer to [106].

A.3 VGG Network Structure

While the U-Net framework is primarily employed for image segmentation tasks, the VGG network is predominantly designed for whole-image classification. This architecture ranked as one of the top contenders in the 2014 ImageNet Challenge [121]. The structure of the VGG network is illustrated in Figure A.3.

The VGG neural network takes a square RGB image as its input, as initially implemented in [121]. A series of convolutional operations with 3×3 kernel size are applied to this input image. The convolutions are padded to maintain the original input dimensions of the tensor [121]. These convolutional operations can be organized into blocks b, each sharing the same width and height dimensions. Each block consists of either two or three convolutional layers. A 2×2 max pooling layer follows

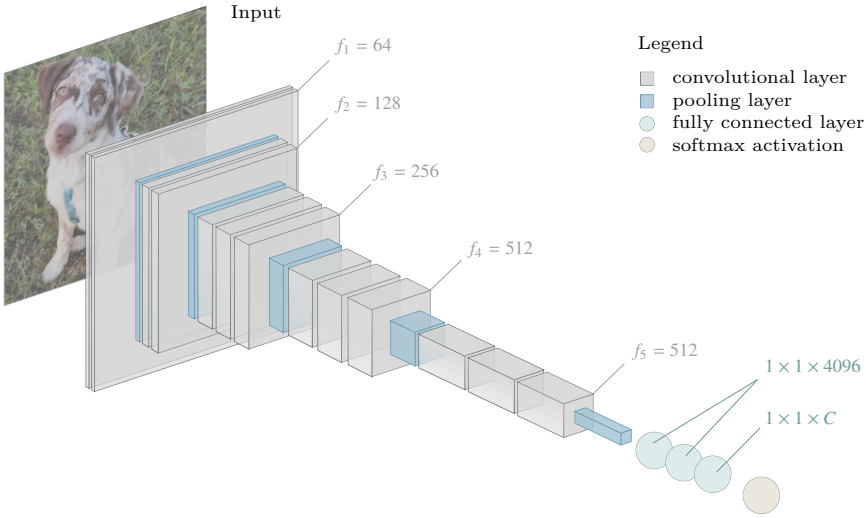


Figure A.3: The VGG-16 structure. The network is organized into blocks of convolutional layers. Each block is parameterized by f_b kernels/filters per convolution layer in block b. Dense layers and a softmax activation conclude the network.

each block. In the original VGG architecture, there are two blocks $b = \{\{\}1, , 2\}$ with two successive convolutions and three blocks $b = \{3, , 4, , 5\}$, each containing three consecutive convolutions. The VGG structure, as presented in [121], ends with several fully connected layers and a softmax layer, ultimately producing class probabilities for C classes.

A.4 Evaluation Metrics for Binary Classification

The training of a neural network involves determining a parameter set $\boldsymbol{\theta}$ that minimizes the loss function $\mathcal{L}(\boldsymbol{\theta})$ by employing gradient-based techniques. Since the neural network model is part of the loss function $\mathcal{L}(\boldsymbol{\theta})$, it influences the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$ of the loss function, subsequently affecting the parameter optimization process. This means that different model architectures may perform quite differently to one another.

After training, it is crucial to assess the performance of different model variants. Multiple metrics can be utilized to evaluate the performance of classification models, all of which compare the predicted classes $\hat{\boldsymbol{y}}$ to the actual classes $\boldsymbol{y}$ of the data in some manner.

Accuracy is a widely used metric, expressed as

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \ . \tag{A.3}$$

Intuitively, accuracy represents the proportion of correct predictions out of the total number of predictions. However, in the presence of significant class imbalance within the data, the accuracy metric might be deceptive. For instance, a model consistently predicting `TRUE` for data predominantly containing `TRUE` classes will demonstrate high accuracy. Consequently, accuracy alone does not provide a comprehensive understanding of the model's capacity to generalize on data belonging to the `FALSE` class.

For imbalanced data, alternative metrics such as precision and recall are often employed. Precision focuses on the accuracy of positive predictions, essentially asking, "Of all the positive predictions, how many were actually correct?" On the other hand, recall emphasizes the detection of actual positive cases, asking, "Of all the genuine positive cases, how many were accurately identified?" The mathematical definitions of precision and recall are as follows

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{A.4}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \ . \tag{A.5}$$

Owing to the definitions of both of these metrics, there is a trade-off between precision and recall. Raising the decision threshold[1] leads to increased precision, while recall diminishes, and vice versa. Therefore, it is crucial to determine which of these two metrics should be prioritized based on the problem being addressed.

Alternatively, the F_1 score can serve as a standalone metric. The F_1 score represents the harmonic mean of precision and recall and is defined as

$$F_1 = 2\,\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \ . \tag{A.6}$$

A.5 Iterative Loop-based Graph Extraction Strategy

The Loop Adjacency Combination (LAC) scheme is an adjacency matrix predictor is built on the Base Adjacency Combination (BAC) scheme by taking into account the degree of each node. Part of this work is published in [116]. The prediction process involves repeatedly analyzing combinations of node pairs, as shown in Figure A.4.

[1] For instance, a cutoff probability value that differentiates between the classes `TRUE` and `FALSE` in binary classification.

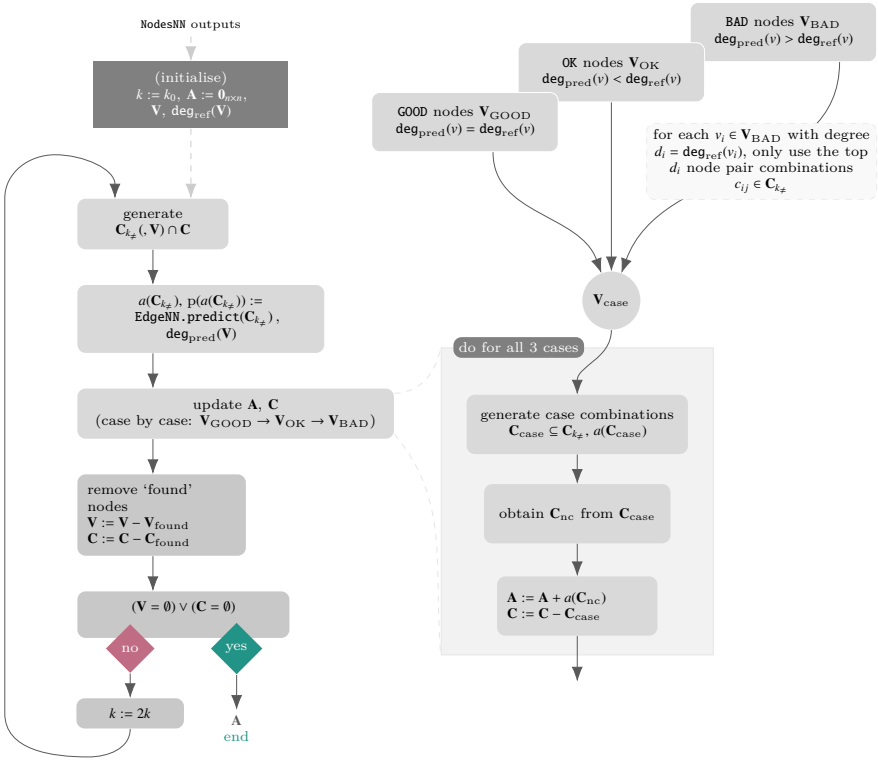


Figure A.4: The Loop Adjacency Combination (LAC) is a technique for creating a full adjacency matrix $\mathbf{A}$ by combining predictions of edges between a set of node pairs $\mathbf{C}_{k_{\neq}}$. The predictions are not immediately added to the matrix. Instead, they are first screened for potential inaccuracies by using information about the degree of each node. The diagram in Figure A.4 illustrates the process of updating the adjacency matrix, starting with nodes $\mathbf{V}_{\text{GOOD}}$, whose predicted edges match the degree values, followed by nodes $\mathbf{V}_{\text{OK}}$ with inadequate predicted edges, and finally, nodes $\mathbf{V}_{\text{BAD}}$ with an excessive number of predicted edges. [116]

Initialization For initialization, information about the degree of each node, represented as $\deg_{\text{ref}}(v_i)$, is extracted from the graph. Based on a pre-determined number of neighboring nodes, denoted as k_0, initial combinations of node pairs, represented by $\mathbf{C}_{k_{\neq}}(k_0, \mathbf{V})$, are generated based on $\mathbf{C}_k$.

Prediction After the combination of node pairs $\mathbf{C}_{k_{\neq}}$ is generated, a batch prediction is executed by using the edge extraction network EdgeNN. This prediction process

delivers a binary classification outcome for each combination, indicated as $a(\mathbf{C}_{k_{\neq}})$, and also the corresponding output probabilities/scores, represented by $\mathrm{p}(a(\mathbf{C}_{k_{\neq}}))$, that fall in the range of 0 to 1. For every node present within the node pair combinations, the total of predicted connections is represented by

$$\mathtt{deg}_{\mathrm{pred}}(v_i) = \sum_{c_{ij} \in \mathbf{C}_{k_{\neq}}} a(c_{ij}) + \sum_{x=0}^{n-1} \mathbf{A}_{ix} \ . \ , \tag{A.7}$$

which is obtained by adding up the current predictions for the node pair combinations and the entries already present in the predicted adjacency matrix, represented by $\mathbf{A}$.

