Institut für Visualisierung und Interaktive Systeme (VIS)
Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Masterarbeit Nr. 11

Ein echtzeitnaher Ansatz für Structure-from-Motion

Boitumelo Ruf

Studiengang:	Informatik
Prüfer/in:	Prof. Dr.-Ing. Andrés Bruhn
Betreuer/in:	Prof. Dr.-Ing. Andrés Bruhn Dr.-Ing. Tobias Schuchert
Beginn am:	20. Oktober 2014
Beendet am:	21. April 2015
CR-Nummer:	I.2.10, I.4.8, G.1.6

IN KOOPERATION MIT DEM

Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB), Karlsruhe
Abteilung Videoauswertesysteme (VID)

ZUSAMMENFASSUNG

In den letzten Jahren haben dreidimensionale Modelle im alltäglichen Geschehen an Bedeutung gewonnen. So zum Beispiel im Zusammenhang mit Fahrerassistenzsystemen oder der Navigation von autonomen Fahrzeugen, wie Autos oder unbemannten Luftfahrzeugen (UAVs). Dreidimensionale Modelle können aber auch zur Planung und Überwachung von schwer zugänglichen und unbekannten Gebieten genutzt werden.

Im Rahmen dieser Arbeit ist ein Framework zur echtzeitnahen 3D-Rekonstruktion mittels Structure-from-Motion umgesetzt worden. Das System ist primär für die Rekonstruktion von urbanen Gebieten auf Basis von Luftaufnahmen vorgesehen. Da sich diese Arbeit lediglich der dichten Rekonstruktion widmet, wurde vorausgesetzt, dass die Eingangsdaten in Form von Einzelbildern mit entsprechenden Kameraposen gegeben sind. Auf Basis dieser Einzelbilder und Posen, führt das umgesetzte Framework eine dichte Tiefenschätzung für einzelne Keyframes der Eingangssequenz durch. Die resultierende Tiefenschätzung wird in einzelnen Tiefenkarten gespeichert. Diese können anschließend fusioniert und in ein dreidimensionales Modell projiziert werden. Die globale Fusion und Modellerstellung ist ebenfalls nicht als Teil der Arbeit vorgesehen.

Um eine echtzeitnahe Berechnung gewährleisten zu können, erfolgt die Tiefenschätzung in zwei Schritten: Im ersten Schritt wird eine Tiefenkarte On-the-fly berechnet. Als zweiter Schritt wird bei Bedarf die spätere Offline-Berechnung eines detaillierten Modells durchgeführt. Für die On-the-fly Berechnung wird ein Plane-Sweep-Verfahren verwendet, das eine Abtastung der Szene mit unterschiedlichen Ebenenorientierungen erlaubt. Dies soll dabei helfen verschiedene Orientierungen der Objekte und Geländeformen besser zu rekonstruieren. Für die offline durchgeführte Verfeinerung der Tiefenkarte wird ein Variationsansatz auf Basis der Total-Generalized-Variation (TGV) zweiter Ordnung verwendet. Die TGV erlaubt eine Begünstigung von affinen Funktionen innerhalb des Models, wodurch geneigte Oberflächen präziser Rekonstruiert werden können.

Abschließend wird das umgesetzte System mit entsprechenden Benchmarks getestet und evaluiert. Die Tests werden auf einer leistungsstarken Desktop-Hardware durchgeführt. Zur Leistungssteigerung werden die Berechnungen parallelisiert und auf einer Nvidia GeForce GTX 980 ausgeführt. Die Auswertung auf gegebener Hardware zeigt, dass das On-the-fly Plane-Sweep-Verfahren durchaus echtzeitfähig ist. Zudem werden anhand der Evaluation verschiedene Erkenntnisse gewonnen, die für die Weiterentwicklung und Verwendung des Systems wichtig sind.

ABSTRACT

In the past few years, 3d-models have received an increasing importance in everyday events. For example in the context of driver assistance systems or navigation of autonomous vehicles, such as cars or unmanned aerial vehicles (UAVs). In addition, 3d-models can also be used for planning and monitoring of operations, in remote and unknown areas.

This thesis is concerned with the implementation of a framework for near real-time 3d-reconstruction as part of Structure-from-Motion. The system is primarily intended for the reconstruction of urban areas based on aerial imagery. As this work only attends the problem of dense 3d-reconstruction, the camera poses to the corresponding frames are part of the given input data. Based on these images and poses the framework performs a dense reconstruction for single keyframes within the input sequence. The resulting depth estimations are stored in depthmaps that are associated with these keyframes. These depthmaps can then be used to project and create a global 3d-model of the reconstructed scene. The problem of global fusion and projection of the depthmaps is not part of this thesis.

To guarantee the near real-time performance, the depth estimation is performed in two successive steps: The first step performs a live reconstruction and computes the depthmap on-the-fly. In the second step, the initially computed depthmap is refined into a more detailed model. This refinement is performed offline. For the live reconstruction in step 1, a Plane-Sweep method is used that can sample the scene with different plane orientations and sweeping directions. This allows a better reconstruction of different orientations in the scene. The offline refinement in step 2, is performed with a second order Total-Generalize-Variation (TGV) method. The TGV fits affine functions into the model, which allows the reconstruction of slanted surfaces.

Finally the implemented framework is tested and evaluated on appropriate benchmarks. This testing is performed on a powerful desktop hardware. In order to increase the performance the algorithms are parallelized and run on a Nvidia GeForce GTX 980. The evaluation on given hardware shows, that with the Plane-Sweep method the depthmaps are computed in real-time. Furthermore, the evaluation reveals a number of insights that are to be considered for the further improvements and use of the implemented system.

INHALTSVERZEICHNIS

1. EINLEITUNG

In einer Zeit, in der im alltäglichen Geschehen Computer eine immer wichtigere Rolle spielen, gewinnen auch dreidimensionale Modelle mehr an Bedeutung. So zum Beispiel im Zusammenhang mit Fahrerassistenzsystemen, die aktuell bereits in vielen neuen Autos verbaut werden, oder in der Navigation von autonomen Fahrzeugen wie Autos oder unbemannten Luftfahrzeugen (UAVs). Dabei werden die 3D Modelle dazu genutzt die Umgebung des Fahrzeuges abzubilden und eine sichere Navigation zu ermöglichen. Dreidimensionale Modelle können aber auch zur Planung von Einsätzen in schwer zugänglichen und unbekannten Gebieten genutzt werden.

Egal welche Anwendung, solche Modelle müssen in irgendeiner Art und Weise erzeugt werden. Während eine synthetische Erstellung in vielen Fällen zu aufwendig ist und die entsprechenden Modelle meist nicht realitätsgetreu sind, wird eine Möglichkeit benötigt eine Umgebung durch Sensoren abzutasten und daraus ein geeignetes Modell zu erstellen. Im Vergleich zu gewöhnlichen 3D-Sensoren, wie beispielsweise „Laser-Range-Scanners“ haben Videosensoren zwar im Allgemeinen eine geringere Genauigkeit, bieten dafür jedoch eine Vielzahl an Vorteilen. Neben den geringeren Kosten, Größe und Gewicht, bieten sie höhere Auflösungen und sind zudem meist ohnehin in vielen Systemen vorhanden. Gerade bei mobilen Systemen wie Autos, UAVs oder Smartphones spielen diese Faktoren eine große Rolle.

Die 3D-Rekonstruktion mittels Videodaten kann dabei auf verschiedene Arten erfolgen. Die klassische Stereo-Rekonstruktion, mittels zwei nebeneinander montierten Kameras, ist dabei an die menschliche Anatomie angelehnt. Hierbei entsteht die räumliche Wahrnehmung durch die zwei verschiedenen Blickwinkel der beiden Kameras. Eine weitere Strategie ist die monokulare Rekonstruktion. Anstelle von zwei Kameras, werden nur die Daten einer einzelnen Kamera verwendet. Um die fehlende Kamera auszugleichen, muss das System sich entlang oder durch eine Szene bewegen, um diese zu rekonstruieren. Dabei wird angenommen, dass die Szene zwischen aufeinanderfolgenden Bildern sich nicht verändert hat. Dadurch können die Einzelbilder genutzt werden, um verschiedene Blickwinkel auf die Szene zu erhalten. Dieses Vorgehen nennt sich Structure-from-Motion und ist gerade bei mobilen Systemen sehr beliebt, da diese oft Begrenzungen in der Nutzlast o. ä. haben.

In einem aktuellen Projekt des Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung in Karlsruhe soll ein Prototyp umgesetzt werden, mit dem die Rekonstruktion eines urbanen Gebietes aus Videodaten mittels Structure-from-Motion möglich ist. Die Eingangsdaten werden dabei von einem Überflug über das zu rekonstruierende Gebiet gewonnen. Hierbei wird zunächst aus den Einzelbildern des Eingangsvideos die Kamerabewegung geschätzt, um anschließend eine bildbasierte Tiefenschätzung durchzuführen. Aus den berechneten Tiefen lässt sich abschließend ein dreidimensionales Modell der Szene berechnen.

1.1 AUFGABENSTELLUNG

Als Teil eines Structure-from-Motion (SfM) Ansatzes zur Rekonstruktion einer Szene soll sich diese Masterarbeit mit der bildbasierten Tiefenschätzung befassen. Der Vorverarbeitungsschritt, der als Teil der SfM-Pipeline die Bewegung der Kamera schätzt, liefert dabei neben den Einzelbildern des Eingangsvideos auch dazugehörige Kameraposen. Diese bilden einen notwendigen Teil der Rekonstruktion und werden im Rahmen dieser Arbeit als gegeben angenommen. Demzufolge ist der Vorgang der Posenschätzung nicht Teil dieser Arbeit. Des Weiteren werden bei einem vollständigen SfM Verfahren die Tiefenschätzungen zu den Einzelbildern, welche in Form von sogenannten Tiefenkarten gespeichert sind, im Anschluss genutzt, um ein dreidimensionales Modell der Szene zu erstellen. Hierbei müssen die einzelnen Ergebnisse der Schätzung zusammengeführt und passend miteinander verbunden werden, um Überlappungen zu erkennen und auszubessern. Die präzise Fusion der Tiefenkarten sowie das Erstellen des endgültigen dreidimensionalen Modells sind ebenfalls nicht im Rahmen dieser Arbeit vorgesehen.

Eine Anforderung an das System zur Rekonstruktion und damit auch an die einzelnen Abschnitte der SfM-Pipeline, ist die Echtzeitfähigkeit. Das Modell der Szene soll On-the-fly, also noch während der Abtastung der Szene, berechnet werden. Hierbei muss die Berechnung aber nicht auf dem Gerät selber erfolgen, sondern kann auf einer Desktop-Hardware umgesetzt werden. Diese sind in der Regel weitaus leistungsfähiger als mobile-Hardware. Während die spärliche (aus dem Englischen „sparse“) Rekonstruktion lediglich für markante Punkte im Bild die Tiefe berechnet, soll in dieser Arbeit eine dichte Tiefenschätzung erfolgen. Dies bedeutet, dass für jedes Pixel im Eingangsbild die Tiefe geschätzt werden soll, was zu einer lückenlosen Tiefenkarte führt.

Die Anforderung an eine dichte Rekonstruktion schränkt je nach Auflösung der Eingangsbilder jedoch die Echtzeitfähigkeit des Systems ein. Um eine echtzeitnahe Berechnung zu gewährleisten, soll die Tiefenschätzung in zwei Schritten erfolgen: Im ersten Schritt soll eine Tiefenkarte On-the-fly berechnet werden. Diese kann dabei gröber aufgelöst sein und geringere Details enthalten um die Laufzeit zu reduzieren. Als zweiter Schritt soll bei Bedarf die spätere Offline-Berechnung eines detaillierten Modells möglich sein.

Da sogenannte Plane-Sweep-Verfahren aus der aktuellen Literatur auch für große Auflösungen bereits dichte Modelle in Echtzeit berechnen können, soll ein solches Verfahren für die On-the-fly Berechnung verwendet werden. Wie der Name bereits erahnen lässt, wird dabei die Rekonstruktion mittels verschiedener Ebenen durchgeführt. Die diskrete und diskontinuierliche Eigenschaft einer solchen Rekonstruktion führt dazu, dass kleinere Details verloren gehen. Der Grund hierfür ist, dass die Objekte auf die einzelnen Ebenen reduziert werden. Aus diesem Grund soll die Offline durchgeführte Rekonstruktion auf einem Variations-Ansatz basieren. Eine Tiefenschätzung mittels eines Variations-Verfahrens erlaubt eine Anpassung des zu berechnenden Modells an kleine Szenenstrukturen und bietet dadurch eine genauere und detailliertere Rekonstruktion.

Abschließend sind die Ergebnisse des im Rahmen dieser Masterarbeit umgesetzten Systems mit entsprechenden Benchmarks und Datensätzen zu testen und zu evaluieren. Dabei sollen auch die Ergebnisse des Plane-Sweep- und des Variations-Ansatzes bzgl. der Echtzeitfähigkeit und Genauigkeit miteinander verglichen werden.

1.2 GLIEDERUNG

Die folgende Ausarbeitung lässt sich wie folgt gliedern:

Zunächst werden in Kapitel 2 die nötigen Grundlagen für ein Verständnis der darauffolgenden Arbeit erläutert. Hierbei wird zunächst auf die Geometrie und Funktionsweise einer Kamera eingegangen. Im Anschluss daran wird erläutert wie es möglich ist aus einem zweidimensionalen Bild ein dreidimensionales Modell zu berechnen. Darin ist eine detailliertere Erklärung zu der Structure-from-Motion Methodik enthalten. Zum Abschluss des Grundlagenkapitels werden verschiedene Operationen erklärt, die für die Rekonstruktion wichtig sind. Darunter auch das Vorgehen zum Wiederfinden von Bildstrukturen in verschiedenen Aufnahmen.