Classification of Prediction Cases It's worth noting that due to the uncertain nature of edge prediction, it may happen that fewer or more edges are detected than expected. To avoid such conflicts, the initial adjacency predictions represented by $a(\mathbf{C}_{k_{\neq}})$ are not directly inserted into the placeholder adjacency matrix, represented by $\mathbf{A}$.

To prevent conflicts caused by inaccuracies in the predictions of node pair combinations, each node represented by v_i present in $\mathbf{C}_{k_{\neq}}(k, \mathbf{V})$ is assigned to one of the three groups: $\mathbf{V}_{\mathrm{GOOD}}$, $\mathbf{V}_{\mathrm{OK}}$, or $\mathbf{V}_{\mathrm{BAD}}$ as illustrated in Figure A.4. This grouping is done by evaluating the discrepancy between the reference node degrees, represented by $\mathtt{deg}_{\mathrm{ref}}(v_i)$, and the sum of the node's predicted connections, represented by $\mathtt{deg}_{\mathrm{pred}}(v)$. The groups are defined by

$$\mathbf{V}_{\mathrm{GOOD}} = \left\{ v_i \in \mathbf{V} \mid \mathtt{deg}_{\mathrm{pred}}(v_i) = \mathtt{deg}_{\mathrm{ref}}(v_i) \right\} \tag{A.8a}$$

$$\mathbf{V}_{\mathrm{OK}} = \left\{ v_i \in \mathbf{V} \mid \mathtt{deg}_{\mathrm{pred}}(v_i) < \mathtt{deg}_{\mathrm{ref}}(v_i) \right\} \tag{A.8b}$$

$$\mathbf{V}_{\mathrm{BAD}} = \left\{ v_i \in \mathbf{V} \mid \mathtt{deg}_{\mathrm{pred}}(v_i) > \mathtt{deg}_{\mathrm{ref}}(v_i) \right\} \tag{A.8c}$$

and are dealt with in a specific order: $\mathbf{V}_{\mathrm{GOOD}}$, then $\mathbf{V}_{\mathrm{OK}}$, and finally $\mathbf{V}_{\mathrm{BAD}}$. During each step, the current group is assigned to the variable $\mathbf{V}_{\mathrm{case}}$.

The processing of certain nodes, referred to as $\mathbf{V}_{\mathrm{BAD}}$, involves an additional step. For each node v_i within this set, a limited number of combinations with the highest adjacency scores $\mathrm{p}(a(c_{ij}))$ are retained. Specifically, only the top $\mathtt{deg}_{\mathrm{ref}}(v_i)$ combinations are kept, and the predicted adjacencies for the remaining combinations are set to zero. This ensures that the resulting node adjacencies satisfy the condition $\mathtt{deg}_{\mathrm{pred}}(v_i) \leq \mathtt{deg}_{\mathrm{ref}}(v_i)$ for all nodes $v_i \in \mathbf{V}$ before further processing can take place. The next step is to identify the node pair case combinations, denoted as $\mathbf{C}_{\mathrm{case}}$. These are combinations containing nodes from the current prediction case $\mathbf{V}_{\mathrm{case}}$, and are selected from the set $\mathbf{C}_{k_{\neq}}$, according to the following

$$\mathbf{C}_{\mathrm{case}} := \left\{ \left\{ v_i, v_j \right\} \in \mathbf{C}_{k_{\neq}} \,\middle|\, (v_i \in \mathbf{V}_{\mathrm{case}}) \vee (v_j \in \mathbf{V}_{\mathrm{case}}) \right\}. \tag{A.9}$$

Removal of Conflicting Combinations For each node v_i included in the set $\mathbf{C}_{\text{case}}$, the adjacencies and degrees of the node are evaluated once more. Any node, whose accumulated adjacencies $\mathtt{deg}_{\text{pred}}(v_i)$ surpasses the reference degrees $\mathtt{deg}_{\text{ref}}(v_i)$ is considered to be in conflict with other nodes. As a result, any combination that includes conflicting nodes is not used in the current iteration. This results in only the non-conflicting case combinations remaining

$$\mathbf{C}_{\text{nc}} = \left\{ c_{ij} \in \mathbf{C}_{\text{case}} \middle| \mathtt{deg}_{\text{pred}}(v) \leq \mathtt{deg}_{\text{ref}}(v) \;\forall\, v \in c_{ij} \right\} . \tag{A.10}$$

Adjacency Matrix Update With the identification of the non-conflicting case combinations, the adjacency matrix $\mathbf{A}$ can be updated to include their adjacencies, denoted as $a(\mathbf{C}_{\text{nc}})$:

$$\mathbf{A} := \mathbf{A} + a(\mathbf{C}_{\text{nc}}) \tag{A.11a}$$

To prevent any repeat predictions in the next iteration of the algorithm, these case combinations $\mathbf{C}_{\text{case}}$ are removed from the set of possible node pair combinations, denoted as $\mathbf{C}$:

$$\mathbf{C} := \mathbf{C} - \mathbf{C}_{\text{case}} . \tag{A.11b}$$

This ensures that the next iteration of the algorithm will not consider the same combinations again and thus avoid duplicated predictions.

Checking of Node Degrees Nodes that have reached their maximum degree based on the degree information from the node extraction output are recognized as

$$\mathbf{V}_{\text{found}} = \left\{ v_i \in \mathbf{V} \middle| \sum_{x=0}^{n-1} \mathbf{A}_{ix} = \mathtt{deg}_{\text{ref}}(v_i) \right\} . \tag{A.12}$$

These nodes are removed from the global set of nodes $\mathbf{V}$, and any combinations $\mathbf{C}_{\text{found}}$ that contain these nodes are also removed from the pool of potential node pair combinations $\mathbf{C}$ with

$$\mathbf{V} := \mathbf{V} - \mathbf{V}_{\text{found}} \tag{A.12a}$$

$$\mathbf{C} := \mathbf{C} - \mathbf{C}_{\text{found}} . \tag{A.12b}$$

This approach allows the algorithm to focus on the nodes that have not reached their maximum degree, and avoid including any combinations that contain those that have.

Continuation/Termination The algorithm follows a gradual process in identifying the neighboring nodes for each node. It starts by selecting a minimal number of neighbors and progressively increases it. At the end of each iteration, the number of neighbors is multiplied by two and a new search is performed on the remaining set of nodes $\mathbf{V}$ to find the closest neighbors, resulting in an updated list of combinations,

$$\mathbf{C}_{k_{\neq}} := \mathbf{C}_{k_{\neq}}(k, \mathbf{V}) \cap \mathbf{C} \ , \tag{A.13}$$

which are then utilized to repeat the steps of predicting edges, updating the matrix and resolving conflicts. The process is repeated with continually rising values of k until either every node has been given combinations that align with their degree or all possible node combinations $\mathbf{C}$ have been used up. The termination condition is given by

$$(\mathbf{V} = \emptyset) \vee (\mathbf{C} = \emptyset). \tag{A.14}$$

The final outcome of the adjacency combination scheme is the adjacency matrix $\mathbf{A}$ of the most recent update step. Once all possible node pair combinations have been exhausted, the program triggers the `STOP` flag and no further iterations are executed.

Abbreviation

ADAM	Adaptive Moment Estimation
BAC	Basic combination scheme
CNN	Convolutional Neural Network
DefSLAM	Deformable SLAM
DoG	Difference of Gaussians
EdgeNN	Edge Neural Network
FEM	Finite Element Method
FN	False negative
FNODR	False none outlier detection rate
FPGA	Field-Programmable Gate Array
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
LAC	Loop adjacency scheme
MIS	Minimally invasive surgery
MRI	Magnetic Resonance Imaging
NNRSfM	Non-Rigid Structure from Motion
ORB	Oriented FAST and Rotated BRIEF
ORB-SLAM	Oriented FAST and Rotated BRIEF SLAM
ODR	Outlier detection rate
PDD	Photodynamic diagnostics
RANSAC	Random sampling consensus
RGB	Red Green Blue
SbOR	Structure-based Outlier Removal
SfM	Structure from Motion
SfT	Shape from Template
SLAM	Simultaneous Localization and Mapping
TP	True positive
U-Net	U-Net Convolutional Neural Network
VGG16	Visual Geometry Group Network with 16 layers
V-SLAM	Visual Simultaneous Localization and Mapping

Symbol List

Symbol	Description
Image Space: Introduced in Chapter 2 and used from page 24.	
α_{skew}	Skew angle of a camera
ϑ	Rodrigues camera angle
$\boldsymbol{\phi}$	Pose of an object
$\hat{\boldsymbol{\phi}}$	Underlined pose of an object
Δp	Pixel length
c	Principal point of a camera
f	Focal length
f_{dist}	Refers to a function that adds distortion to an image to simulate the effects of a specific camera lens
f_{undist}	Refers to a function that removes distortion from an image caused by the camera lens
$\mathbf{i}$	Rodrigues unit vector
$\boldsymbol{K}$	Camera calibration matrix
k	Radial distortion parameter
$\mathtt{M}$	Camera projection matrix that describes how a 3D point is projected onto a 2D image by a camera
O_{cam}	Camera origin (position of camera)
$\boldsymbol{P}$	Point Cloud in 3D
$\boldsymbol{p}^{\mathtt{h},\mathtt{w}}$	Continuous reference position in image space for discrete pixel location $(\mathtt{h},\mathtt{w})$
$\mathcal{R}$	Retinal image plane
$\boldsymbol{R}$	Rotation matrix
$\mathbf{r}$	Rodrigues vector
s_{skew}	Skew factor of a camera
$\boldsymbol{T}$	Translation vector
t	Tangential distortion parameter
$\tilde{x}$	Real distorted x-coordinate

$\tilde{y}$	Real distorted y-coordinate
$\blacksquare_x$	specifies x-dimension of vector representation
$\blacksquare_y$	specifies y-dimension of vector representation
$\blacksquare_z$	specifies z-dimension of vector representation

Endoscope Kinematics: Introduced in Chapter 2 and used from page 34.

α_c	cystoscope tip angle
θ_n	cystoscope shaft/notch rotation
$\{W\}$	world coordinate system
O_W	origin of $\{W\}$
${}^{W}\boldsymbol{x}$	vector $\boldsymbol{x}$ given in $\{W\}$
${}^{W}_{B}\dot{\boldsymbol{x}}$	derivative of vector $\boldsymbol{x}$ with $\{B\}$ as the reference CS for the differentiation, given in $\{W\}$
$\overline{\boldsymbol{x}}$	4D quaternion representation of $\boldsymbol{x} \in \mathbb{R}^3$
$\boldsymbol{R}\{\boldsymbol{q}\}$	rotation matrix which corresponds to the quaternion $\boldsymbol{q}$
${}^{\mathrm{W}}\boldsymbol{R}_{\mathrm{B}}$	rotation matrix which rotates ${}^{B}\boldsymbol{x}$ to ${}^{W}\boldsymbol{x}$
${}^{\mathrm{W}}\boldsymbol{p}_{\mathrm{C/B}}$	position of O_C relative to O_B, given in $\{W\}$
${}^{\mathrm{W}}\boldsymbol{p}_{\mathrm{B}}$	position of O_B relative to O_W, given in $\{W\}$, ${}^{\mathrm{W}}\boldsymbol{p}_{\mathrm{B}} = {}^{W}\boldsymbol{p}_{B/W}$
${}^{\mathrm{C}}\boldsymbol{T}_{\mathrm{B}}$	4×4 homogeneous transformation matrix containing ${}^{\mathrm{C}}\boldsymbol{R}_{\mathrm{B}}$ and ${}^{\mathrm{C}}\boldsymbol{p}_{\mathrm{B}}$
$\blacksquare_\omega$	quantity related to the IMU angular velocity
$\blacksquare_a$	quantity related to the IMU acceleration
$\blacksquare_{\mathrm{m}}$	measured quantity
χ	degrees of freedom of the joints

Mesh Parameters: Introduced in Chapter 2 and used from page 38 .

$\mathbf{C}$	Vertex feature matrix
$\mathbf{F}$	Set of mesh faces
$\mathbf{F}^{\blacktriangle}$	Neighboring faces
$\mathbf{F}^{\blacktriangle}_j \in \mathcal{N}(\mathbf{F}_j)$	All neighboring faces of a specific face $\mathbf{F}_j$
$\mathbf{M}$	Mesh
$\mathcal{N}$	Function returning adjacent mesh data, as common vertices, features and faces
$\mathbf{N}$	Normal matrix associated to the surface vertices $\mathbf{V}$

$\mathbf{V}$	Vertex matrix of a mesh
$\mathbf{V}^{\blacktriangle}$	Neighboring vertices of a vertex $\mathbf{V}_j$
$\mathbf{\Delta V}$	Vertex displacement matrix
$\mathtt{u}$	First barycentric coordinate
$\mathtt{v}$	Second barycentric coordinate
$\mathtt{uv}$	Texture coordinates
$\measuredangle$	Angle between two vectors
$\blacksquare_{\text{silhou}}$	Subscript related to silhouetted image data
$\blacksquare_{\text{a}}$	Index of vertex A of a face
$\blacksquare_{\text{b}}$	Index of vertex B of a face
$\blacksquare_{\text{c}}$	Index of vertex C of a face
3D	Three-dimensional space
2D	Two-dimensional space

Image Rendering: Introduced in Chapter 2 and used from page 41.

α_{dirt}	Angle of incidence of the direct light in the Phong reflection model.
α_{spec}	Angle of reflection of the specular light in the Phong reflection model.
$\kappa_{\text{ref,spec}}$	Phong specular reflection parameter, which controls the size and sharpness of the reflective highlights in a scene.
$\boldsymbol{\phi}_{\text{cam}}$	Camera's perspective, including the position and orientation in 3D space.
$\boldsymbol{\phi}_{\text{light}}$	Direction of the light source in a reflection model.
$\mathcal{L}_{\text{euc}}$	Loss function for Euclidean distance between points
$\mathcal{L}_{\text{tex}}$	Loss function for texture consistency
$\mathbf{R}_{\text{ref}}$	Refers to the ideal reflection ray in a reflection model.
$\mathbf{R}_{\text{view}}$	Refers to the ray from the camera to a pixel in an image.
I_{int}	Intensity of the light source in a reflection model.
I	Light reflection intensity for a pixel.
κ	Material reflectance parameters that determine the appearance of an object in a scene.
$\blacksquare_{\text{amb}}$	Refers to the ambient lighting component in the Phong reflection model, which represents the indirect light in a scene.
$\blacksquare_{\text{diff}}$	Refers to the diffuse lighting component in the Phong reflection model, which represents the direct light in a scene.

$\blacksquare_{\text{spec}}$	Refers to the specular lighting component in the Phong reflection model, which represents the reflective highlights in a scene.

Differentiable Rendering: Introduced in Chapter 2 and used from page 47.

τ_{sil}	Threshold for silhouetting
γ_{diff}	Control Parameter to influence the spatial distribution of auxiliary data
ϵ_{transp}	Parameters used to calculate the aggregation weight in diffusion rendering
$\delta_j^{\text{h,w}}$	Delta sign from pixel (h, w) to face j
σ_{diff}	Sigma used in differentiable rendering for sharpness
d	Closest distance from a pixel to a face
$d_j^{\text{h,w}}$	Closest distance from pixel (h, w) to face j
$\boldsymbol{p}$	Continuous pixel positions in an image
$\boldsymbol{p}^{\text{h,w}}$	Continuous pixel location for given pixel index tuple (h, w)
$\mathcal{D}$	Distribution used for differentiable rendering
$\mathcal{D}_j^{\text{h,w}}$	Distribution of face j on pixel (h, w)
$f_{\text{ID,proj}}$	Indices of projected face indices
sigmoid	Sigmoid function used in differentiable rendering
$\mathcal{W}$	Weighting function that determines the respective data aggregation
$\boldsymbol{w}$	Weights for the aggregation of auxiliary data points
$\mathscr{R}$	Rendering function
$\mathscr{R}_{\text{sil}}$	Silhouette rendering function
$\boldsymbol{t}_j^{\text{h,w}}$	Barycentric coordinate parameterization of the closest boundary point to pixel (h, w) on face j
$\boldsymbol{U}$	Barycentric coordinates of a pixel in image space
$\blacksquare_j$	Index of a face in a mesh
$\blacksquare_{\text{sil}}$	Silhouette indices to specify silhouette-specific operations

Reconstruction Formulation: Introduced in Chapter 3 and used from page 58.

$\mathcal{L}$	General loss function to formulate the optimization objective
$\mathcal{L}_{\text{edg}}$	Loss function for mesh edge lengths

$\boldsymbol{l}_{\text{edg}}$	Length of edge used for edge loss
$\mathcal{L}_{\text{lap}}^{\diamond}$	Loss function for predefined Laplacian smoothing
$\mathcal{L}_{\text{lap}}$	Loss function for Laplacian smoothness on a sphere
$\mathcal{L}_{\text{euc}}$	Loss function for Euclidean distance between points
$\mathcal{L}_{\text{nor}}$	Loss function for normal consistency with respect to similar points
$\mathcal{L}_{\text{nor}}$	Loss function for normal consistency on a sphere
$\mathcal{L}_{\text{nor}}$	Temporary variable for normal consistency on a sphere
$\mathcal{L}_{\text{nor}}^{\diamond}$	Loss function for normal consistency between the mesh and the template mesh
$\mathcal{L}_{\text{iou}}$	Loss function for silhouetting
$\mathcal{L}_{\text{tex}}$	Loss function for texture consistency
λ	Weighting factor for the loss function
λ_{edg}	Weighting factor for mesh edge lengths
λ_{lap}	Weighting factor for Laplacian smoothness on a sphere
λ_{nor}	Weighting factor for normal consistency on a sphere
λ_{tex}	Weighting factor for texture loss

Subdivision Strategy: Introduced in Chapter 3 and used from page 67.

$\mathbf{C}_{\text{T}}$	Texture features associated with vertices of a texture mesh
$\mathbf{C}_{\text{T}}^{\text{skel}}$	Texture features associated with vertices of a texture mesh used to represent skeleton information
$\mathbf{M}_{\text{G}}$	Mesh with only geometry
$\mathbf{M}_{\text{T}}$	Mesh with texture coordinates
$\mathbf{M}_{\text{T}}^{\text{rgb}}$	Mesh with texture coordinates and color information
$\mathbf{M}_{\text{T}}$	Mesh with texture coordinates and skeleton information
$\mathbf{V}_{\text{G}}$	Vertex set of a geometry mesh
$\mathbf{V}_{\text{T}}$	Vertex set of a mesh with texture coordinates
$\mathscr{S}$	Subdivision function that refines a mesh
$\texttt{itr}_{\text{subdiv}}$	Number of iterations for mesh subdivision
$\mathbf{M}^{+}$	Mesh resulting from the application of a subdivision algorithm
$\blacksquare_{\text{G}}$	Subscript indicating a mesh with only geometry
$\blacksquare_{\text{T}}$	Subscript indicating a mesh with texture coordinates
$\blacksquare_{\text{skel}}$	Subscript indicating a mesh with skeleton information

Inverse Rendering: Introduced in Chapter 4 and used from page 77.