Nach dem Erläutern der Grundlagen widmet sich Kapitel 3 der Vorstellung und detaillierten Erklärung der verwendeten Verfahren. Zunächst werden hierbei in 3.1 einige verwandte Arbeiten vorgestellt, die sich mit der Thematik der 3D-Rekonstruktion befassen. In 3.2 wird die Rekonstruktion mittels eines Plane-Sweep-Verfahrens vorgestellt. Darin wird auch die Berechnung der homographischen Abbildung hergeleitet. In Kapitel 3.3 wird die Variationsbasierte Berechnung des Modells vorgestellt und erklärt.

Mit dem Wissen über die Berechnung eines dreidimensionalen Modells aus einem zweidimensionalen Bild, wird in 4.1 die eigentliche Umsetzung des Systems für eine echtzeitnahe Rekonstruktion erläutert. Die weiteren Abschnitte des Kapitel 4 widmen sich zusätzlichen Erweiterungen und Techniken, die aus verschiedenen Arbeiten übernommen und adaptiert wurden. Diese beinhalten unter anderem verschiedene Diffusionstensoren, den Umgang mit Verdeckungen, und eine Methodik zur adaptiven Nachbarschaftsgröße.

Bevor die Arbeit in Kapitel 6 noch einmal zusammengefasst und ein Ausblick vorgestellt wird, ist in Kapitel 5 eine Auswertung des umgesetzten Systems zu finden. Hierzu werden zunächst in den Abschnitten 5.1 & 5.2 die Testdatensätze, sowie ein geeignetes Qualitätsmaß zur quantitativen Auswertung vorgestellt. Die experimentelle Ergebnisse und Auswertung werden im Anschluss (Abschnitt 5.3) für die beiden Teilmodule (Plane-Sweep-basierte und Variations-basierte Rekonstruktion) getrennt voneinander vorgestellt und kurz diskutiert. Eine abschließende Diskussion, die die Ergebnisse beider Verfahren berücksichtigt und miteinander in Relation stellt, erfolgt in Abschnitt 5.4. Darin wird auch ein endgültiges Ergebnis der Arbeit vorgestellt.

2. GRUNDLAGEN

Die Thematik der Bildauswertung kann durchaus komplex sein und benötigt einen gewissen Grad an Vorwissen. Aus diesem Grund werden in diesem Kapitel die nötigen Grundlagen für diese Arbeit erläutert. Hierbei wird zunächst auf die Kamerageometrie eingegangen. Anschließend wird erklärt, wie aus einem zweidimensionalen Bild ein dreidimensionales Modell berechnet werden kann. Danach wird kurz auf zwei Methoden eingegangen, die in dieser Arbeit zum Abgleich von Bildstrukturen verwendet werden. Abschließend folgt eine Erläuterung paar mathematischer Operatoren, die für die Bildauswertung nötig sind.

2.1 KAMERAGEOMETRIE

Eine der wesentlichen Aufgaben der Computer Vision ist die Gewinnung von Informationen aus digitalen Bildern. Ein wichtiger Schritt in diesem Prozess ist zu verstehen, wie eine dreidimensionale Szene durch eine Kamera perspektivisch auf ein zweidimensionales Bild abgebildet wird. Bereits Albrecht Dürer hat im frühen 16. Jahrhundert damit begonnen, Objekte und Szenen perspektivisch darzustellen und gilt als Pionier der perspektivischen Malerei. Einer seiner Holzschnitte, „Der Zeichner der Laute“ (vgl. Abb. 2.1), zeigt wie Dürer damals eine perspektivische Zeichnung angefertigt hat.

2.1.1 *Das Lochkamera-Modell*

Die wesentliche Erkenntnis der damaligen Untersuchungen, welche für den weiteren Verlauf der perspektivischen Malerei und der späteren Entwicklung der Kamera grundlegend war, ist, dass alle Punkte eines dreidimensionalen Objekts sich über einen Punkt auf die Bildebene abbilden lassen: Dem *Brennpunkt*. Diese Feststellung wurde in ein Gedankenmodell formuliert, welches wir heute als *Lochkamera* kennen. Diese besteht lediglich aus einer Box, die auf der einen Seite eine Öffnung hat, welche so klein ist, dass nur ein einziger Lichtstrahl hindurch passt. Alle Lichtstrahlen, die dabei von einer Szene ausgehen, werden durch diese infinitesimale Öffnung auf die gegenüberliegende Bildebene projiziert.

In Abbildung 2.2 *(a)* ist diese Projektion schematisch dargestellt. Hierbei liegt der Brennpunkt C auf der *Fokalebene* F und bildet das *optische Zentrum* des Aufbaus. Orthogonal zur Fokalebene verläuft die *optische Achse* und zeigt in Richtung der Szene. Die *Bildebene* I, auf der die Szene abgebildet wird, liegt mit dem Abstand f, aus Sicht der Szene, hinter der Fokalebene. Hierbei wird f als *Brennweite* bezeichnet. Wie oben bereits erwähnt, verläuft von jedem Szenenpunkt M_i ein optischer Strahl durch den Brennpunkt und trifft im Punkt m_i auf die Bildebene (vgl. Abb.2.2 *(a)*). Die Projektion des Brennpunktes auf die Bildebene, oder auch

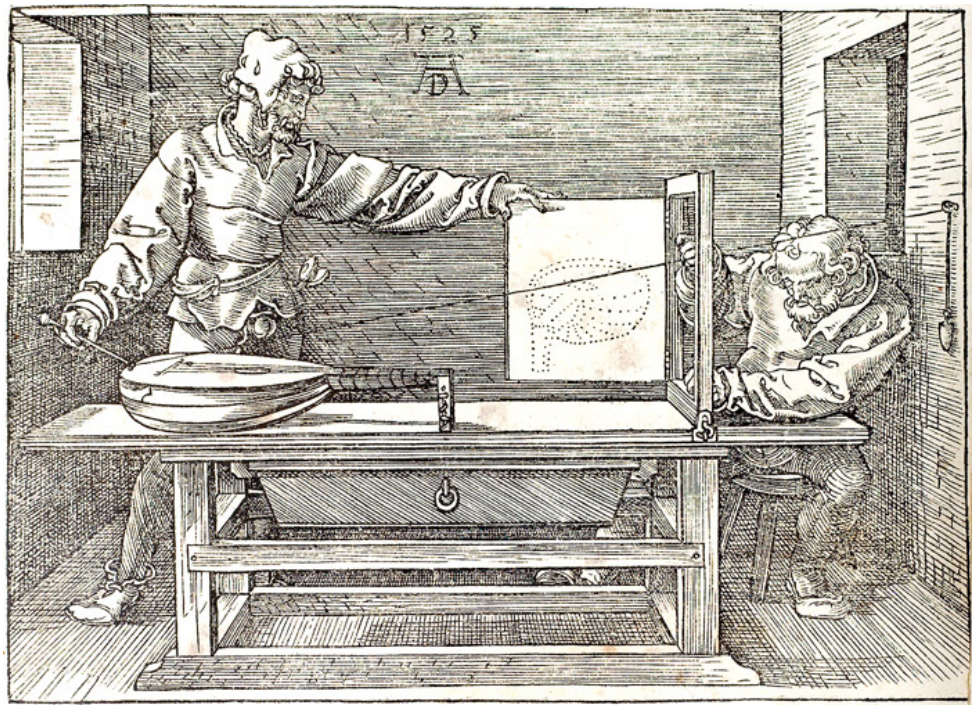

Abbildung 2.1: Albrecht Dürers Holzschnitt „Der Zeichner der Laute“, aus dem Jahr 1525, zeigt Albrecht Dürer, wie er eine perspektivische Zeichnung einer Laute anfertigt. Hierbei benutzte er eine Hilfskonstruktion um die Sichtstrahlen, und damit die perspektivische Wahrnehmung, nachvollziehen zu können. Quelle: http://www.martin-missfeldt.de/perspektive-zeichnen-tutorial/perspektive-albrecht-duerer.php (Zugriff: 16.01.2015)

der Schnittpunkt zwischen Bildebene und optischer Achse, bildet den *Hauptpunkt c*. Dieser liegt im Idealfall im Mittelpunkt der Bildebene.

Ein Nachteil dieser Darstellung ist, dass die entstehende Abbildung der Szene invertiert ist. Dies spielt in der Realität zwar keine Rolle, da der Effekt durch erneute Invertierung ausgeglichen werden kann, dennoch ist eine zusätzliche Invertierung für das Gedankenmodell Lochkamera unpraktisch. Aus diesem Grund wird das Modell noch einmal vereinfacht. Statt der hinter der Fokalebene liegenden Bildebene, wird eine *virtuelle* Bildebene betrachtet, die im Abstand f vor dem Brennpunkt liegt (vgl. Abb. 2.2 *(b)*). Durch diese Vereinfachung haben die abgebildeten Objekte die gleiche Orientierung wie die der realen Szene.

2.1.2 *Projektive Mathematik und homogene Koordinaten*

Mit der Kenntnis über den Aufbau einer Lochkamera, kann nun untersucht werden, wie die einzelnen Szenenpunkte auf die Bildebene projiziert werden. Wird die betrachtete Szene um das optische Zentrum (Brennpunkt) der Kamera zentriert, dann lassen sich alle Punkte M_i der

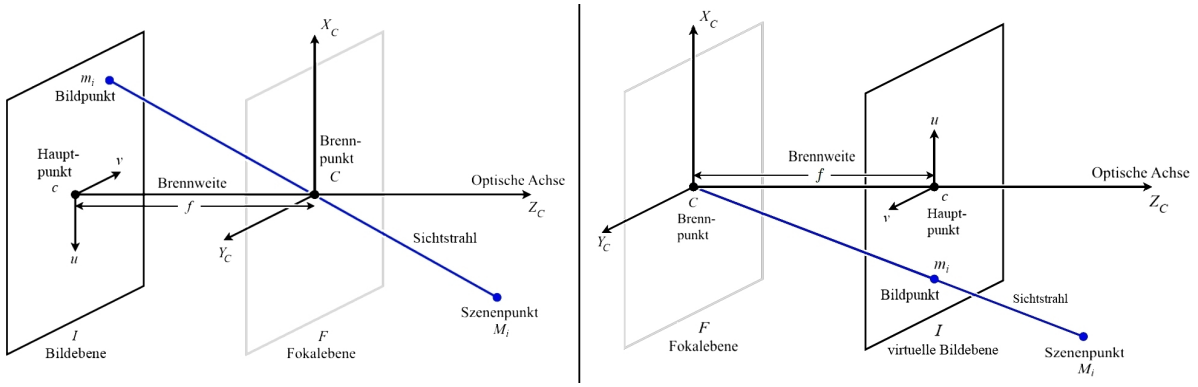

Abbildung 2.2: Von links nach rechts: **(a)** Schematische Darstellung der perspektivischen Projektion einer *Lochkamera*. Ein Szenenpunkt M_i wird über den *Brennpunkt* C auf die Bildebene in den Punkt m_i abgebildet. **(b)** Vereinfachtes Lochkamera-Modell mit einer „virtuellen" Bildebene vor der Fokalebene. Hierdurch haben Bildobjekte dieselbe Orientierung wie die Szenenobjekte. Quelle: Vorlesung zu *Computer Vision*, gehalten von Prof. Andrés Bruhn, Wintersemester 2013/2014, Universität Stuttgart.

Szene durch die Koordinaten X_i, Y_i und Z_i aus Sicht der Kamera eindeutig lokalisieren. Es sollte dabei beachtet werden, dass das Koordinatensystem der Kamera mit X_C, Y_C und Z_C ein rechtshändiges Koordinatensystem ist mit einer positiven z-Richtung entlang der optischen Achse. Auch die *Bildpunkte* m_i auf der Bildebene können durch die Koordinaten u_i und v_i, mit dem Hauptpunkt c als Zentrum, eindeutig bestimmt werden. Der Strahlensatz liefert dabei eine kompakte Gleichung, die diese Koordinaten in Relation stellt:

$$\frac{u_i}{X_i} = \frac{v_i}{Y_i} = \frac{f_i}{Z_i} \quad \rightarrow \quad Z_i u_i = f X_i \ , \ \ Z_i v_i = f Y_i \ . \tag{2.1}$$

Zwar beschreibt Gleichung 2.1 die durch das Lochkamera-Modell dargelegte Geometrie, jedoch ist sie durch die enthaltene Division nicht linear, was zukünftige Rechnungen erschwert. Eine Abhilfe hierfür schaffen die *homogenen Koordinaten*. Dabei lässt sich durch die Hinzunahme einer weiteren Koordinate, also einer weiteren Dimension, die nichtlineare Geometrie der Projektion durch lineare Abbildungen in Form von Matrizen beschreiben. Die Transformation in homogene Koordinaten für einen zweidimensionalen, sowie für einen dreidimensionalen Punkt ist wie folgt:

$$\begin{pmatrix} u \\ v \end{pmatrix} \rightarrow \begin{pmatrix} wu \\ wv \\ w \end{pmatrix} \ , \quad \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} wX \\ wY \\ wZ \\ w \end{pmatrix} . \tag{2.2}$$

Die Rücktransformation ist dabei durch

$$\begin{pmatrix} wu \\ wv \\ w \end{pmatrix} \rightarrow \begin{pmatrix} \frac{wu}{w} \\ \frac{wv}{w} \\ \frac{w}{w} \end{pmatrix} \rightarrow \begin{pmatrix} u \\ v \end{pmatrix} \quad , \quad \begin{pmatrix} wX \\ wY \\ wZ \\ w \end{pmatrix} \rightarrow \begin{pmatrix} \frac{wX}{w} \\ \frac{wY}{w} \\ \frac{wZ}{w} \\ \frac{w}{w} \end{pmatrix} \rightarrow \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{2.3}$$

gegeben. Bei der Transformation in homogene Koordinaten ist zu beachten, dass die bestehenden Koordinaten u und v, bzw. X, Y, und Z, ebenfalls mit w erweitert werden müssen um zu gewährleisten, dass eine Rücktransformation wieder zu dem ursprünglichen Punkt führt. Hierbei ist $w \in \mathbb{R}\backslash 0$ definiert. Für $w = 0$ beschreiben die homogenen Koordinaten Punkte in der Unendlichkeit und lassen sich dadurch nicht im Euklidischen Raum darstellen. Für die Verwendung in der Computer Vision wird in der Regel $w = 1$ gewählt.