$\odot\delta_j^{\mathrm{h,w}}$	Sign of the delta between the pixel and a specific face
σ_{inv}	Blurring factor applied to the pixel
σ	Standard deviation of Gaussian blur applied to pixels
$\mathcal{A}$	Aggregation function used in the inverse rendering pipeline
$\boldsymbol{c}$	2D pixel position used to compute inverse-rendered feature
$\odot d_j^{\mathrm{h,w}}$	Closest distance between the pixel and a specific face
$\odot\mathcal{D}_j^{\mathrm{h,w}}$	Distribution used to calculate weights for a specific face $\mathbf{F}_j$
$\tilde{k}$	Number of depth intersections to account for in inverse rendering
n_{isec}	Number of ideal intersections per ray
$\tilde{\boldsymbol{p}}$	Distorted pixel position due to blurring
$\tilde{\mathbf{P}}$	Re-projected3D position for blurred pixel $\tilde{\boldsymbol{p}}$
$^\star\boldsymbol{P}_j^{\mathrm{h,w}}$	Intersection point on mesh in ideal 3D position
$\boldsymbol{\mathcal{P}}^{\mathrm{h,w}}$	Intersection point cloud with blurred positions
$\mathbf{R}^{\star\mathrm{h,w}}$	Ideal ray passing through a pixel in camera coordinates
$\tilde{\mathbf{R}}^{\mathrm{h,w}}$	Ray passing through a blurred pixel in camera coordinates
$\boldsymbol{X}$	Auxiliary data of inverse rendering aggregation parameters
$\boldsymbol{X}$	Result of inverse rendering aggregation
$\boldsymbol{U}$	Barycentric coordinates of a point in the mesh
$\mathsf{uv}_j^{\mathrm{h,w}}$	The barycentric coordinates of a 3D face, used for inverse diffusion rendering
$\boldsymbol{p}$	Position of a specific node in the mesh, typically the pixel position
k	Number of faces to consider during the inverse rendering process
$\odot\mathscr{R}^{\boldsymbol{P},\mathbf{N},\mathbf{C}}$	Inverse rendering function for a given point, normal, and feature
$\odot\mathscr{R}$	Inverse rendering pipeline with unspecified inputs
$\odot\boldsymbol{w}^{\mathrm{h,w}}$	Weighting of faces in the mesh based on their distance to the pixel
$\boldsymbol{z}$	Depth at which inverse rendering aggregation is performed
γ_{inv}	Weight parameter used in the aggregation step of the inverse rendering pipeline
$\blacksquare_{\leftrightarrow}$	Symbol indicating a matching relation between two graphs

$\blacksquare_{\leftrightarrows}$	Symbol indicating that two graphs are compared to determine their differences
$\blacksquare_{\odot}$	Subscript symbol indicating an inverse rendering operation

Image Segmentation: Introduced in Chapter 5 and used from page 97.

$\circledast$	Convolution operation between two functions
$\boldsymbol{\mathcal{I}}_{\mathtt{DoG}}$	Image convoluted with the Difference of Gaussian (DoG) filter
$\mathtt{DoG}$	Difference of Gaussian filter
$\sigma^2_{\mathtt{DoG}}$	Standard deviation of the Difference of Gaussian (DoG) filter
l	Length of the segmentation filter distance set
$\boldsymbol{\iota}$	Set of possible distances for segmentation filters
$[1, n_\iota]$	Range of possible values for the segmentation filter distance set
$\lvert\boldsymbol{\iota}\rvert$	Maximum number of distances in the segmentation filter distance set
$\boldsymbol{\Psi}$	Set of possible orientations for segmentation filters
$\left[1, n_\psi\right]$	Range of possible values for the segmentation filter orientation set
$\lvert\boldsymbol{\Psi}\rvert$	Maximum number of orientations in the segmentation filter orientation set
$\boldsymbol{\Omega}$	Set spatial segmentation

Image Segmentation: Introduced in Chapter 5 and used from page 100.

$\mathtt{p}$	Probability of a pixel in an image
$\mathbf{p}$	Cumulative probability of a pixel in an image
σ^2_{var}	Variance of pixel intensities in an image
τ_{otsu}	Threshold determined by Otsu's method
τ_{vasc}	Threshold for high intensity pixels for Otsu thresholding
τ_{back}	Threshold for low intensity pixels for Otsu thresholding
$\widehat{\mu}$	Mean intensity of pixels in an image

Structure Skeletonization: Introduced in Chapter 5 and used from page 102.

$\mathcal{I}_{\text{skel}}$	Image skeleton
$\tilde{\boldsymbol{p}}$	Point on the surface of the skeleton
$\boldsymbol{p}_{skel}$	Position of a pixel in the skeleton
$\delta\Omega$	Surface of the skeleton
Ω	Binarized version of the original image
$\xi\Omega$	Set of pixels belonging to the skeleton
$\lceil$	Distance function used for skeletonization

Graph Extraction: Introduced in Chapter 5 and used from page 104.

$\mathbf{A}$	adjacency matrix
n	number of nodes in a graph
$\mathbf{E}$	Set of edges in the mesh
e	edge
$\otimes$	End node
$\otimes$	Pixel whose segmentation is ambiguous
$\ominus$	Truncation operation

Data Driven Graph Extraction: Introduced in Chapter 5 and used from page 110.

$H(p)$	entropy of a distribution p
$\mathcal{I}$	image tensor
l	layer of a neural network
L	total number of layers of a neural network
ϕ	activation function
$\boldsymbol{W}_l$	weights of the l-th layer
$\boldsymbol{b}_l$	biases of the l-th layer
$\boldsymbol{\theta}$	learnable model parameters, $\boldsymbol{\theta} = \{\boldsymbol{W},\ \boldsymbol{b}\}$
$\mathcal{L}(\boldsymbol{\theta})$	loss function which depends on $\boldsymbol{\theta}$
$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$	gradient of the loss function
α	learning rate
$\boldsymbol{\lambda}$	model hyperparameters

f_b	number of convolution filters in block b
bn	use of batch normalisation
b	block number (out of all convolution blocks)
D	depth of the network; total levels $b_{\max}$
nc_2	number of double convolution blocks
nc_3	number of triple convolution blocks

Combination Scheme: Introduced in Chapter 5 and used from page 117.

acc	accuracy
k	number of neighbour nodes
F_1	harmonic mean of precision and recall

Graph Embedding: Introduced in Chapter 6 and used from page 125.

$\boldsymbol{\alpha}$	Set of inclusion angles for a node
δ_{str}	Discount factor for weighting degrees in descriptor generation
$\overline{\overline{\boldsymbol{A}}}$	Tensor of edge attributes constructed analogously to the adjacency matrix $\mathbf{A}$
$\boldsymbol{d}$	Descriptor vector
$\boldsymbol{D}_u^k$	Set of descriptor degrees for a node k steps away from the reference node
$\boldsymbol{d}_u^k$	Descriptor vector for a node k steps away from the reference node
n_{deg}	Position of a degree in the descriptor
e	An edge in the graph
$\mathbf{E}$	Set of edges in a graph
$\boldsymbol{l}$	Length of an edge in a graph
$\mathcal{X}$	Vector variable with arbitrary dimension used for the data embedding
$\mathcal{N}_4$	Function for defining the 4-neighborhood of a pixel
$\mathcal{N}_8$	Function for defining the 8-neighborhood of a pixel
n	A node in the graph
$\mathbf{V}$	Set of nodes in a graph

$\boldsymbol{p}_n$	Subset of pixels in the image that correspond to nodes in the graph
k	Number of steps away from the reference node for descriptor design
str	Descriptor structure used for encoding local information
$\blacksquare_{\text{start}}$	Starting node for an edge

Graph Matching: Introduced in Chapter 6 and used from page 130.

$\boldsymbol{d}_a$	Descriptor vector for node a
$\boldsymbol{d}_b$	Descriptor vector for node b
$\mathcal{F}_{\text{Graph}}$	Function for extracting the current graph from a set of graphs
$\mathcal{G}$	(Current) graph
$\tilde{\mathcal{G}}$	Current graph to be matched
$\mathcal{G}$	Observed graph
$\mathbf{m}$	Set of matched node pairs
$\mathbf{m}_{\text{RANSAC}}$	Set of matched node pairs after checking with RANSAC
$\mathbf{m}_{\text{SbOR}}$	Set of matched node pairs after checking for correct structure
$\mathbf{m}^\star$	Set of filtered matched node pairs
$\tilde{\mathbf{m}}$	Set of raw matched node pairs
τ	Threshold for similarity metric
$\blacksquare_u$	Index of node a in the graph
$\blacksquare_v$	Index of node b in the graph
$\blacksquare_{\texttt{sim}}$	Similarity metric used for matching
$\blacksquare_{\texttt{euc}}$	Euclidean distance
$\blacksquare^*_{\mathbf{m}}$	Set of matched node pairs after checking for correct structure and with RANSAC
$\blacksquare_{\texttt{x}}$	Subscript indicating ambiguous self-intersection
$\blacksquare^{\ominus}$	Subscript indicating a pruned object

Graph Editing: Introduced in Chapter 6 and used from page 147.