Die in Gleichung 2.2 aufgestellten Beziehungen der Kamerageometrie lassen sich nun mit Hilfe der homogenen Koordinaten linear durch eine Matrixmultiplikation wie folgt umschreiben:

$$\widetilde{m}_i = \begin{pmatrix} Z_i u_i \\ Z_i v_i \\ Z_i \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = P' \cdot \widetilde{M}_i. \tag{2.4}$$

Hierbei gibt die Tilde (~) an, dass die Punkte in homogene Koordinaten dargestellt sind. Die hinzugekommene Matrix P' wird als *Projektionsmatrix* bezeichnet. Durch das Verschwinden der Dimension Z einer Projektion aus dem dreidimensionalen Raum auf eine zweidimensionale Ebene wird impliziert, dass alle Punkte, die auf demselben optischen Strahl liegen, auf einen einzigen Bildpunkt abgebildet werden. Dies führt zum Verlust der Tiefeninformationen einer Szene.

2.1.3 *Extrinsische und intrinsische Kameraparameter*

Im vorherigen Kapitel wird die Transformation eines Punktes aus einem im Brennpunkt zentrierten Koordinatensystems in das der Bildebene vorgestellt. Für die spätere 3D-Rekonstruktion werden noch zwei weitere Koordinatensysteme und damit zwei weitere Transformationen benötigt. Auch diese Transformationen werden durch lineare Matrixmultiplikationen realisiert, welche aus den *extrinsischen* und *intrinsischen* Kameraparametern zusammengesetzt sind.

In der Computer Vision wird zunächst immer von einem externen Koordinatensystem ausgegangen. Dies wird als *3D-Weltkoordinatensystem* bezeichnet und ist in einem beliebigen Referenzpunkt O zentriert. In diesem Weltkoordinatensystem liegt das *3D-Kamerakoordinatensystem*, welches aus dem vorherigen Kapitel bekannt ist. Um von dem Weltkoordinatensystem

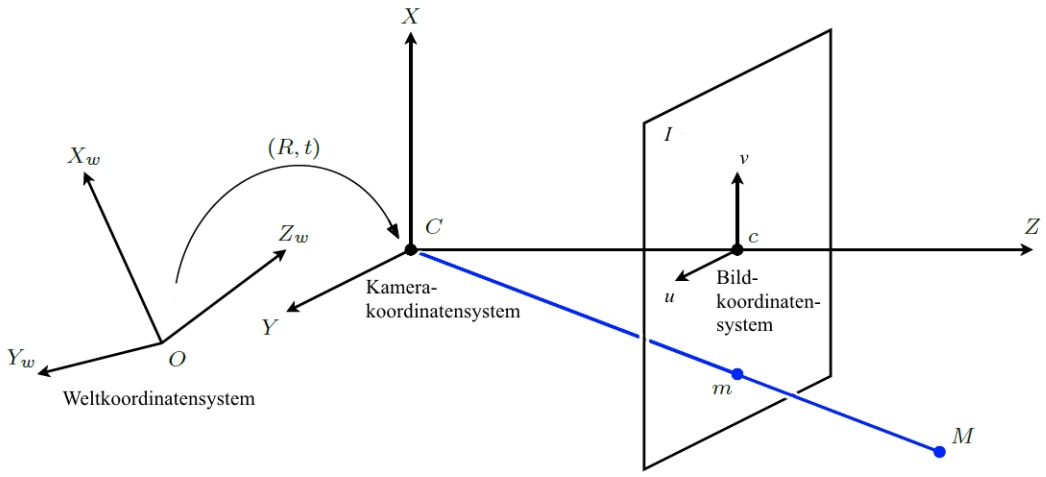

Abbildung 2.3: Extrinsische Transformation aus dem Weltkoordinatensystem in das Kamerakoordinatensystem. Die Transformation ist durch die relative Rotation R und Translation $\vec{t}$ gegeben. Quelle: Vorlesung zu *Computer Vision*, gehalten von Prof. Andrés Bruhn, Wintersemester 2013/2014, Universität Stuttgart.

in das der Kamera zu gelangen wird die relative Translation $\vec{t}$ und Rotation R zwischen den Zentren (O und C) der beiden Koordinatensysteme benötigt (vgl. Abb. 2.3). Die Translationsmatrix T setzt sich aus dem dreidimensionalen Translationsvektor $\vec{t} = (t_1, t_2, t_3)^T$ zusammen und ist wie folgt definiert:

$$T := \begin{pmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \end{pmatrix}. \tag{2.5}$$

Die Rotationsmatrix wird multiplikativ aus drei Drehmatrizen zusammengesetzt. Jede der drei Drehmatrizen beschreibt die Rotation um eine der drei Koordinatenachsen. Hierbei wird die Stärke der Drehung durch die entsprechenden Winkel φ_x, φ_y und φ_z angegeben. Die Formel zur Berechnung der Rotationsmatrix ist durch

$$\begin{aligned} R &:= \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{pmatrix} \\ &= \begin{pmatrix} cos\varphi_z & -sin\varphi_z & 0 \\ sin\varphi_z & cos\varphi_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} cos\varphi_y & 0 & sin\varphi_y \\ 0 & 1 & 0 \\ -sin\varphi_y & 0 & cos\varphi_y \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\varphi_x & -sin\varphi_x \\ 0 & sin\varphi_x & cos\varphi_x \end{pmatrix} \end{aligned} \tag{2.6}$$

gegeben. Da eine Matrixmultiplikation von rechts nach links erfolgt, wird eine Rotation, in der Reihenfolge X-Y-Z, gemäß Gleichung 2.6 durchgeführt. Die oben notierten Drehmatrizen

beschreiben eine aktive Drehung um die jeweilige Koordinatenachse. Das heißt, dass das Koordinatensystem fix bleibt und nur der Punkt sich mit gegebenen Winkel um die entsprechende Koordinatenachse dreht. Das Gegenteil zu einer aktiven Drehung ist die passive Drehung. Hierbei bleibt der Punkt fest und das Koordinatensystem wird transformiert. Eine solche passive Drehung um eine Koordinatenachse wird mit der Inversen der jeweiligen Drehmatrix erzielt.

Die relativen Bewegungen werden als extrinsische Kameraparameter bezeichnet und ergeben zusammen die *extrinsische Kameramatrix* A_{ext}. Auch hier werden homogene Koordinaten verwendet, wodurch die Translations-, Rotations- und extrinsische Kameramatrix als 4×4 Matrizen formuliert werden können. Die Multiplikation der Rotations- und Translationsmatrix ergibt wie folgt die extrinsische Kameramatrix:

$$\begin{aligned} A_{ext} := T \cdot R &= \begin{pmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & 0 \\ r_{2,1} & r_{2,2} & r_{2,3} & 0 \\ r_{3,1} & r_{3,2} & r_{3,3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \neq R \cdot T. \end{aligned} \tag{2.7}$$

Diese Multiplikation ist dabei nicht kommutativ, wodurch sich die Reihenfolgen der Transformationen nicht vertauschen lassen. Hierbei wird die Verschiebung vor der Rotation durchgeführt.

Mit der extrinsischen Kameramatrix A_{ext} und der Projektionsmatrix P' erfolgt die Transformation aus dem Weltkoordinatensystem in das der Bildebene. Das zweidimensionale Bildkoordinatensystem in der Bildebene hat dabei den Ursprung im Hauptpunkt c. In der Bildverarbeitung wird meist jedoch die linke obere Ecke des Bildes als Nullpunkt des *2D-Pixelkoordinatensystems* festgelegt. Die Adressierung der Pixel eines Bildes beginnt also mit den Koordinaten $(0, 0)$ und läuft bis (Bildbreite – 1, Bildhöhe – 1), wobei das erstere die linke obere Ecke und das letztere die rechte untere Ecke des Bildes identifiziert.

Die Transformation von dem zweidimensionalen Koordinatensystem der Bildebene in das idealisierte 2D-Pixelkoordinatensystem erfolgt mit der 3×3 großen *intrinsischen Kameramatrix*

$$A_{int} := \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.8}$$

welche sich aus den intrinsischen Kameraparametern ergibt. Die intrinsischen Parameter beschreiben dabei die Geometrie und die Lage der Bildebene relativ zum Bildsensor, der das Bild schlussendlich aufzeichnet.

- x_0 und y_0 identifizieren die Position des Hauptpunktes c.
- α_x und α_y geben den Skalierungsfaktor in x bzw. y Richtung an.
- s gibt die Schiefe an. Diese entsteht, wenn Bildsensor nicht orthogonal zur optischen Achse, wodurch die Koordinatenachsen nicht rechtwinklig zueinander stehen.

Mit den in Gleichungen 2.4, 2.7 und 2.8 vorgestellten Matrizen kann nun eine Projektion eines in homogenen Koordinaten gegebenen Punktes aus dem 3D-Weltkoordinatensystem in zweidimensionale Pixelkoordinaten (ebenfalls in homogenen Koordinaten) erfolgen. Hierzu werden die intrinsische Kameramatrix A_{int}, die Projektionsmatrix P' und die extrinsische Kameramatrix A_{ext} wie folgt zu der 3×4 *vollständigen Projektionsmatrix* P_{full} verkettet:

$$P_{full} := \underbrace{A_{int} \cdot P'}_{\substack{\text{Kalibrierungsmatrix } K}} \cdot A_{ext} = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{pmatrix}. \tag{2.9}$$

Hierbei wird die Teilverkettung zwischen der intrinsischen Matrix und der Projektionsmatrix als *Kamera-Kalibrierungsmatrix* K bezeichnet. Diese 3×4 Matrix beinhaltet alle internen Parameter einer Kamera, die für eine Projektion einer dreidimensionalen Szene in ein zweidimensionales Bild, in Pixelkoordinaten von $(0, 0)$ bis (Breite - 1, Höhe – 1), nötig sind.

In der Theorie wird meist angenommen, dass keine Skalierung stattfindet, dass der Bildsensor orthogonal zur optischen Achse steht und dass der Hauptpunkt c im Mittelpunkt des Bildes liegt. Eine solche theoretische Kalibrierungsmatrix K für eine Kamera mit der Brennweite $f = 615$, wie sie im neuen *Tsukuba-Stereo-Datensatz* (vgl. Kap. 5.1) verwendet wird, und einer Bildgröße von 640×480 Pixel ist durch

$$K := \begin{pmatrix} f\alpha_x & s & x_0 & 0 \\ 0 & f\alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 615 & 0 & 320 & 0 \\ 0 & 615 & 240 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.10}$$

gegeben.

Zusammenfassend zeigt Abbildung 2.4 wie die Transformation eines Punktes aus dem 3D-Weltkoordinatensystem in das 2D-Pixelkoordinatensystem erfolgt und welche linearen Abbildungen in welcher Transformation eine Rolle spielen. Im Ersten Schritt wird das Weltkoordinatensystem, welches den Ursprung in einem beliebigen Referenzpunkt hat, in das 3D-Kamerakoordinatensystem transformiert und somit das System im Brennpunkt der Kamera zentriert. Dies geschieht mit Hilfe der extrinsischen Kameramatrix A_{ext}, die die Informationen der relativen Kameraposition zum Referenzpunkt des Weltkoordinatensystem enthält. Im nächsten Schritt wird durch die Projektionsmatrix P' der dreidimensionale Punkt (X_C, Y_C, Z_C) in einen zweidimensionalen Punkt (u, v) auf der Bildebene projiziert. Der Ursprung des 2D-Bildkoordinatensystem liegt dabei im Hauptpunkt c, welcher die Projektion des Brennpunktes

auf die Bildebene darstellt. Die letzte Transformation berücksichtigt die interne Geometrie des Bildsensors und bildet den Bildpunkt auf ein Pixel ab. Dies erfolgt durch die intrinsische Kameramatrix A_{int}. Das Pixel ist nun durch die Koordinaten (x, y) mit dem Ursprung in der linken oberen Ecke $(0, 0)$ eindeutig identifizierbar. Die gesamte Projektion eines dreidimensionalen Punktes aus dem Weltkoordinatensystem in einen Punkt in Pixelkoordinaten mittels der vollständigen Projektionsmatrix P_{full} kann durch

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{pmatrix} \cdot \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix} \tag{2.11}$$

beschrieben werden.

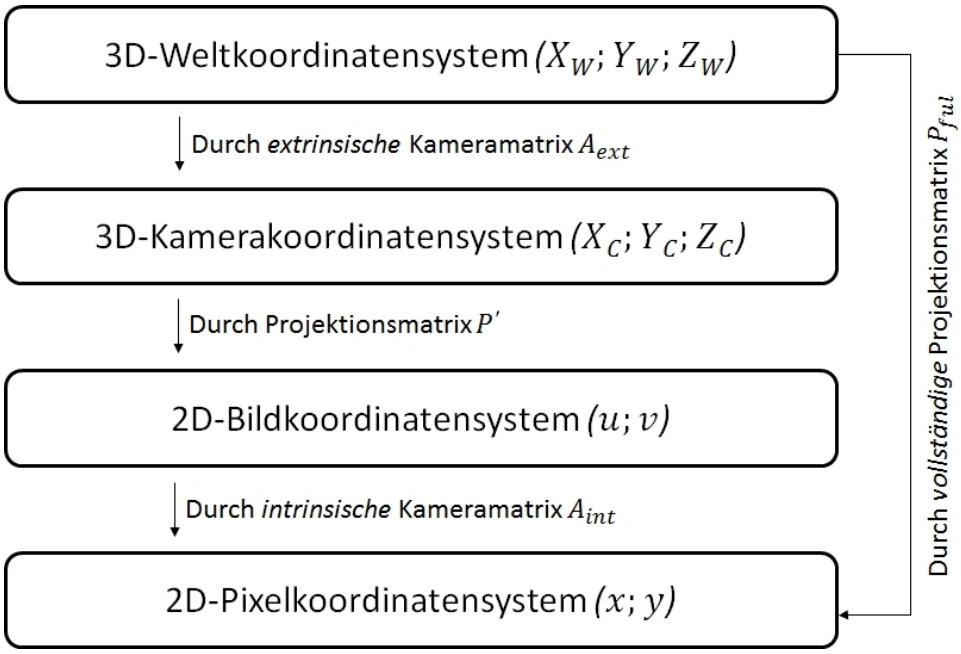

Abbildung 2.4: Schrittweise Transformation eines Punktes aus dem Weltkoordinatensystem in das Pixelkoordinatensystem. Die Verkettung der einzelnen Transformationsmatrizen ergeben die vollständige Projektionsmatrix P_{full}. Quelle: Vorlesung zu *Korrespondenzprobleme in Computer Vision*, gehalten von Prof. Andrés Bruhn, Sommersemester 2014, Universität Stuttgart.