$A^\star$	A* algorithm
$a_{\text{upd,rec}}$	recurrence rate for parameters

a_{rec}	Recurrence rate for edge features
c	Modification costs
c_0	Constant scaling factor for modification cost
$\boldsymbol{d}^*$	Descriptor vector for a target object
$\tilde{\boldsymbol{d}}$	Transformed descriptor vector
$\boldsymbol{d}_{\text{rec}}$	Recurrence rate for descriptors
f_{embd}	Function for embedding features
$\tilde{\mathcal{G}}$	Graph with distortions for verification purposes
$\mathcal{G}_{\mathsf{G}}$	Global graph
$\mathcal{G}_i$	Global graph at update iteration i
$\mathcal{G}_{\mathsf{G}}^{\ominus}$	Truncated global graph
$\mathcal{G}_{\mathsf{G}}^{\ominus}$	Noisy graph
$\mathcal{G}^*$	Graph representing a target object
`Mean`	Mean value
$\mathbb{M}_{\text{matched}}$	Mask for the convex hull of current matches
n_{detect}	Number of iterations for object detection
n_{view}	Number of iterations for object viewing
n_{kd}	Number of most similar items to find using a kd-tree
π	Policy for modifying a graph
Γ	Set of modification policies
$\varnothing$	Empty set
`sim`	Similarity metric
$\blacksquare^*$	Denotes a target object

Overall Scene Reconstruction: Introduced in Chapter 7 and used from page 155.

$\boldsymbol{D}$	An image that contains the distance of each pixel from the camera.
$\boldsymbol{O}$	A variable representing an observation or measurement.
$\boldsymbol{p}^{\star}_{\text{vas}}$	3D model origin for a point cloud and a pixel
$\boldsymbol{p}_I$	3D observation origin for a point cloud and a pixel
$\boldsymbol{p}_{I\mathbb{M}}$	3D observation origin for a point cloud, a pixel and a skeleton with optional additional subscript
$\blacksquare_{\text{def}}$	Subscripts used to distinguish between variables after deformation.

$\square_{ud}$	Subscripts used to distinguish between variables before deformation.

List of Figures

List of Tables

Bibliography

[1] I. Abaspur Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal. "A Survey of State-of-the-Art on Visual SLAM". *Expert Systems with Applications* 205 (2022), p. 117734 (cited on page 10).

[2] W. Abdulla. *Mask R-CNN for Object Detection and Instance Segmentation on Keras and TensorFlow*. https://github.com/matterport/Mask_RCNN. 2017 (cited on page 62).

[3] Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, and P. Martineau. "An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems". *Proceedings of the International Conference on Pattern Recognition Applications and Methods - Volume 1*. ICPRAM 2015. Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, 2015, pp. 271–278 (cited on pages 147, 148).

[4] B. Amend, A. Kelp, M. Vaegler, M. Klünder, V. Frajs, G. Klein, K.-D. Sievert, O. Sawodny, A. Stenzl, and W. K. Aicher. "Precise Injection of Human Mesenchymal Stromal Cells in the Urethral Sphincter Complex of Göttingen Minipigs Without Unspecific Bulking Effects". *Neurourology and Urodynamics* 36 (7 2017), pp. 1723–1733 (cited on page 8).

[5] A. Arakala, S. Davis, and K. Horadam. "Retina Features based on Vessel Graph Substructures". *2011 International Joint Conference on Biometrics, IJCB 2011* (2011) (cited on pages 97, 147, 148).

[6] G. Azzopardi and N. Azzopardi. "Trainable COSFIRE Filters for Keypoint Detection and Pattern Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.2 (2013), pp. 490–503 (cited on page 97).

[7] G. Azzopardi, N. Strisciuglio, M. Vento, and N. Petkov. "Trainable COSFIRE Filters for Vessel Delineation with Application to Retinal Images". *Medical Image Analysis* 19.1 (2015), pp. 46–57 (cited on pages 92, 97).

[8] O. Bănică and C. L. Bajaj. "A Comparison of Subdivision Schemes for Triangle Meshes". *The Visual Computer* 19.6-8 (2003), pp. 361–376 (cited on pages 67, 69).

[9] E. Brauchle, J. Kasper, R. Daum, N. Schierbaum, C. Falch, A. Kirschniak, T. E. Schäffer, and K. Schenke-Layland. "Biomechanical and Biomolecular Characterization of Extracellular Matrix Structures in Human Colon Carcinomas." eng. *Matrix Biology : Journal of the International Society for Matrix Biology* 68-69 (2018), pp. 180–193 (cited on page 8).

[10] D. Braun, S. Yang, J. N. Martel, C. N. Riviere, and B. C. Becker. "EyeSLAM: Real-time Simultaneous Localization and Mapping of Retinal Vessels during Intraocular Microsurgery". *The International Journal of Medical Robotics and Computer Assisted Surgery* 14.1 (2018), e1848 (cited on pages 10, 13).

[11] A. M. Bronstein, M. M. Bronstein, and R. Stoop. "Thinning Algorithms for Shape Description". *International Journal of Computer Vision* 74.1 (2008), pp. 75–91 (cited on page 102).

[12] D. T. Butcher, T. Alliston, and V. M. Weaver. "A Tense Situation: Forcing Tumour Progression". eng. *Nature reviews. Cancer* 9 (2 2009), pp. 108–22 (cited on page 8).

[13] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. "BRIEF: Binary Robust Independent Elementary Features". *Computer Vision ECCV 2010*. Edited by K. Daniilidis, P. Maragos, and N. Paragios. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792 (cited on page 11).

[14] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM" (2020) (cited on pages 11, 12).

[15] A. Chakravarty and J. Sivaswamy. "A Supervised Joint Multi-Layer Segmentation Framework for Retinal Optical Coherence Tomography Images Using Conditional Random Field." eng. *Computer Methods and Programs in Biomedicine* 165 (2018), pp. 235–250 (cited on page 96).

[16] R. Chen, T. Bobrow, T. Athey, Mahmood, and N. Durr. "SLAM Endoscopy Enhanced by Adversarial Depth Prediction". 2019 (cited on page 13).

[17] D. Claus, P. M. Schumacher, T. Labitzke, M. Mlikota, U. Weber, et al. "Intraoperative Model Based identification of Tissue Properties Using a Multimodal and Multiscale Elastographic Measurement Approach". *Novel Biophotonics Techniques and Applications 3*. Optica Publishing Group, 2015, p. 95400M (cited on page 8).

[18] T. Collins, B. Compte, and A. Bartoli. "Deformable Shape-From-Motion in Laparoscopy Using a Rigid Sliding Window". *Proceedings of Medical Image Understandings and Analysis (MIUA 11)*. 2011 (cited on page 11).

[19] R. L. Cook and K. E. Torrance. "A Reflectance Model for Computer Graphics". *ACM Transaction on Graphics* 1.1 (1982), pp. 7–24 (cited on page 45).

[20] P. Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer, 2011 (cited on page 30).

[21] J. S. Dai. "Euler–Rodrigues Formula Variations, Quaternion Conjugation and Intrinsic Connections". *Mechanism and Machine Theory* 92 (2015), pp. 144–152 (cited on page 30).

[22] J. Denavit and R. S. Hartenberg. “A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices”. *Journal of Applied Mechanics, Transactions ASME* 22.2 (1965), pp. 215–221 (cited on pages 34, 35).

[23] K. Deng, J. Tian, J. Zheng, X. Zhang, X. Dai, and M. Xu. “Retinal Fundus Image Registration via Vascular Structure Graph Matching”. *International Journal of Biomedical Imaging* 2010 (2010), pp. 1–13 (cited on page 12).

[24] C. Doignon, F. Nageotte, and M. de Mathelin. “Segmentation and Guidance of Multiple Rigid Objects for Intra-operative Endoscopic Vision”. *WDV*. 2006 (cited on page 62).

[25] J. Dolz, X. Xu, J. Rony, J. Yuan, Y. Liu, E. Granger, C. Desrosiers, X. Zhang, I. Ben Ayed, and H. Lu. “Multiregion Segmentation of Bladder Cancer Structures in MRI with Progressive Dilated Convolutional Networks.” *Medical physics* 45 (12 2018), pp. 5482–5493 (cited on page 62).

[26] D. Doo and M. Sabin. “Noise-Free Quadrilateral Mesh Generation”. *Computer-Aided Design* 10.3 (1978), pp. 199–205 (cited on page 67).

[27] D. Eigen, C. Puhrsch, and R. Fergus. “Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network”. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’14. Cambridge, MA, USA: MIT Press, 2014, pp. 2366–2374 (cited on page 13).

[28] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, et al. “SOFA: A Multi-Model Framework for Interactive Physical Simulation”. *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*. Edited by Y. Payan. Volume 11. Studies in Mechanobiology, Tissue Engineering and Biomaterials. Springer, 2012, pp. 283–321 (cited on page 11).

[29] M. A. Fischler and R. C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. *Communications of the Association for Computing Machinery* 24.6 (1981), pp. 381–395 (cited on pages 32, 133).

[30] H. Fritzsche, A. Boese, M. Schostak, and M. Friebe. “Resectoscope With an Easy-to-Use Twist Mechanism for Improved Handling”. *Current Directions in Biomedical Engineering* 2.1 (2016), pp. 379–382 (cited on page 11).

[31] Z. Fu, Z. Jin, C. Zhang, Z. He, Z. Zha, C. Hu, T. Gan, Q. Yan, P. Wang, and X. Ye. “The Future of Endoscopic Navigation: A Review of Advanced Endoscopic Vision Technology”. *IEEE Access* 9 (2021), pp. 41144–41167 (cited on page 10).

[32] A. S. Glassner, ed. *An Introduction to Ray Tracing*. GBR: Academic Press Ltd., 1989 (cited on page 44).

[33] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. USA: Prentice-Hall, Inc., 2006 (cited on pages 102, 104).

[34] P. F. U. Gotardo and A. M. Martinez. "Kernel Non-Rigid Structure from Motion". *2011 International Conference on Computer Vision*. 2011, pp. 802–809 (cited on page 12).

[35] O. G. Grasa, J. Civera, and J. Montiel. "EKF Monocular SLAM with Relocalization for Laparoscopic Sequences". *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 4816–4821 (cited on page 12).

[36] O. G. Grasa, E. Bernal, S. Casado, I. Gil, and J. M. M. Montiel. "Visual SLAM for Handheld Monocular Endoscope." eng. *IEEE Transactions on Medical Imaging* 33 (1 2014), pp. 135–46 (cited on page 10).

[37] P. E. Hart, N. J. Nilsson, and B. Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107 (cited on page 150).

[38] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003 (cited on page 30).

[39] M. Heimann, H. Shen, T. Safavi, and D. Koutra. "REGAL". *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (2018) (cited on page 125).

[40] M. Hu, G. Penney, D. Rueckert, P. Edwards, F. Bello, R. Casula, M. Figl, and D. Hawkes. "Non-Rigid Reconstruction of the Beating Heart Surface for Minimally Invasive Cardiac Surgery". *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*. 2009, pp. 34–42 (cited on page 12).

[41] M. Hu, S. Thompson, S. Johnsen, K. Gurusamy, B. Davidson, and D. Hawkes. "3D Reconstruction of Internal Organ Surfaces for Minimally Invasive Laparoscopic Surgery". *Proceedings of MICCAI 2007*. Springer Berlin Heidelberg, 2007, pp. 68–77 (cited on page 13).

[42] D. Q. Huynh. "Metrics for 3D Rotations: Comparison and Analysis". *Journal of Mathematical Imaging and Vision* 35.2 (2009), pp. 155–164 (cited on page 30).

[43] K. Incetan, I. Celik, A. Obeid, I. Gokceler, K. Ozyoruk, et al. *VR-Caps: A Virtual Environment for Capsule Endoscopy*. 2020 (cited on page 10).

[44] *Institute of Applied Optics (ITO)*. https://www.ito.uni-stuttgart.de/en/. Accessed: 2023-01-19 (cited on page 172).

[45] F. Isensee, P. Kickingereder, W. Wick, and M. Bendszus. "Automated Brain Tumor Segmentation Using Deep Neural Networks". *International MICCAI Brainlesion Workshop*. Springer. 2018, pp. 3–11 (cited on pages 62, 94).

[46] S. H. N. Jensen, M. E. B. Doest, H. Aanæs, and A. Del Bue. "A Benchmark and Evaluation of Non-Rigid Structure from Motion". *International Journal of Computer Vision* 129.4 (2021), pp. 882–899 (cited on page 12).

[47] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su. "DUNet: A Deformable Network for Retinal Vessel Segmentation". *Knowledge-Based Systems* 178 (2019), pp. 149–162 (cited on page 96).

[48] R. Juarez-Salazar, J. Zheng, and V. H. Diaz-Ramirez. "Distorted Pinhole Camera Modeling and Calibration". *Appl. Opt.* 59.36 (2020), pp. 11310–11318 (cited on page 27).

[49] A. M. Kamat and P. C. Black. *Bladder Cancer*. Cham: Springer International Publishing, 2021 (cited on page 4).

[50] H. Kato, D. Beker, M. Morariu, T. Ando, T. Matsuoka, W. Kehl, and A. Gaidon. "Differentiable Rendering: A Survey". *Computing Research Repository* abs/2006.12057 (2020) (cited on pages 37, 47).

[51] A. Kaufman and J. Wang. "3D Surface Reconstruction from Endoscopic Videos". *Visualization in Medicine and Life Sciences*. Springer Berlin Heidelberg, 2008, pp. 61–74 (cited on page 13).

[52] A. Kelp, A. Albrecht, B. Amend, M. Klünder, P. Rapp, O. Sawodny, A. Stenzl, and W. K. Aicher. "Establishing and monitoring of Urethral Sphincter Deficiency in a Large Animal Model." eng. *World Journal of Urology* 35 (12 2017), pp. 1977–1986 (cited on page 8).

[53] A. R. Khairuddin, M. S. Talib, and H. Haron. "Review on simultaneous localization and mapping (SLAM)". *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. 2015, pp. 85–90 (cited on page 10).

[54] M. Khakzar and H. Pourghassem. "A Retinal Image Authentication Framework based on a Graph-based Representation Algorithm in a Two-Stage Matching Structure". *Biocybernetics and Biomedical Engineering* 37.4 (2017), pp. 742–759 (cited on pages 96, 103–105).

[55] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". *3rd International Conference on Learning Representations (ICLR)*. Edited by Y. Bengio and Y. LeCun. 2015 (cited on page 185).

[56] T. Kohlberger, V. Singh, C. Alvino, C. Bahlmann, and L. Grady. "Evaluating Segmentation Error without Ground Truth". *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*. Edited by N. Ayache, H. Delingette, P. Golland, and K. Mori. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 528–536 (cited on page 62).

[57] M. Kok, J. D. Hol, and T. B. Schön. "Using Inertial Sensors for Position and Orientation Estimation". *Foundations and Trends® in Signal Processing* 11.1-2 (2017), pp. 1–153 (cited on page 30).

[58] R. [Kramme, ed. *Medizintechnik: Verfahren - Systeme - Informationsverarbeitung ; mit 161 Tabellen*. Deutsch. Mit 926 Abb., davon 123 in Farbe, und 161 Tab. Berlin ; Heidelberg: Springer, 2011, XXIII, 1071 Seiten (cited on pages 1, 2).

[59] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. "Deeper Depth Prediction with Fully Convolutional Residual Networks". *2016 Fourth International Conference on 3D Vision (3DV)*. 2016, pp. 239–248 (cited on page 13).

[60] S. M. Lajevardi, A. Arakala, S. A. Davis, and K. J. Horadam. "Retina Verification System Based on Biometric Graph Matching". *IEEE Transactions on Image Processing* 22.9 (2013), pp. 3625–3635 (cited on pages 126, 147, 148).

[61] J. Lamarca, S. Parashar, A. Bartoli, and J. M. M. Montiel. "DefSLAM: Tracking and Mapping of Deforming Scenes From Monocular Sequences". *IEEE Transactions on Robotics* 37.1 (2021), pp. 291–303 (cited on pages 11, 12).

[62] P. Lamata, T. Morvan, M. Reimers, E. Samset, and J. Declerck. "Addressing Shading-based Laparoscopic Registration". *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany*. Springer Berlin Heidelberg, 2009, pp. 189–192 (cited on page 13).

[63] C. Y. Lee and C. Y. Kuo. "Thinning Methodologies-A Comprehensive Survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.9 (1994), pp. 868–885 (cited on page 102).

[64] S. Lee, M. Lerotic, V. Vitiello, S. Giannarou, K. Kwok, M. Visentini-Scarzanella, and G. Yang. "From Medical Images to Minimally Invasive Intervention: Computer Assistance for Robotic Surgery". *Computerized Medical Imaging and Graphics* 34.1 (2010), pp. 33–45 (cited on page 13).

[65] S. Liu, W. Chen, T. Li, and H. Li. "Soft Rasterizer: A Differentiable Renderer for Image-Based 3D Reasoning". *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019 (cited on pages 16, 48–50, 53, 55).

[66] S. Liu, T. Li, W. Chen, and H. Li. "A General Differentiable Mesh Renderer for Image-Based 3D Reasoning". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.1 (2022), pp. 50–62 (cited on page 54).

[67] W. Liu, H. Wang, Q. Wu, X. Liu, and H. Zhang. "Pixel Pattern Analysis and Removal in Skeletonized Images". *IEEE Transactions on Information Forensics and Security* 10.5 (2015), pp. 1091–1104 (cited on pages 102, 104).

[68] C. Loop. *Smooth Subdivision Surfaces based on Triangles*. Tech. rep. Department of Computer Science, University of Utah, 1987 (cited on pages 67, 69).

[69] M. M. Loper and M. J. Black. "OpenDR: An Approximate Differentiable Renderer". *Computer Vision – ECCV 2014*. Edited by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Cham: Springer International Publishing, 2014, pp. 154–169 (cited on page 49).

[70] L. Ma, Y. Chen, and K. L. Moore. "A Family of Simplified Geometric Distortion Models for Camera Calibration". *CoRR* cs.CV/0308003 (2003) (cited on page 28).