2.2 VON 2D ZU 3D

Im vorherigen Kapitel wurde die Kamerageometrie und die damit verbundene perspektivische Projektion einer Lochkamera vorgestellt. Bekanntlich geht bei der Projektion von 3D auf 2D jedoch die Tiefeninformation der Objekte, also die Information wie weit sie vom Betrachten entfernt sind, verloren. Für die Rekonstruktion der im Bild abgebildeten Szene in den dreidimensionalen Raum muss diese Projektion wieder rückgängig gemacht und damit die Tiefe jedes Objektes geschätzt werden. Diese Tiefenschätzung ist der wesentliche Bestandteil der 3D-Rekonstruktion und erfordert zusätzliche Informationen. Da im Rahmen dieser Arbeit nur mit Bildern gewöhnlicher Digitalkameras gearbeitet wird, sind Tiefeninformationen von Entfernungsmessern nicht vorhanden. Aus diesem Grund werden für die Rekonstruktion zwei oder mehr Bilder der gleichen Szene aus verschieden Blickwinkeln herangezogen, um damit die zusätzlichen Informationen zu ermitteln.

2.2.1 *Stereoskopie*

Das Betrachten einer Szene mit zwei Kameras oder zwei Augen wird als *Stereoskopie*, kurz *Stereo*, bezeichnet und ermöglicht eine räumliche Wahrnehmung der Szene. Diese Wahrnehmung resultiert aus der Tatsache, dass die Objekte der Szene in den beiden Bildern an verschiedenen Orten platziert sind. Abbildung 2.5 zeigt eine stereoskopische Aufnahme einer statischen Szene. Es ist zu erkennen, dass die Szenenobjekte im rechten Bild weiter links erscheinen als im linken Bild. Diese Verschiebung entsteht durch die verschiedenen Blickwinkel der Kameras auf die Szene und wird als *Parallaxe* bezeichnet. Diese hängt von der räumlichen Platzierung der Objekte innerhalb der Szene ab. Gegenstände die näher am Betrachter sind (z.B. die Büste oder die Lampe) weisen eine größere Parallaxe zwischen den zwei Aufnahmen als Hintergrundobjekte (z.B. die Kamera).

Für eine Stereoaufnahme werden im einfachsten Fall zwei Kameras so nebeneinander platziert, dass deren optische Achsen parallel zueinander stehen und in dieselbe Richtung zeigen. Zusätzlich steht dabei die *Baseline*, d.h. die Verbindungslinie der Brennpunkte beider Kameras, orthogonal zu den optischen Achsen (vgl. Abb. 2.6 *(a)*). Ein solcher *orthoparalleler* Aufbau ist ein Spezialfall der Stereoskopie, da die beiden Kameras so zueinander ausgerichtet sind, dass die Objekte in den Kameraaufnahmen nur horizontal zueinander verschoben sind (vgl. Abb. 2.5). Durch die Untersuchung der Objektverschiebungen zwischen den zwei Aufnahmen kann die relative Tiefe der Objekte zueinander ermittelt werden. Sind die genauen Parameter, wie die Pixelkoordinaten (x, y) der Bildpunkte, die Brennweite f der Kameras, und die Länge der Baseline b bekannt, können die genauen Koordinaten des entsprechenden dreidimensionalen Szenenpunkt berechnet werden.

Gemäß Abbildung 2.6 *(b)* sei der Szenenpunkt $M = (X_C, Y_C, Z_C)^T$ im Kamerakoordinatensystem, welches im Brennpunkt der linken Kamera zentriert ist, gesucht. Dabei sind die Länge b der Baseline, die Brennweite f der Kameras, die in diesem Beispiel für beide Kameras gleich ist, und die Pixelkoordinaten der jeweiligen Bildpunkte $m_1 = (x_1, y_1)$ und $m_2 = (x_2, y_2)$

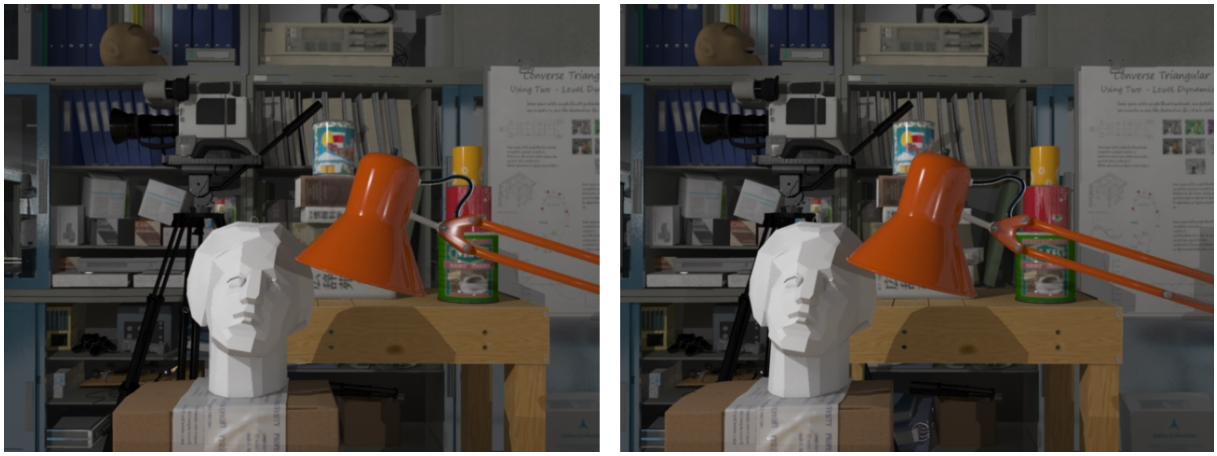

Abbildung 2.5: Stereoaufnahme des neuen Tsukuba-Stereo-Datensatzes (vgl. Kap. 5.1). Erkennbare Verschiebung der Objekte zwischen linker und rechter Aufnahmen. Vordergrundobjekte haben dabei eine größere Verschiebung als Hintergrundobjekte.

bekannt. Nach den Ähnlichkeitssätzen für Dreiecke folgen die Gleichungen in 2.12, wobei erstere auf der Ähnlichkeit der Dreiecke P_1MC_1 und $c_1m_1C_1$, und die zweite auf der Ähnlichkeit der Dreiecke P_2MC_2 und $c_2m_2C_2$ beruht. Auflösen der Gleichungen

$$\frac{X_C}{Z_C} = \frac{u_1}{f} = \frac{x_1 - c_1}{f} \quad \text{und} \quad \frac{X_C - b}{Z_c} = \frac{u_2}{f} = \frac{x_2 - c_2}{f} \tag{2.12}$$

nach X_C und anschließendes Einsetzen führt zu

$$Z_C = \frac{b \cdot f}{u_1 - u_2} \quad \text{mit} \quad u_1 := x_1 - c_1 \,, \quad u_2 := x_2 - c_2 \,. \tag{2.13}$$

Wie bereits in Abbildung 2.5 visualisiert ist, zeigt Gleichung 2.13, dass die Tiefe invers proportional zur Verschiebung der Objekte zwischen den Bildern ist. Ist die Verschiebung bekannt, ist die Tiefe des Objektes rechnerisch effizient zu ermitteln. Jedoch stellt die Berechnung der Verschiebung eine schwierige Aufgabe dar. Eine Schwierigkeit hierbei liegt in der Suche zusammengehörender Bildpunkte in den einzelnen Bildern. Das heißt, dass die jeweiligen Szenenobjekte in den Bildern eindeutig einander zugeordnet werden müssen. Solche zusammengehörende Bildpunkte nennen sich *Punktkorrespondenzen*. Wird eine Struktur in Bild 1, einer anderen Struktur in Bild 2 zugeordnet, welche nicht zu demselben Objekt gehört, so verfälscht dies die Tiefenschätzung. Mehr zum Abgleich von Bildstrukturen findet sich in Kapitel 2.3. Eine weitere Schwierigkeit ist die Wahl des Abstandes der Kameras zueinander. Die Verschiebung zwischen den Bildern kann nur in diskreter Pixelgenauigkeit gemessen werden, was für eine lange Baseline spricht. Je weiter die Kameras jedoch auseinander sind, desto mehr können Szenenveränderungen aufgrund von Verdeckungen auftreten. Dies führt zu Mehrdeutigkeiten in der Strukturzuordnung zwischen den Bildern.

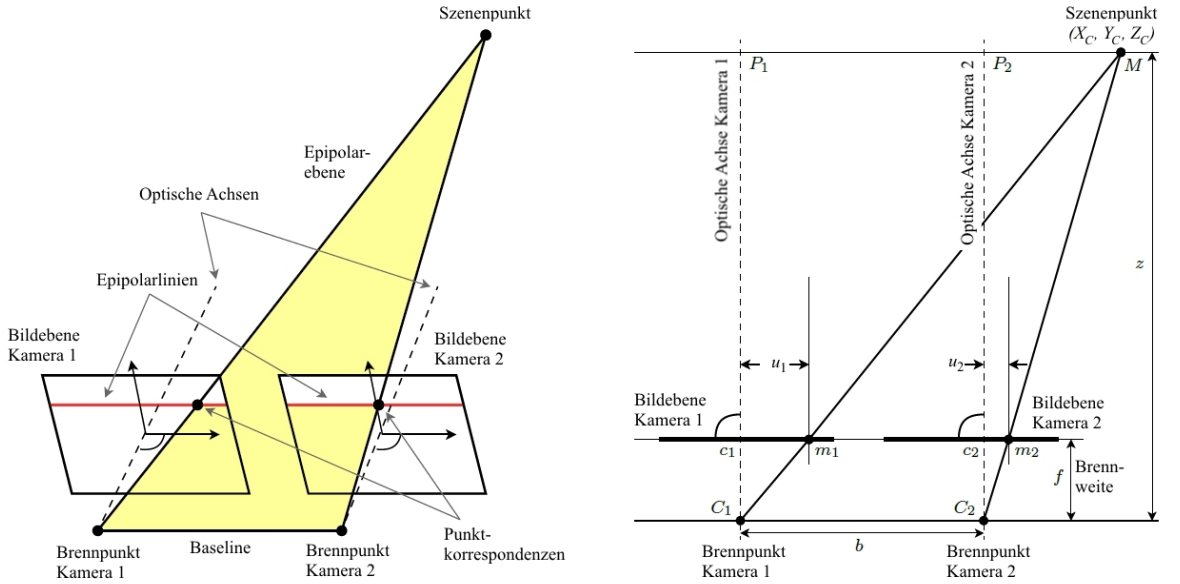

Abbildung 2.6: Von links nach rechts: **(a)** Orthoparalleler Stereoskopischer Kameraaufbau. Parallele Ausrichtung der optischen Achsen der beiden Kameras. Parallele Epipolarlinien. Epipolarebene wird durch den Szenenpunkt und die Brennpunkte der beiden Kameras aufgespannt. **(b)** Berechnung der Koordinaten des Szenenpunktes aus einem *orthoparalleler stereoskopischer* Kameraaufbau. Berechnung erfolgt mittels den Ähnlichkeitssätzen für Dreiecke. Quelle: Vorlesung zu *Computer Vision*, gehalten von Prof. Andrés Bruhn, Wintersemester 2013/2014, Universität Stuttgart.

2.2.2 *Epipolargeometrie*

Wichtige Hilfsstrukturen in der Stereoskopie sind die *Epipolarebene* und die damit verbundenen *Epipolarlinien* . Die Epipolarebene wird durch den dreidimensionalen Szenenpunkt M_i und die Brennpunkte C der beiden Kameras aufgespannt (vgl. Abb. 2.6 *(a)*). Die Epipolarlinien stellen die Schnittlinien der Epipolarebene mit den Bildebenen der beiden Kameras dar. Dies bedeutet zugleich, dass die Epipolarlinie die Projektion des Sichtstrahls durch den Bildpunkt m_i^1 des ersten Bildes auf die Bildebene der zweiten Kamera ist (vgl. Abb. 2.7). Sie gibt somit den möglichen Aufenthaltsort der Punktkorrespondenz zu m_i^1 im zweiten Bild an. Somit wird der Suchraum nach Punktkorrespondenzen zwischen den Bildern durch die Epipolarlinien eingeschränkt. Denn aufgrund der oben genannten Konstruktion der Ebene und der Linien, können zusammengehörige Punktkorrespondenzen nur auf den Epipolarlinien liegen, was den Suchraum von zwei Dimensionen auf eine Dimension reduziert.

Im bisher behandelten orthoparallelen Kameraaufbau sind die Epipolarlinien parallel und liegen in beiden Bildern auf derselben Bildzeile. Dies entspricht einer einfachen Parallaxe in horizontaler Richtung. Es wird in diesem Fall auch von zwei rektifizierten Bildern gesprochen. Häufig werden Bilder im Rahmen der 3D-Rekonstruktion zuerst rektifiziert um die Suche

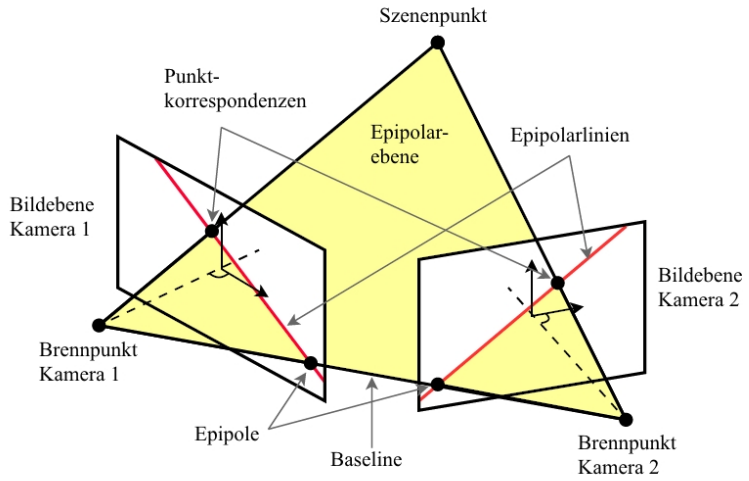

Abbildung 2.7: Allgemeiner *stereoskopischer* Kameraaufbau. Zusammenlaufende optische Achsen der Kameras. Epipolarlinien stehen geneigt zueinander. Epipolarlinien bilden die Projektion des Sichtstrahls aus der jeweils anderen Kamera. Damit bestimmen die Epipolarlinien den möglichen Aufenthaltsort der entsprechenden Punktkorrespondenz. Quelle: Vorlesung zu *Computer Vision*, gehalten von Prof. Andrés Bruhn, Wintersemester 2013/2014, Universität Stuttgart.

nach Punktkorrespondenzen zu erleichtern. Im Allgemeinen sind die Kameras jedoch nicht orthoparallel zueinander ausgerichtet, sondern haben zusammenlaufende optische Achsen (vgl. Abb. 2.7). In diesem Fall liegen die Epipolarlinien nicht parallel zueinander sondern sind zueinander geneigt.

Die, durch die Epipolarebene und die Epipolarlinien herbeigeführte, geometrische Beziehung wird formal durch die *Epipolarbedingung*

$$\widetilde{m}_2^T \cdot F \cdot \widetilde{m}_1 = 0 \tag{2.14}$$

beschrieben. Hierbei bezeichnet F die 3×3 große *Fundamentalmatrix*. Sie wird aus der relativen Translation und Rotation zwischen den zwei Kameras berechnet. Ist die Fundamentalmatrix bekannt, kann für jeden Punkt m_1 im ersten Bild die entsprechende Epipolarlinie l_2 im zweiten Bild gemäß

$$\widetilde{m}_2^T \cdot l_2 = 0\,, \quad l_2 = F \cdot \widetilde{m}_1 \tag{2.15}$$

bestimmt werden. In umgekehrter Richtung gilt:

$$\widetilde{m}_1^T \cdot l_1 = 0\,, \quad l_1 = F^T \cdot \widetilde{m}_2\,. \tag{2.16}$$

Hierin beschreiben $l_i = (a, b, c)^T$ die Epipolarlinien der Form $ax + by + c = 0$. Sind die relativen Bewegungen zwischen den Kameras und damit die Fundamentalmatrix nicht bekannt, kann sie z. B. durch eine aufwendige „Brute-Force-Suche“ nach Bildkorrespondenzen

Abbildung 2.8: Von links nach rechts: **(a)** Aufnahme aus dem neuen Tsukuba-Stereo-Datensatz (vgl. Kap. 5.1). **(b)** Eigenfärbte Tiefenkarte. Schwarz steht für große und Weiß für kleine Entfernungen zwischen Kamera und Szenenobjekte.

geschätzt werden. Bleibt die relative Position der Kameras zueinander gleich, so muss die Fundamentalmatrix nur einmal berechnet werden.

2.2.3 *Tiefenkarten*

Ein weiteres wichtiges Konstrukt im Zusammenhang der 3D-Rekonstruktion ist die sogenannte *Tiefenkarte*. Wie der Name bereits vermuten lässt, enthält sie die Tiefe der jeweiligen Szenenobjekte. Hierbei ist eine Tiefenkarte nichts anderes als ein zweidimensionales Bild der Szene, welches anstelle von Farbintensitäten, Entfernungsinformationen enthält. Zur Darstellung der Tiefenkarte werden diese Tiefeninformationen normiert und farblich markiert. Hierbei gibt es verschiedene Methoden, wie eine Tiefenkarte eingefärbt werden kann. So z. B. entsprechend einer Grauwertskala oder gemäß des HSV-Farbraums. Im Rahmen dieser Arbeit werden die Tiefenkarten mittels verschiedener Graustufen eingefärbt. Dabei repräsentiert Schwarz eine große, und Weiß eine geringe Entfernung. Abbildung 2.8 zeigt exemplarisch eine solche eingefärbte Tiefenkarte.

Sind die Tiefenkarten der zu rekonstruierenden Szene, sowie die intrinsischen Parameter der Kamera vorhanden, so kann die Tiefenkarte problemlos in eine dreidimensionale Punktwolke projiziert werden. Dabei werden zunächst die in der Tiefenkarte enthalten Informationen als Z-Koordinate des dreidimensionalen Szenenpunkt gewählt. Anschließend können durch die in Gleichung 2.1 aufgestellte Beziehungen, sowie der zusätzlichen Kenntnis über die Brennweite und die Pixelkoordinaten, die restlichen Kamerakoordinaten des Szenenpunktes berechnet werden. Dieser dadurch ermittelte Szenenpunkt liegt dabei im Koordinatensystem der Kamera.

Vergleichbar mit der Tiefenkarte gibt es ein weiteres, sehr ähnliches, Konstrukt: die sogenannte Disparitätskarte. Ähnlich zur Tiefenkarte enthält diese dabei die Disparität der Szenenobjekte, die sich durch die Betrachtung der Szene aus verschiedenen Blickwinkeln ergibt. Mittels dieser Disparitätskarte kann jedoch lediglich die relative Tiefe der Objekte zueinander angegeben werden. Diese wird meistens dann verwendet, wenn keine genauen Angaben zu den Positionen der Kamera gemacht werden können.

2.2.4 *Structure-from-Motion*

Die Stereoskopie ist ein effizientes Verfahren für die Rekonstruktion einer drei dimensionalen Szene aus Bilddaten und liefert dabei sehr gute Ergebnisse. Dennoch hat sie ihre Grenzen und ist für manche Anwendungsfälle nicht praktikabel. So werden für das Verfahren der Stereoskopie zwei Kameras benötigt, was nicht nur die Kosten für ein System erhöht, sondern beispielsweise mobile Systeme durch mehr Gewicht und größere Dimensionen zusätzlich einschränkt.

Gerade für mobile Aufbauten ist das sogenannte *Structure-from-Motion (SfM)* Verfahren zur 3D-Rekonstruktion sehr gut geeignet. Es benötigt nur eine Kamera, wobei es die fehlende Kamera durch eine Eigenbewegung des Systems ausgleicht. Genauer gesagt wird bei SfM das 3D-Modell aus einem Video, also einer Reihe von Einzelbildern rekonstruiert. Dabei ist es wichtig, dass das Video nicht von einer statischen Position aus aufgenommen wird, vielmehr sollte die Kamera sich dabei mit Blick auf die zu rekonstruierende Szene bewegen. Bei der Rekonstruktion mittels SfM werden zwei oder mehr Einzelbilder der Eingangssequenz herangezogen, für die angenommen wird, dass sie dieselbe Szene aus verschiedenen Blickrichtungen betrachten und die Szene zwischen den Einzelbildern statisch (unverändert) geblieben ist. Abbildung 2.9 zeigt die einzelnen Verarbeitungsschritte, die bei SfM durchlaufen werden:

- **Tracking:** Als erster Schritt werden signifikante Bildstrukturen über mehrere Einzelbilder des Eingangsvideos hinweg verfolgt. Solche charakteristische Strukturen sind meist eindeutig identifizierbare Bildbereiche wie beispielsweise Ecken oder Kanten von Objekten. Mit den bereits bekannten intrinsischen Kameraparametern und der Verschiebung dieser charakteristischen Strukturen kann die Bewegung der Kamera und damit die relative Bewegung der Einzelbilder zueinander geschätzt werden. Bei SfM ist es nicht notwendig, dass jedes Einzelbild des Eingangsvideos zur Rekonstruktion verwendet wird. Grund hierfür ist, dass die Bildrate des Videos meistens so hoch ist, dass sich zwischen direkt aufeinanderfolgenden Einzelbilder die Kamera noch nicht weit genug bewegt hat. Aus diesem Grund werden lediglich sogenannte *Keyframes* und die dazugehörigen Posen an den nächsten Verarbeitungsschritt weitergegeben.

- 3**D-Rekonstruktion:** Mittels der vom Tracking erhaltenen Einzelbildern und den dazugehörigen Posen wird in diesem Verarbeitungsschritt eine Rekonstruktion der Szene durchgeführt. Die Vorgehensweise dieser Rekonstruktion ist dabei schematisch gleich wie die im Rahmen der Stereoskopie vorgestellte Rekonstruktion. Aus der Verschiebung

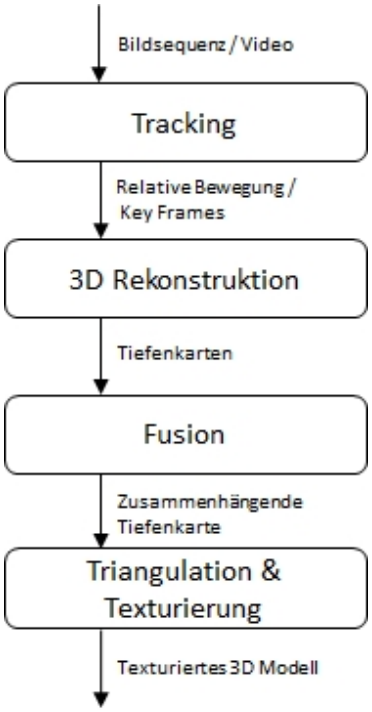

Abbildung 2.9: Verarbeitungskette von *Structure-from-Motion*. Eingabedaten bilden eine Sequenz aus Einzelbilder. Ausgabe ist ein dreidimensionales Modell der Szene.

der Bildstrukturen und der relativen Posen der Bilder kann die Tiefe der Objekte geschätzt werden. Weitere Details zur Schätzung der Tiefe werden im Rahmen dieser Arbeit weiter erläutert. Das Ergebnis der Rekonstruktion ist eine Tiefenkarte der abgebildeten Szene. Diese Tiefenkarte ist dabei an eines der Keyframes gebunden, dem sogenannten Referenzbild.

- **Fusion:** Die einzelnen Tiefenkarten, die bei der 3D-Rekonstruktion erstellt werden, enthalten jeweils die Tiefeninformationen eines bestimmten Ausschnittes der Szene. Dabei können sich diese Ausschnitte beliebig überlappen. Um ein umfassendes dreidimensionales Modell der Szene zu erhalten werden die einzelnen Tiefenkarten im vierten Verarbeitungsschritt, ähnlich wie bei der Erstellung eines Panoramabildes, aneinander geheftet und zu einer globalen Tiefenkarte der Szene fusioniert.

- **Triangulation & Texturierung:** Im letzten Schritt der SfM-Pipeline wird die fusionierte Tiefenkarte in den dreidimensionalen Raum projiziert, trianguliert und gegebenenfalls texturiert. Zunächst wird für jedes Pixel der Tiefenkarte mittels des Strahlensatzes aus den Tiefeninformationen und den intrinsischen Kameraparametern ein dreidimensionaler Punkt erstellt. Die dadurch erhaltene 3D-Punktwolke wird anschließend trianguliert, um ein dichtes Gitter zu erhalten, welches die 3D-Szene darstellt. Bei der Triangulation

wird dabei jeder Punkt mit seinem nächsten Nachbarn verbunden und somit eine Oberfläche aus vielen kleinen Dreiecken erzeugt. Diese dichte Struktur der Szene kann nun noch mit einer Aufnahme der Kamera texturiert werden.

2.3 ABGLEICH VON BILDSTRUKTUREN

Im vorherigen Kapitel wurde erläutert wie aus einem zweidimensionalen Bild ein dreidimensionales Modell entsteht. Ein wichtiger Schritt dabei ist das zuverlässige Finden von Punktkorrespondenzen zwischen einzelnen Aufnahmen. Mittels dieser Korrespondenzen kann die Verschiebung/Parallaxe der einzelnen Objekte zwischen den Bildern ermittelt und damit deren Tiefe geschätzt werden. Hierzu werden die in den Bildern auftretenden Strukturen abgeglichen und einander zuzuordnen. Die Zuordnung erfolgt dabei über das Ermitteln von Kosten für den Vergleich einzelner Pixel. Am Ende wird die Zuordnung zwischen zwei Pixeln ausgewählt, die die geringsten Kosten hat. Für die Berechnung der Kosten gibt es verschiedene Funktionen, die auf den Pixelintensitäten, d. h. Farbe und Helligkeit, basieren. Um der Kürze Willen werden im Folgenden lediglich die *Kostenfunktionen* vorgestellt, die im Rahmen dieser Arbeit verwendet werden.

2.3.1 *Summe absoluter Differenzen*

Eine der schlichtesten und unkompliziertesten Methoden um Bildstrukturen zu vergleichen ist die *Summe absoluter Differenzen (SAD)*. Sie beruht auf die Annahme, dass der mittlere Grauwert, d. h. die Pixelintensitäten, eines Objektes über die verschiedenen Bilder hinweg gleich bleibt. Um somit Objekte wiederzufinden werden die Grauwerte aus dem einen Bild mit denen aus dem anderen Bild verglichen. Die Gleichung

$$d_{x,y}(u,v) = \underset{u,v}{\operatorname{argmin}} \{ \mid f(x,y) - g(x+u, y+v) \mid \} \tag{2.17}$$

beschreibt dabei dieses Vorgehen formal. Für jedes Pixel (x,y) in Bild f wird eine Verschiebung $d(u,v)$ gesucht, wobei u die Verschiebung in x-Richtung und v die Verschiebung in y-Richtung beschreibt, sodass der Grauwert des Pixels $(x+u, y+v)$ in Bild g dem des Pixels aus Bild f am meisten ähnelt. Die Ähnlichkeit wird durch die Minimierung der absoluten Differenz der Intensitäten erzwungen. Dabei beschreiben die Absoluten Differenzen die Abweichung zwischen den beiden Pixeln. Aufgrund der Tatsache, dass die Grauwerte der Pixel kein eindeutiges Merkmal sind, liefert der Pixel-Pixel-Vergleich der Intensitäten jedoch mehrdeutige Ergebnisse. Dies ist in Abbildung 2.10 verdeutlicht, welche die Pixelgrauwerte für zwei 5×5 große Bilder darstellt. Für das Pixel $(1,1)$ aus dem linken Bild mit dem Grauwert 84 soll eine Verschiebung (u,v) zum rechten Bild gefunden werden, sodass Gleichung 2.17 erfüllt wird. Dies würde die Verschiebungen $(-1,2)$ und $(2,1)$ ergeben.