[71] N. Mahmoud, I. Cirauqui, A. Hostettler, C. Doignon, L. Soler, J. Marescaux, and J. Montiel. "ORBSLAM-based Endoscope Tracking and 3D Reconstruction". *International Workshop on Computer-Assisted and Robotic Endoscopy*. Springer. 2016, pp. 72–83 (cited on page 11).

[72] L. Maier-Hein, P. Mountney, A. Bartoli, H. Elhawary, D. Elson, et al. "Optical Techniques for 3D Surface Reconstruction in Computer-Assisted Laparoscopic Surgery". *Medical Image Analysis* (2013) (cited on page 14).

[73] A. Malti, A. Bartoli, and T. Collins. "Template-based Conformal Shape-from-Motion from Registered Laparoscopic Images." *MIUA*. Volume 1. 2. 2011, p. 6 (cited on page 12).

[74] M. F. Maritz. "Rotations in Three Dimensions". *SIAM Review* 63.2 (2021), pp. 395–404 (cited on page 30).

[75] F. L. Markley and J. L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. Springer New York, 2014 (cited on page 30).

[76] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation". *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4040–4048 (cited on page 13).

[77] D. Mirota, M. Ishii, and G. Hager. "Vision-based Navigation in Image-Guided Interventions". *Annual Review of Biomedical Engineering* 13 (2011), pp. 297–319 (cited on page 14).

[78] D. Mirota, H. Wang, R. Taylor, M. Ishii, and G. Hager. "Toward Video-based Navigation for Endoscopic Endonasal Skull Base Surgery". *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*. 2009, pp. 91–99 (cited on page 11).

[79] F. Mirzaei and S. Roumeliotis. "A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation". *IEEE Transactions on Robotics* 24.5 (2008), pp. 1143–1156 (cited on page 32).

[80] P. Mountney, D. Stoyanov, and G. Yang. "Recovering Tissue Deformation and Laparoscope Motion for Minimally Invasive Surgery" (2011) (cited on pages 11, 13).

[81] R. Mukundan. *3D Mesh Processing and Character Animation: With Examples Using OpenGL, OpenMesh and Assimp*. Springer International Publishing, 2022 (cited on pages 58, 67).

[82] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163 (cited on page 11).

[83] R. Mur-Artal and J. D. Tardos. "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262 (cited on page 11).

[84] T. Nagelhus Hernes, F. Lindseth, T. Selbekk, A. Wollf, O. Solberg, E. Harg, O. Rygh, G. Tangen, I. Rasmussen, S. Augdal, et al. "Computer-Assisted 3D Ultrasound-Guided Neurosurgery: Technological Contributions, Including Multimodal Registration and Advanced Display, Demonstrating Future Perspectives". *The International Journal of Medical Robotics and Computer Assisted Surgery* 2.1 (2006), pp. 45–59 (cited on page 12).

[85] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. "Laplacian Mesh Optimization". *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*. GRAPHITE '06. New York, NY, USA: Association for Computing Machinery, 2006, pp. 381–389 (cited on page 58).

[86] S. Nikolov, S. Blackwell, A. Zverovitch, R. Mendes, M. Livne, et al. "Clinically Applicable Segmentation of Head and Neck Anatomy for Radiotherapy: Deep Learning Algorithm Development and Validation Study." eng. *Journal of Medical Internet Research* 23 (7 2021), e26151 (cited on page 62).

[87] N. Otsu. "A Threshold Selection Method from Gray-Level Histograms". *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66 (cited on page 101).

[88] A. Özgün Çiçek, Ö. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation". *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2016, pp. 424–432 (cited on page 94).

[89] K. B. Ozyoruk, G. I. Gokceler, G. Coskun, K. Incetan, Y. Almalioglu, et al. "EndoSLAM Dataset and An Unsupervised Monocular Visual Odometry and Depth Estimation Approach for Endoscopic Videos: Endo-SfMLearner" (2020) (cited on pages 10, 11).

[90] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". *Computing Research Repository* (2019) (cited on page 160).

[91] J. Peddie. *Ray Tracing: A Tool for All*. Cham: Springer, 2019 (cited on page 43).

[92] D. L. Pham, C. Xu, and J. L. Prince. "Current Methods in Medical Image Segmentation". *Annual Review of Biomedical Engineering* 2.1 (2000), pp. 315–337 (cited on page 96).

[93] B. T. Phong. “Illumination for Computer Generated Pictures”. *Commun. ACM* 18.6 (1975), pp. 311–317 (cited on pages 45, 47).

[94] M. Pollefeys. “Visual 3D Modeling from Images.” 2004, p. 3 (cited on pages 29, 31, 32).

[95] J. M. Prendergast, G. A. Formosa, C. R. Heckman, and M. E. Rentschler. “Autonomous Localization, Navigation and Haustral Fold Detection for Robotic Endoscopy”. *2018 IEEE RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 783–790 (cited on page 11).

[96] *PyOpenGL*. https://pypi.org/project/PyOpenGL. 2020 (cited on page 41).

[97] L. Qiu and H. Ren. “Endoscope Navigation with SLAM-based Registration to Computed Tomography for Transoral Surgery”. *International Journal of Intelligent Robotics and Applications* 4.2 (2020), pp. 252–263 (cited on pages 10, 11).

[98] P. Ram and K. Sinha. “Revisiting Kd-Tree for Nearest Neighbor Search”. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 1378–1388 (cited on page 131).

[99] P. Rapp, O. Sawodny, C. Tarín, C. R. Pech, J. Mischinger, and C. Schwentner. “A concept for a novel surgical navigation system”. *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2014, pp. 3884–3889 (cited on page 8).

[100] A. Rau, P. J. E. Edwards, O. F. Ahmad, P. Riordan, M. Janatka, L. B. Lovat, and D. Stoyanov. “Implicit Domain Adaptation with Conditional Generative Adversarial Networks for Depth Prediction in Endoscopy”. *International Journal of Computer Assisted Radiology and Surgery* 14.7 (2019), pp. 1167–1176 (cited on page 14).

[101] N. Ravi, J. Reizenstein, D. Novotný, T. Gordon, W. Lo, J. Johnson, and G. Gkioxari. “Accelerating 3D Deep Learning with PyTorch3D”. *Computing Research Repository* abs/2007.08501 (2020) (cited on pages 16, 48, 49, 55, 159).

[102] *Real-time Endoscopic Image Stitching for Cystoscopy*. Koblenz-Landau, Universiät: Cuviller Verlag, Göttingen, 2017 (cited on page 7).

[103] *Realistic and accurate 3D model of Full Male Anatomy*. https://www.turbosquid.com/3d-models/male-anatomy-organs-model-1373822. Accessed: 2023-04-12 (cited on page 63).

[104] K. Riesen and H. Bunke. “Approximate Graph Edit Distance Computation by Means of Bipartite Graph Matching”. *Image and Vision Computing* 27.7 (2009). 7th IAPR-TC15 Workshop on Graph-based Representations (GbR 2007), pp. 950–959 (cited on pages 126, 147, 148, 151).

[105] S. Röhl. “Intraoperative Modellierung und Registrierung für ein laparoskopisches Assistenzsystem”. PhD thesis. 2013. 258 pp. (cited on pages 10, 26, 33).

[106] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". *Computing Research Repository* (2015) (cited on pages 94, 187).

[107] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, Cham, 2015, pp. 234–241 (cited on pages 186, 187).

[108] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional Networks for Biomedical Image Segmentation". *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2015, pp. 234–241 (cited on page 94).

[109] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An Efficient Alternative to SIFT or SURF". *2011 International Conference on Computer Vision*. IEEE, 2011 (cited on pages 10, 91).

[110] O. Sanli, J. Dobruch, M. A. Knowles, M. Burger, M. Alemozaffar, M. E. Nielsen, and Y. Lotan. "Bladder Cancer". *Nature Reviews Disease Primers* 3.1 (2017), p. 17022 (cited on page 7).

[111] N. Schierbaum, J. Rheinlaender, and T. E. Schäffer. "Viscoelastic Properties of Normal and Cancerous Human Breast Cells are Affected Differently by Contact to Adjacent Cells." eng. *Acta biomaterialia* 55 (2017), pp. 239–248 (cited on page 8).

[112] C. Schlick. "An Inexpensive BRDF Model for Physically-based Rendering". *Computer Graphics Forum* 13.3 (1994), pp. 233–246 (cited on page 45).

[113] J. Schüle, V. Aslani, C. Stärk, P. Somers, C. Veil, C. Tarín, A. Herkommer, and O. Sawodny. "In-plane Strain Analysis by Correlating Geometry and Visual Data Through a Gradient-Based Surface Reconstruction". *2023 45rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*. 2023 (cited on pages 168, 172–177).

[114] J. Schüle, J. Haag, P. Somers, C. Veil, C. Tarín, and O. Sawodny. "A Model-based Simultaneous Localization and Mapping Approach for Deformable Bodies*". *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2022, pp. 607–612 (cited on page 11).

[115] J. Schüle, F. Krauß, C. Veil, S. Kunkel, P. Somers, C. Tarín, and O. Sawodny. "Multi-Physical Tissue Modeling of a Human Urinary Bladder". *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*. 2021, pp. 4297–4302 (cited on page 8).

[116] J. Schüle, A. Salehah, P. Somers, N. Harland, C. Tarín, A. Stenzl, and O. Sawodny. *Real-Time Vascular Graph Extraction for Surgical Navigation*. 2022 (cited on pages 92, 189, 190).