Um diese Mehrdeutigkeiten zu reduzieren werden anstelle von einzelnen Pixeln ganze Pixelblöcke miteinander verglichen. Gemäß

$$d_{x,y}(u,v) = \underset{u,v}{\operatorname{argmin}} \left\{ \sum_{i,j \in N_m} \mid f(x+i, y+j) - g((x+u)+i, (y+v)+j) \mid \right\} \tag{2.18}$$

wird dabei die Summe der absoluten Differenzen der Pixelintensitäten in der Nachbarschaft N_m mit dem Radius m um Pixel (x,y) in Bild f und Pixel $(x+u, y+v)$ in Bild g minimiert.

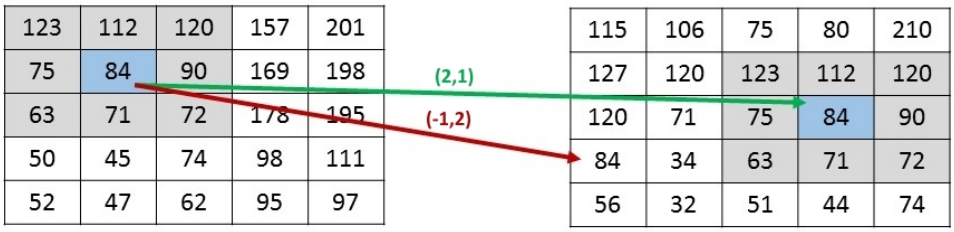

Abbildung 2.10: Mehrdeutigkeiten im Abgleich von Bildstrukturen mittels der *Absoluter Differenzen (AD)*. *Rot:* Fehlerhafte Verschiebung. *Grün:* Korrekte Verschiebung. Falsche Zuordnungen werden durch die Summe der AD über einer lokalen Nachbarschaft reduziert.

Dies ergibt die in Abbildung 2.10 mit grün markierte Verschiebung $(2, 1)$. Dabei sind die entsprechenden Nachbarschaften mit dem Radius 1 mit grau schraffiert.

2.3.2 *Hammingdistanz der Census-Transformation*

Ein großer Nachteil der oben vorgestellten SAD-Methode ist die Annahme, dass der mittlere Grauwert der Objekte von Bild zu Bild gleich bleibt. Diese Annahme ist nur bedingt zutreffend, da sich durch eine Veränderung der Beleuchtungsverhältnisse der Szene auch die Grauwerte der Objekte ändern können. Eine solche Veränderung kann unter anderem durch das Hinzukommen einer weiteren Lichtquelle, Verschattung oder Reflexion hervorgerufen werden.

In [1] haben Zabih und Woodfill eine neue Methode zum Abgleich von Bildstrukturen vorgestellt. Die sogenannte *Census-Transformation (CT)* beruht dabei nicht direkt auf den Grauwerten des Bildes, sondern vielmehr auf der relativen Anordnung der Grauwerte innerhalb einer lokalen Nachbarschaft. Sie projiziert jedes Pixel in eine Bitfolge, die Aufschluss darüber gibt, welcher Pixelgrauwert in der Nachbarschaft N um das Pixel P geringer ist als der Grauwert von P. Abbildung 2.11 zeigt exemplarisch wie eine Nachbarschaft mit dem Radius 1 um das Pixel P gemäß der CT in eine Bitfolge transformiert wird. Beginnend mit dem linken oberen Pixel wird Zeile für Zeile die Intensität jedes Pixels in der Nachbarschaft zu der des zentralen Pixels verglichen. Ist die Intensität kleiner als die des zentralen Pixels, so wird das entsprechende Bit auf 1, ansonsten auf 0 gesetzt.

Der Abgleich zweier, mittels der CT transformierten Bilder, erfolgt dabei über die *Hammingdistanz* h. Diese gibt die Anzahl der sich unterscheidenden Bits zweier Bitfolgen an. Dies berechnet sich beispielsweise wie folgt:

$$h(c_1, c_2) = 4 \quad mit \quad c_1 = 1101000\,, \quad c_2 = 1010001. \tag{2.19}$$

Bei der Korrespondenzsuche werden *die* zwei Pixel einander zugeordnet, bei denen die Hammingdistanz der Bitfolgen am geringsten ist. Analog zu SAD gilt auch hier, dass mehrere Hammingdistanzen innerhalb einer Nachbarschaft aufsummiert werden können, um Mehrdeutigkeiten in der Korrespondenzsuche zu reduzieren.

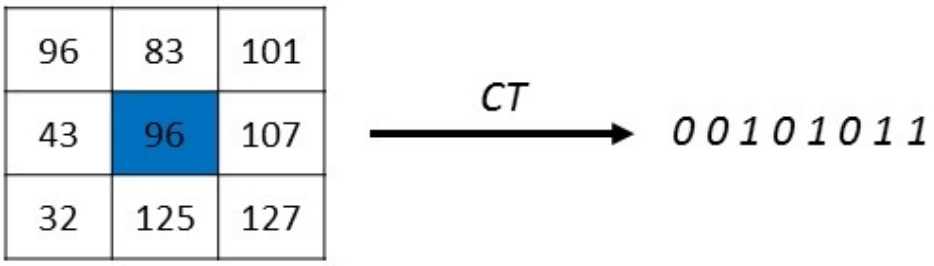

Abbildung 2.11: Exemplarische Umwandlung einer Pixelnachbarschaft in die *Census-Transformation*. Umwandlung der relativen Grauwerte einer Nachbarschaft in eine Bitfolge, beginnend mit dem linken oberen Pixel. Ist der Grauwert des Nachbarpixels geringer als der des zentralen Pixels, wird das entsprechende Bit auf 1 gesetzt.

2.4 MATHEMATISCHE OPERATOREN

Im letzten Abschnitt des Grundlagenkapitels sollen nun noch ein paar Mathematische Operatoren eingeführt werden, die zum Verständnis der Arbeit wichtig sind. Zwar treten diese Operatoren in der Regel im Zusammenhang von Skalar- und Vektorfeldern auf, jedoch kann ein auch ein Bild als ein Feld aus einzelnen Skalaren (Pixelintensitäten) aufgefasst werden. Dies erlaubt das Anwenden der sonst für die Bildauswertung unüblichen Operatoren. Sie helfen dabei, zusätzliche Informationen aus den Aufnahmen zu extrahieren und zu analysieren. In diesem Kapitel werden alle mathematischen Operatoren, die für diese Arbeit von Bedeutung sind, erklärt und deren Anwendung in der Computer Vision erläutert.

2.4.1 *Der Nabla-Operator*

Um einzelne Objekte und Strukturen in einem Bild voneinander unterscheiden zu können, müssen deren Grenzen zueinander erkannt werden. Dabei verändern sich in den meisten Fällen die Intensitäten benachbarter Pixel an diesen Grenzen schlagartig. Grund hierfür ist, dass verschiedene Objekte meist andere Farben oder Strukturen aufweisen. Somit kann davon ausgegangen werden, dass Objektgrenzen dann auftreten, wenn die Unterschiede benachbarter Pixel groß genug sind. Aus Sicht eines Skalarfeldes bedeutet dies, dass das Feld an den entsprechenden Stellen einen hohen Gradienten aufweist. Zur Berechnung dieses Bildgradienten wird der Differenzialoperator *Nabla* ∇ angewendet. Dieser setzt sich aus den partiellen Ableitungen zusammen und ist wie folgt definiert:

$$\nabla f = \operatorname{grad}(f) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)^T, \tag{2.20}$$

wobei $f = f(x, y)$ das Bild beschreibt, und $\frac{\partial f}{\partial x}$ bzw. $\frac{\partial f}{\partial y}$ jeweils für die partiellen Ableitungen in x- bzw. y-Richtung stehen.

Der Nabla-Operator ist auf ein Skalarfeld mit beliebigen Dimensionen n anwendbar. Dieser vektorielle Gradient setzt sich dabei aus n partiellen Ableitungen zusammen. Dadurch wird jedem Wert des Skalarfeldes ein n-dimensionaler Vektor zugeordnet. Bei einer Anwendung auf ein zweidimensionales Skalarfeld $\boldsymbol{f} \in \mathbb{R}^{w \times h}$, wie beispielsweise ein Bild, ergibt sich somit ein zweidimensionales Vektorfeld der Form $\nabla \boldsymbol{f} \in \mathbb{R}^{w \times h \times 2}$. Im kontinuierlichen Fall würden sich die partiellen Ableitungen aus den gewöhnlichen Ableitungen nach den entsprechenden Funktionsvariablen ergeben. Da Bilder jedoch aus diskreten Werten bestehen, müssen die partiellen Ableitungen mit Hilfe der sogenannten finiten Differenzen approximiert werden. Dies Methode wird im Kapitel 2.4.3 näher erläutert.

2.4.2 *Divergenz*

Ein weiterer Differenzialoperator, der Anwendung in der Computer Vision findet, ist die *Divergenz* div. Als Gegenstück zum Nabla-Operator ordnet diese einem Vektorfeld ein Skalarfeld zu. Dieses Skalarfeld gibt Aufschluss über die Struktur des Vektorfeldes um den entsprechenden Punkt. Wird beispielsweise ein Strömungsfeld betrachtet, so gibt die Divergenz das Verhältnis zwischen Abflüsse und Zuflüsse an den jeweiligen Punkten an. Ist die Divergenz positiv, so strömt aus der Umgebung mehr in diesen Punkt hinein als hinaus. Das Feld weist an dieser Stelle eine Quelle auf. Schlussfolgernd bedeutet eine negative Divergenz, dass mehr hinaus fließt als hinein und somit eine Senke vorliegt.

Die Divergenz eine Vektorfeldes ist das vektorielle Produkt zwischen dem Nabla-Operator ∇ und dem Vektorfeld. Damit wird sie aus der Summe der partiellen Ableitungen gebildet und ist für den zweidimensionalen Fall folgendermaßen definiert:

$$\operatorname{div} \boldsymbol{F} = \nabla \cdot \boldsymbol{F} = \frac{\partial \boldsymbol{F}^1}{\partial x} + \frac{\partial \boldsymbol{F}^2}{\partial y} \quad \text{mit} \quad \boldsymbol{F} \in \mathbb{R}^{w \times h \times 2}. \tag{2.21}$$

Auch hier gilt, das die partiellen Ableitungen mittels der finiten Differenzen aus Kapitel 2.4.3 approximiert werden müssen.

2.4.3 *Finite Differenzen*

Bei kontinuierlichen eindimensionalen Funktionen $f(x)$ ergibt sich der Gradient, auch bekannt als Steigung, an einem bestimmten Punkt x_0 durch die Ableitung der Funktion nach x. Die erste Ableitung $f'(x)$ der Funktion ist dabei wie folgt definiert:

$$f'_x(x_0) = \frac{\partial}{\partial x} f(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}. \tag{2.22}$$

In Abbildung 2.12 *(a)* wird die in Gleichung 2.22 gezeigte Berechnung der Ableitung grafisch dargestellt. Darin ist die zu berechnende Tangente an der Stelle x_0 in blau abgebildet. Durch den in Gleichung 2.22 definierten Differenzenquotienten wird die in grün gezeichnete Sekante durch die Punkte x_0 und $(x_0 + h)$ berechnet. Bei differenzierbaren Funktionen kann nun die

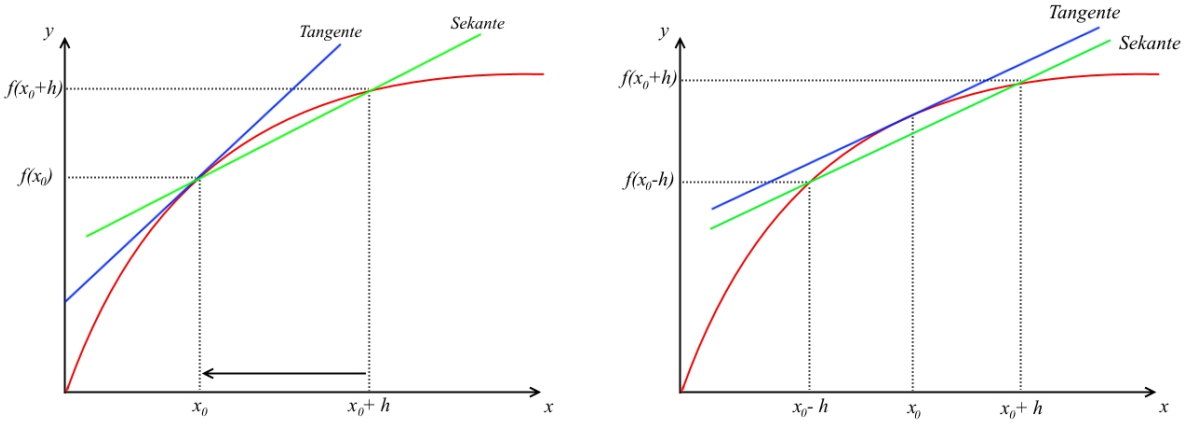

Abbildung 2.12: Von links nach rechts: **(a)** Differenzierung einer Funktion. Berechnung der Sekante durch den Differenzenquotienten zwischen Punkt x_0 und $x_0 + h$. Durch $\lim_{h\to 0}$ wird die Sekante der Tangente angenähert. **(b)** Zentrale Differenzen. Direkte Approximation der Tangente durch Differenzenquotienten zwischen Punkt $x_0 - h$ und $x_0 + h$.

Schrittweite h immer weiter gegen 0 ($\lim_{h\to 0}$) reduziert werden. Dadurch wird die Sekante der Tangente immer weiter angenähert.

Da ein Bild Werte in diskreten Abständen (Pixeln) enthält, ist dieses nicht differenzierbar. Dadurch kann der Gradient nicht direkt durch die erste Ableitung berechnet werden. Ähnlich wie bei der Herleitung der differenziellen Ableitung wird stattdessen die Tangente an der Stelle x_0 mittels einer Sekante durch die zwei benachbarten Punkte $x_0 - h$ und $x_0 + h$ approximiert (vgl. Abb. 2.12 *(b)*). Diese Sekante wird erneut durch einen Differenzenquotienten gebildet. Dieser ist als *zentrale Differenzen* bekannt und wird für den eindimensionalen Fall durch

$$u'_x(x_0) = \frac{u_{x_0+h} - u_{x_0-h}}{2h} \tag{2.23}$$

gebildet. Weitere Variationen der finiten Differenzen sind die *Vorwärts-* und *Rückwärtsdifferenzen*. Sie sind durch

$$u'_x(x_0) = \frac{u_{x_0+h} - u_{x_0}}{h} \tag{2.24}$$

bzw.

$$u'_x(x_0) = \frac{u_{x_0} - u_{x_0-h}}{h} \tag{2.25}$$

definiert. Während sich bei den Vorwärtsdifferenzen die Sekante mittels des Differenzenquotienten durch die Punkte x_0 und $(x_0 + h)$ bildet, werden bei den Rückwärtsdifferenzen die Punkte $(x_0 - h)$ und x_0 verwendet. Analog zur Gleichung 2.22 steht das h in Gleichungen 2.23, 2.24 und 2.25 für die Schrittweite zwischen den, für die Berechnung gewählten, Punkte.

Da für die Berechnung des Gradienten an einem bestimmten Bildpunkt meist die direkten Nachbarpixel gewählt werden, wird in der Regel $h = 1$ gewählt.

Wie auch im kontinuierlichen Fall, müssen bei einem zweidimensionalen Definitionsbereich für den mittels finiten Differenzen ermittelten Gradienten zwei partielle Ableitungen in x- und y-Richtung gebildet werden. Die partiellen Ableitungen basierend auf den zentralen Differenzen mit $h = 1$ sind wie folgt gegeben:

$$\boldsymbol{u}'_x = \frac{\boldsymbol{u}_{x+1,y} - \boldsymbol{u}_{x-1,y}}{2} \quad , \quad \boldsymbol{u}'_y = \frac{\boldsymbol{u}_{x,y+1} - \boldsymbol{u}_{x,y-1}}{2} \; . \tag{2.26}$$

Da der Definitionsbereich Ω eines Bildes begrenzt ist, sind die finiten Differenzen an den Bildrändern nicht definiert. Aus diesem Grund werden in der Berechnung der Differenzenquotienten Randbedingungen angewendet, die es erlauben die finiten Differenzen auch an den Bildrändern zu berechnen. Bekannt aus der Rechnung mit Differentialgleichungen gibt es unter anderem die *Dirichlet-* und *Neumann-Randbedingung*.

Die Dirichlet-Randbedingung ist die einfachste der Randbedingungen. Durch sie werden die Funktionswerte an den Ränder vorgegeben und als 0 definiert. Somit gilt für eine Funktion $\boldsymbol{f}$ mit dem Definitionsbereich Ω, die der Dirichlet-Randbedingung unterliegt,

$$f(0) = f(\Omega) = 0 \; . \tag{2.27}$$

Wird die Dirichlet-Randbedingung auf ein Bild angewendet, so wird dieses mit einem Rahmen der Breite h (Schrittweite der finiten Differenzen) umzogen, welcher mit den Werten 0 gefüllt ist.

Bei der Neumann-Randbedingung werden nicht die Randwerte der Funktion selbst vorgegeben. Stattdessen wird der Wert der vektoriellen Ableitung in Richtung des auf dem Rand stehenden Normalenvektors vorgegeben. Formal ist die Neumann-Randbedingung so definiert:

$$\frac{\partial \boldsymbol{f}}{\partial n} := n \cdot \nabla \boldsymbol{f} = 0, \tag{2.28}$$

wobei n der äußere Normalenvektor am Rand des Definitionsbereiches ist. Der Gradient senkrecht zum Bildrand, und damit in Richtung des Normalenvektors, ist genau dann Null, wenn die Werte außerhalb des Randes gleich den Werten innerhalb des Randes sind. Bei Gleichheit dieser Werte fällt die Differenz im Zähler des Differenzenquotienten weg, wodurch dieser 0 wird. Somit wird ein Bild unter der Neumann-Randbedingung mit einem Rahmen umzogen, der die Werte innerhalb des Bereiches der Breite h am Bildrand spiegelt.

3. VERFAHREN

Nach Erläuterung der Aufgabenstellung und der benötigten Grundlagen können nun die Verfahren vorgestellt werden, die im Rahmen dieser Arbeit zur Rekonstruktion eines dreidimensionalen Modells verwendet werden. Hierfür werden zunächst ein paar Arbeiten genannt und vorgestellt, die sich ebenfalls mit der Thematik beschäftigen. Im Anschluss wird dann zunächst auf das Plane-Sweep-basierte Verfahren eingegangen. Darin wird auch die sogenannte homographische Abbildung hergeleitet. Sie ermöglicht eine direkte Ermittlung von Punktkorrespondenzen zwischen einzelnen Aufnahmen. Nach der Erläuterung des Plane-Sweep-Verfahrens wird der Variations-Ansatz zur 3D-Rekonstruktion erklärt. Dabei wird auch auf die Herkunft und die Stärke von Variations-basierten Methoden eingegangen.

3.1 VERWANDTE ARBEITEN

Die Thematik der 3D-Rekonstruktion genießt in der Computer Vision eine große Relevanz, wodurch die Forschung in diesem Bereich auch eine große Bedeutung hat. Gerade durch die in den letzten Jahren zunehmende Entwicklung von leistungsfähigeren Computern und Kameras wurden die Anwendungsgebiete der 3D-Rekonstruktion immer interessanter und weiträumiger. Dies führte unter anderem zu großen Fortschritten in der Forschung. Im Folgenden wird eine Auswahl an Arbeiten erwähnt, die sich ebenfalls mit der Thematik der 3D-Rekonstruktion mittels Structure-from-Motion befassen. Besonders werden hierbei Plane-Sweep- und Variations-Ansätze berücksichtigt.

Als Teil der Structure-from-Motion (SfM) Verarbeitungskette wird meist ein „*Simultaneous-Localisation-and-Mapping*“-Verfahren (SLAM) dazu verwendet die Kamerabewegung zu schätzen und eine erste Analyse über die Struktur der zu rekonstruierenden Szene durchzuführen. Da SfM lediglich die Daten aus einer Kamera bezieht, können hier nur monokulare SLAM-Verfahren verwendet werden. Solche Ansätze wurden unter anderem von Davison in [2], Klein und Murray in [3], Strasdat *et al.* in [4] und von Engel *et al.* in [5] vorgestellt. Während die ersten drei Verfahren auf den Abgleich von speziellen Deskriptoren, sogenannten „Features“ basieren, werden in [5] direkt die Pixelwerte zum Abgleich verwendet. In allen Verfahren werden die Kameraposen, sowie die Tiefe der einzelnen Punkte in Echtzeit geschätzt.

Ein Plane-Sweep-Verfahren zur 3D-Rekonstruktion mittels mehreren Bildern wurde von Collins in [6] vorgestellt. Seitdem kommt es in zahlreichen Arbeiten zum Einsatz und wurde schon mehrmals erweitert. So zum Beispiel auch von Yang und Pollefeys in [7], die es für eine Verwendung auf einer handelsüblichen Grafikkarte erweitert haben. Damit wurde eine Berechnung in Echtzeit erzielt. Gallup *et al.* haben das Verfahren in [8] auf eine Verwendung mit mehreren Verschiebungsrichtungen und Ebenenorientierungen erweitert. Dabei wird zunächst für jede Orientierung das Optimum gefunden und anschließend aus diesen

Ebenen die kostengünstigste ausgewählt. In [9] wird von Pollefeys *et al.* ein System vorgestellt mit dem eine monokulare Rekonstruktion von Wohngebieten in Echtzeit erfolgt. Dabei werden diese durch eine Kamera, die auf einem Auto montiert ist, aufgezeichnet. Pollefeys *et al.* nutzen eine grobe Szenenanalyse um die Orientierung der Bodenebene und der Häuserfassaden zu ermitteln. Eine sehr aktuelle Abhandlung von Häne *et al.* [10] befasst sich mit der Plane-Sweep-basierten Rekonstruktion in Echtzeit auf Basis von verzerrten Bildern, die durch ein Fischaugen-Objektiv aufgenommen wurden. Dabei wird das Projektionsmodell der Kamera entsprechend angepasst, um weiterhin mittels einer Homographie direkte Punktkorrespondenzen berechnen zu können, ohne die Bilder rektifizieren zu müssen. Eine ebenfalls sehr aktuelle Veröffentlichung von Sinha *et al.* [11] ermöglicht eine effiziente und sehr genaue Rekonstruktion für hochauflösende Bilder. Diese basiert auf einem Plane-Sweep-Verfahren, welches die Ebenen innerhalb kleiner lokalen Bereiche unterschiedlich orientiert und damit den Szenenobjekten präziser anpasst.

Als zweites Verfahren zur 3D-Rekonstruktion soll im Rahmen dieser Arbeit ein Variationsbasiertes Verfahren verwendet werden. Eine Einführung zu Variations-Ansätzen im Zusammenhang der Bildanalyse ist in [12] zu finden. Variations-Ansätze werden in vielen Bereichen der Computer Vision verwendet, so zum Beispiel auch in der Berechnung des optischen Flusses. Erstmals vorgestellt von Horn und Schunk in [13], wird in [14] von Brox *et al.* eine Methode vorgestellt mit der sich der optische Fluss auf Basis eines Variations-Ansatzes sehr genau berechnen lässt. Verschiedene Variations-basierte Methoden zur 3D-Rekonstruktion wurden in [15], [16], [17] und [18] vorgestellt. Gerade auch die Arbeiten von Kuschk *et al.* aus [19], [20] und [21] zur Rekonstruktion mittels Variationsverfahren, sind für diese Arbeit interessant. Darin werden Variations-Verfahren zur Rekonstruktion von urbanen Gebieten basierend auf Luftbildern angewendet. Des Weiteren wurden die Tiefenkarten in [19] mittels dem Verallgemeinerten Variations-Ansatzes aufgebaut. Dieser begünstigt nicht nur stückweise konstante Funktionen, sondern stückweise affine Funktionen, wodurch die schrägen Oberflächen der Hausdächer besser rekonstruiert werden. Weitere Abhandlungen , die Verfahren oder Systeme im Zusammenhang von 3D-Rekonstruktionen aus Luftbildern vorstellen, sind die von Scaramuzza *et al.* [22] oder von Weiss *et al.* [23].

Neben den Methoden, die auf einem Plane-Sweep- oder Variations-Verfahren basieren, sind zahlreiche andere Vorgehen zur echtzeitnahen 3D-Rekonstruktion vorhanden. So zum Beispiel das Verfahren von Newcombe *et al.* [24]. Dabei wird zunächst ein grobes Modell basierend auf ein paar wenige Punktkorrespondenzen aufgebaut. Daraufhin wird das Basismodell sukzessiv immer weiter trianguliert und verfeinert. Eine weitere Vorgehensweise ist der probabilistische Aufbau einer Tiefenkarte, welcher unter anderem in [25] und [26] verwendet wird. Dabei wird eine probabilistische Tiefenschätzung für jedes Pixel durchgeführt um dadurch die wahrscheinlichste Tiefenkarte zu erhalten.

In der Verarbeitungskette der SfM Methode folgt der 3D-Rekonstruktion die Fusion der einzelnen Tiefenkarten. Merrell *et al.* stellen in [27] eine echtzeitfähige direkte Fusion der Tiefenkarten vor. Dabei werden Tiefeninformationen einzelner Pixel verglichen und überprüft, ob Verletzungen gegen die Sichtbarkeitsbedingungen auftreten. Dadurch werden Fehler

und überflüssige Tiefenschätzungen verworfen. Auch in der Fusion der Tiefenkarten können Variations-Ansätze verwendet werden. So zum Beispiel von Pock *et al.* in [28]. Darin werden Tiefenkarten von Gebäuden mittels des Verallgemeinerten Variations-Ansatzes fusioniert, welcher die geneigten Oberflächen der Gebäude präziser rekonstruiert.

Für die Umsetzung im Rahmen dieser Arbeit wird zunächst die Erweiterung des Plane-Sweep-Verfahrens auf mehrere Ebenenorientierungen von Gallup *et al.* [8] sowie der Variations-Ansatz [19] von Kuschk und Cremers verwendet. Diese Auswahl basiert zum einen auf der bereits erfolgreichen Anwendung des Variations-Ansatzes auf Luftbildern sowie auf der Flexibilität, die das erweiterte Plane-Sweep-Verfahren in der Anpassung an verschiedene Ebenen bietet. Unter Berücksichtigung, dass Luftaufnahmen einer Szene meist aus verschiedenen planaren Bereichen aufgebaut sind, bietet der Plane-Sweep-Ansatz mit verschiedenen Ebenenorientierungen eine gute Möglichkeit diese an die Objekte anzupassen. Zudem können Plane-Sweep-Verfahren äußerst effizient auf handelsüblichen Grafikkarten umgesetzt werden. Im Folgenden wird die Vorgehensweise der beiden Verfahren näher erläutert.