[117] J. Schüle, P. Somers, A. R. Salehah, V. Aslani, C. Veil, C. Tarín, A. Herkommer, N. Harland, A. Stenzl, and O. Sawodny. *Differentiable Rendering for Endoscopic Scene Reconstruction*. SSRN Scholarly Paper. Rochester, NY, 2022 (cited on page 76).

[118] Z. Shen and M. Savvides. "MEAL V2: Boosting Vanilla ResNet-50 to 80%+ Top-1 Accuracy on ImageNet without Tricks". *Computing Research Repository* abs/2009.08453 (2020) (cited on page 94).

[119] E. Shkolyar, X. Jia, T. C. Chang, D. Trivedi, K. E. Mach, M. Q.-H. Meng, L. Xing, and J. C. Liao. "Augmented Bladder Tumor Detection Using Deep Learning". *European Urology* 76.6 (2019), pp. 714–718 (cited on page 10).

[120] V. K. Shopov and V. D. Markova. "Application of Hungarian Algorithm for Assignment Problem". *2021 International Conference on Information Technologies (InfoTech)*. 2021, pp. 1–4 (cited on page 150).

[121] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". *3rd International Conference on Learning Representations (ICLR)*. Edited by Y. Bengio and Y. LeCun. 2015 (cited on pages 114, 187, 188).

[122] J. Smith and J. Johnson. *The Science of Color*. Springer, 2022, pp. 67–68 (cited on page 42).

[123] P. Somers, S. Holdenried-Krafft, J. Zahn, J. Schuele, C. Veil, et al. "Cystoscopic Depth Estimation Using Gated Adversarial Domain Adaptation". *Biomedical Engineering Letters* (2023), pp. 1–11 (cited on page 14).

[124] P. Somers, J. Schule, C. Veil, O. Sawodny, and C. Tarin. "Geometric Mapping Evaluation for Real-Time Local Sensor Simulation." eng. Volume 2022. United States, 2022, pp. 609–612 (cited on pages 67, 69).

[125] P. Somers, J. Schüle, C. Tarín, and O. Sawodny. "2D to 3D Segmentation: Inclusion of Prior Information using Random Walk Kalman Filters". *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*. 2021, pp. 4222–4225 (cited on page 8).

[126] J. Song, J. Wang, L. Zhao, S. Huang, and G. Dissanayake. "MIS-SLAM: Real-Time Large-Scale Dense Deformable SLAM System in Minimal Invasive Surgery Based on Heterogeneous Computing". *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4068–4075 (cited on page 11).

[127] A. Stenzl, U. Nagele, M. Kuczyk, K.-D. Sievert, A. Anastasiadis, J. Seibold, and S. Corvin. "Cystectomy – Technical Considerations in Male and Female Patients". *EAU Update Series* 3.3 (2005). Technical Aspects of Radical Cystectomy, pp. 138–146 (cited on page 2).

[128] D. Stoyanov. "Surgical Vision". *Annals of Biomedical Engineering* (2012), pp. 1–14 (cited on pages 10, 13).

[129] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. Chiang, Z. Wu, and X. Ding. *Embracing Imperfect Datasets: A Review of Deep Learning Solutions for Medical Image Segmentation*. 2019 (cited on pages 62, 96).

[130] T. Taketomi, H. Uchiyama, and S. Ikeda. "Visual SLAM Algorithms: A Survey from 2010 to 2016". *IPSJ Transactions on Computer Vision and Applications* 9.1 (2017), p. 16 (cited on page 10).

[131] *Telepräsenz: Bildgebende Systeme, Dokumentation, Beleuchtung, Gerätewagen*. https://www.karlstorz.com/cps/rde/xbcr/karlstorz_assets/ASSETS/3331210.pdf. Accessed: March 27, 2023. 2023 (cited on page 32).

[132] T. Theoharis, G. Papaioannou, and A. Karabassi. "The Magic of the Z-Buffer: A Survey" (2001) (cited on page 44).

[133] *Tissue Differentiation Using Model-based Optical Sensor Systems*. https://www.grk2543.uni-stuttgart.de/en/research/a-sensor-development/a1-optical-sensor-systems/. Accessed: 2023-01-19 (cited on pages 167, 172).

[134] M. Turan, Y. Almalioglu, H. Araujo, E. Konukoglu, and M. Sitti. "A Non-Rigid Map fusion-based Direct SLAM Method for Endoscopic Capsule Robots". *International Journal of Intelligent Robotics and Applications* 1.4 (2017), pp. 399–409 (cited on page 11).

[135] E. A. of Urology. *Non-Muscle-Invasive Bladder Cancer*. https://uroweb.org/. Accessed: 2022-12-14 (cited on page 7).

[136] C. Veil, D. Müller, S. Walz, J. Schüle, P. Somers, C. Tarín, A. Stenzl, and O. Sawodny. "Enhancing Tissue Impedance Measurements Through Modeling of Fluid Flow During Viscoelastic Relaxation". *IEEE Transactions on Biomedical Engineering* 70.2 (2023), pp. 650–658 (cited on page 8).

[137] V. Verma and R. K. Aggarwal. "A Comparative Analysis of Similarity Measures Akin to the Jaccard Index in Collaborative Recommendations: Empirical and Theoretical Perspective". *Social Network Analysis and Mining* 10.1 (2020), p. 43 (cited on page 62).

[138] D. G. Viswanathan. "Features from Accelerated Segment Test (FAST)". *Proceedings of the 10th Workshop on Image Analysis for Multimedia Interactive Services, London, UK*. 2009, pp. 6–8 (cited on page 10).

[139] S. Walz, V. Aslani, O. Sawodny, and A. Stenzl. "Robotic radical cystectomy more precision needed?" eng. *Current Opinion in Urology* 33 (2 2023), pp. 157–162 (cited on page 8).

[140] G. Wang, J. Han, and X. Zhang. "Three-Dimensional Reconstruction of Endoscope Images by a Fast Shape from Shading Method". *Measurement Science and Technology* 20.125801 (2009) (cited on page 13).

[141] H. Wang, Q. Wu, and W. Liu. "Pixel Patterns in Skeletonized Images: Analysis, Identification and Removal". *Pattern Recognition* 45.5 (2012), pp. 1857–1868 (cited on page 104).

[142] L. Wang and Z.-L. Chen. “Rodrigues Vector based 3D Rotation Representation: A Review”. *International Journal of Computer Science and Information Security* 7.2 (2009), pp. 8–14 (cited on page 30).

[143] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. “Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images” (2018) (cited on pages 58–60).

[144] *What Is Camera Calibration? - MATLAB & Simulink - MathWorks Deutschland*. https://de.mathworks.com/help/vision/ug/camera-calibration.html (cited on page 28).

[145] A. R. Widya, Y. Monno, M. Okutomi, S. Suzuki, T. Gotoda, and K. Miki. “Stomach 3D Reconstruction Using Virtual Chromoendoscopic Images”. *IEEE Journal of Translational Engineering in Health and Medicine* 9 (2021), pp. 1–11 (cited on pages 12, 13).

[146] Y. Wu, F. Tang, and H. Li. “Image-based Camera Localization: An Overview”. *Visual Computing for Industry, Biomedicine, and Art* 1 (2018) (cited on page 10).

[147] *Z-Buffer*. https://de.wikipedia.org/wiki/Z-Buffer. Accessed: 2023-01-10 (cited on page 44).

[148] W. Zeng and R. L. Church. “Finding Shortest Paths on Real Road Networks: The Case for A*”. *International Journal of Geographical Information Science* 23.4 (2009), pp. 531–543 (cited on page 151).

[149] O. Zenteno, A. Krebs, S. Treuillet, Y. Lucas, Y. Benezeth, and F. Marzani. “Spatial and Spectral Calibration of a Multispectral-Augmented Endoscopic Prototype”. 2019 (cited on pages 26, 32).

[150] L. Zhang, J. Chen, X. Wang, and Z. Su. “Graph Extraction from Skeletonized Images based on Improved Thinning Algorithm and Pixel Ordering Method”. *Neurocomputing* 245 (2017), pp. 114–126 (cited on pages 92, 104).

[151] R. Zhang, R. Li, S. Zhang, and H. Wu. “Graph Extraction from Skeletonized Images Based on Improved Pixel Ordering and Thinning Algorithm”. *IEEE Access* 6 (2018), pp. 74622–74631 (cited on pages 92, 104).

[152] T. Y. Zhang and C. Y. Suen. “A Fast Parallel Algorithm for Thinning Digital Patterns”. *Commun. ACM* 27.3 (1984), pp. 236–239 (cited on page 104).

[153] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang. “Pattern-Affinitive Propagation Across Depth, Surface Normal and Semantic Segmentation”. 2019, pp. 4101–4110 (cited on page 62).

[154] F. Zhao and X. Tang. “Preprocessing and Postprocessing for Skeleton-based Fingerprint Minutiae Extraction”. *Pattern Recognition* 40.4 (2007), pp. 1270–1281 (cited on pages 103, 104, 106, 107).

[155] Z. Zhou, J. Yang, H. Chen, L. Xu, Q. Dou, C. Ding, and P.-A. Heng. “Unet++: A Nested U-net Architecture for Medical Image Segmentation”. *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 3–11 (cited on pages 94, 96).