3.2 DAS PLANE-SWEEP-VERFAHREN

Wie der englische Begriff *Plane-Sweep* bereits andeutet wird bei einem solchen Verfahren ein Modell mittels der Verschiebung einer Ebene durch den dreidimensionalen Raum ($\mathbb{R}^3$) erstellt. Ein wichtiges mathematisches Hilfsmittel ist dabei die projektive Transformation, auch bekannt als Homographie. Allgemein beschreibt diese die Abbildung von Punkten und Geraden aus *einem* Projektiven Raum in einen anderen. In der Computer Vision wird die Homographie dazu verwendet, Bilder aus dem Koordinatensystem *einer* Kamera in das einer anderen zu transformieren. Im Folgenden wird zunächst die Homographie näher erläutert und die benötigte Gleichung hergeleitet. Daraufhin wird auf die Vorgehensweise des Plane-Sweep-Verfahren eingegangen und gezeigt, wie damit ein 3D-Modell berechnet werden kann.

3.2.1 *Ebenen induzierte Homographie*

Die im Kapitel 2.2.2 vorgestellte Epipolargeometrie bestimmt aus einem Punkt m_i in einem Bild eine Epipolarlinie l_i im zweiten Bild. Diese ist bekanntlich die Projektion des Sichtstrahls der ersten Kamera durch m_i auf die Bildebene der zweiten Kamera und stellt den möglichen Aufenthaltsort der Punktkorrespondenz m_i' zu m_i dar. Der Grund warum lediglich eine Linie aus einem Punkt bestimmt werden kann, ist die fehlende Information wo im Raum der Sichtstrahl durch m_i das Szenenobjekt schneidet. Ist dieser Schnittpunkt bekannt lässt sich aus dem Punkt m_i direkt ein anderer Punkt m_i' bestimmen. Diese direkte Beziehung zwischen zwei Punktkorrespondenzen wird als *Homographie* bezeichnet, die durch eine Ebene Π im $\mathbb{R}^3$ induziert wird. Sie transformiert Punkte aus der einen in die andere Kamera unter der Annahme, dass die Punkte auf der gegebenen Raumebene liegen (vgl. Abb. 3.1).

Zur Berechnung der Homographie werden die relativen Positionen der Kameras zueinander, sowie der zur Ebene orthogonal stehende Normalenvektor und der Abstand der Ebene zum Zentrum des Koordinatensystems benötigt. Die Herleitung der Gleichung für die homographische Abbildung basiert auf dem Beweis von Hartley und Zissermann in [29]:

Unter der Annahme, dass das Weltkoordinatensystem im Brennpunkt der ersten Kamera zentriert ist, lassen sich folgende Projektionsmatrizen P_i für die beiden Kameras aufstellen:

$$P_1 = K_1 \cdot [\, I \mid 0 \,] \quad , \quad P_2 = K_2 \cdot \left[\, R \mid \vec{t} \,\right]. \tag{3.1}$$

Hierbei entspricht die Matrix K_i der 3×3 großen Kalibrierungsmatrix der jeweiligen Kamera, die Matrix I einer Einheitsmatrix der Größe 3×3, die Matrix R der 3×3 relativen Rotationsmatrix zwischen dem Koordinatensystem der ersten und zweiten Kamera, und der Vektor $\vec{t}$ dem dreidimensionalen Translationsvektor der die relative Verschiebung zwischen den beiden Kameras beschreibt.

Ferner sei die fiktive Ebene Π durch $\pi^T \cdot \widetilde{M}_p = 0$ mit $\pi = (\vec{n}^T, d)^T$ definiert. Hierbei beschreibt $\vec{n}$ den dreidimensionalen Normalenvektor der Ebene und d den Abstand der Ebene zum Zentrum des Koordinatensystems, welches im Brennpunkt der ersten Kamera liegt. Die

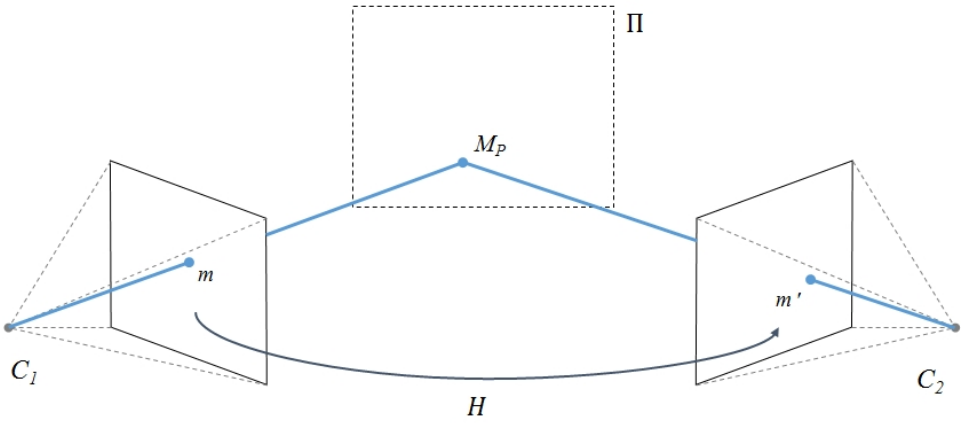

Abbildung 3.1: Ebenen induzierte Homographie. Der Punkt m_i aus Kamera C_1 wird über eine Ebene auf den Punkt m_i' in Kamera C_2 abgebildet. Dazu wird der Punkt zunächst auf die Ebene Π in den Punkt M_P Rückprojiziert. Anschließend wird der Punkt M_P in der Kamera C_2 auf den Punkt m_i' abgebildet.

Herleitung der Homographie H, welche $\widetilde{m}' = H \cdot \widetilde{m}$ erfüllt, erfolgt nach folgenden Schritten (vgl. Abb. 3.1):

---

1. Rückprojektion des Punktes m_i aus der ersten Kamera, was in einem Sichtstrahl V durch m_i resultiert.
2. Ermittlung des Schnittpunktes M_P zwischen dem Sichtstrahl V und der Ebene Π.
3. Die anschließende Projektion des Punktes M_P in die zweite Kamera liefert m_i'.

Aus dem Kapitel 2.1.2 zur projektiven Geometrie ist bekannt, dass für m in der ersten Kamera

$$\widetilde{m}_i = P_1 \cdot \widetilde{M}_W = \overbrace{K_1 \cdot [\, I \mid 0 \,]}^{3\times 4} \cdot \widetilde{M}_W \tag{3.2}$$

gelten muss. Hierbei beschreibt M_W den Szenenpunkt im Weltkoordinatensystem, welches in diesem Beispiel im Brennpunkt der ersten Kamera zentriert ist. Die Tilde (~) gibt bekanntlich an, dass die Punkte in homogenen Koordinaten gegeben sind. Die Rückprojektion gemäß

$$V = P_1^{-1} \cdot \widetilde{x}_1 = \overbrace{K_1^{-1} \cdot \begin{bmatrix} I \\ 0 \end{bmatrix}}^{4\times 3} \cdot \widetilde{m}_i = \begin{pmatrix} \vec{v} \\ \rho \end{pmatrix}, \tag{3.3}$$

mit $\vec{v} = K_1^{-1} \cdot \widetilde{m}_i$, liefert einen Sichtstrahl $V = (\vec{v}^T, \rho)^T$. Gemäß der Perspektivischen Projektion liegen alle Punkte im dreidimensionalen Raum, die auf den Punkt m_i abgebildet werden, auf diesem Sichtstahl V. Der Ort der jeweiligen Szenenpunkte auf V wird dabei durch ρ parametrisiert.

Da ein Punkt auf der Ebene Π gesucht wird, muss dieser die Gleichung $\pi^T \cdot V = 0$ erfüllen. Durch Einsetzen folgt

$$\pi^T \cdot V = \begin{pmatrix} n_1 \\ n_2 \\ n3 \\ d \end{pmatrix}^T \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \rho \end{pmatrix} = 0. \tag{3.4}$$

Dies kann nach $\rho = (-\vec{n}^T \cdot \vec{v})/d$ aufgelöst werden. Dadurch ergibt sich für den Schnittpunkt des Sichtstrahls V mit der Ebene Π: $\widetilde{M}_P = (\vec{v}^T, (-\vec{n}^T \cdot \vec{v})/d)^T$.

Durch die Projektion des Punktes $\widetilde{M}_P$ in die zweite Kamera lässt sich $\widetilde{m}_i'$ gemäß

$$\begin{aligned} \widetilde{m}_i' &= P_2 \cdot \widetilde{M}_P = K_2 \cdot \left[R \mid \vec{t} \right] \cdot \begin{pmatrix} \vec{v} \\ \frac{-\vec{n}^T \cdot \vec{v}}{d} \end{pmatrix} = K_2 \cdot \left[R\vec{v} - \frac{\vec{t} \cdot \vec{n}^T}{d} \vec{v} \right] \\ &= K_2 \cdot \left(R - \frac{\vec{t} \cdot \vec{n}^T}{d} \right) \cdot \vec{v} \end{aligned} \tag{3.5}$$

ermitteln. Ein letztes Einsetzen von $\vec{v} = K_1^{-1} \cdot \widetilde{m}_i$ ergibt

$$\widetilde{m}_i' = \underbrace{K_2 \cdot \left(R - \frac{\vec{t} \cdot \vec{n}^T}{d} \right) \cdot K_1^{-1}}_{\textit{Homographie } H} \cdot \widetilde{m}_i. \tag{3.6}$$

Dies entspricht der gesuchten Gleichung der Form $\widetilde{m}_i' = H \cdot \widetilde{m}_i$. Die endgültige und relevante Formel für die Homographie zwischen zwei Bildern, welche durch eine Ebene mit dem Normalenvektor $\vec{n}$ und dem Abstand d vom Zentrum des Koordinatensystems induziert wird, ist wie folgt:

$$H = K \cdot \left(R - \frac{\vec{t} \cdot \vec{n}^T}{d} \right) \cdot K^{-1}. \tag{3.7}$$

Dabei geben R und $\vec{t}$ die relative Bewegung der zwei Bilder bzw. Aufnahmeorte zueinander an. Da sich diese Arbeit ausschließlich mit der Monokularen Rekonstruktion beschäftigt, und die Bilder somit nur aus einer Kamera bezogen werden, sind die beiden Kalibrierungsmatrizen K_1 und K_2 aus Gleichung 3.6 identisch. Sie werden lediglich als K gekennzeichnet. Diese 3×3 große Matrix H ermöglicht nun die direkte Ermittlung einer Punktkorrespondenz zu Punkt m_i unter der Annahme, dass der dazugehörige Szenenpunkt auf der entsprechenden Ebene

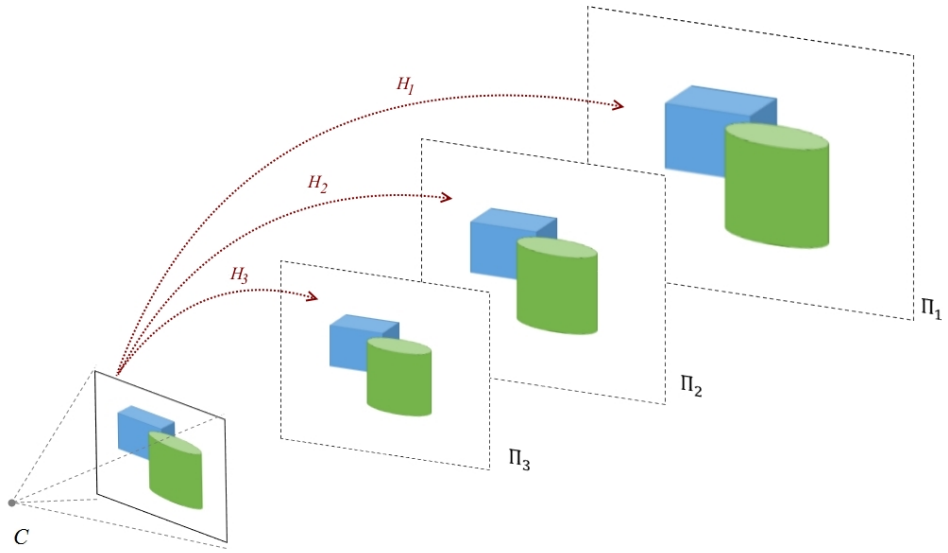

Abbildung 3.2: Projektion der Aufnahme aus Kamera C, auf verschiedene Ebenen im $\mathbb{R}^3$ mittels der Homographie. Dabei wird für jede Projektion eine andere Homographie benötigt.

im $\mathbb{R}^3$ liegt. An diesem Punkt sollte erwähnt werden, dass dabei keines der optischen Zentren der beiden Kameras auf dieser Ebene liegen darf.

Aus Gleichung 3.7 geht ebenfalls hervor, dass sich für $\lim_{d\to\infty} H = K \cdot R \cdot K^{-1} = H_\infty$ eine besondere Homographie ergibt. Eine Abbildung, die durch eine Ebene im Unendlichen induziert wird. Diese sogenannte *unendliche Homographie* ist ausschließlich von der relativen Rotation zwischen den beiden Kameras abhängig. Sie wird ebenfalls erhalten, wenn keine Translation zwischen den Aufnahmen vorliegt, welches einer alleinigen Rotation um den Brennpunkt der Kamera entspricht. Dies bedeutet zugleich, dass mittels der unendlichen Homographie keine Tiefenschätzung möglich ist, da eine Rotation der Kamera alleine keine Parallaxe zwischen verschiedenen Aufnahmen hervorruft.

3.2.2 *Ebenen-spezifisches Abtasten*

Durch die Anwendung der im vorherigen Kapitel vorgestellten homographischen Abbildung H auf alle Punkte eines Bildes kann dieses von der Bildebene, auf der es aufgenommen wurde, auf eine andere Ebene im $\mathbb{R}^3$ projiziert werden (vgl. Abb. 3.2). Aus dieser Ebene im dreidimensionalen Raum, kann die Aufnahme wieder auf der Bildebene einer anderen Kamera abgebildet werden. Somit können mittels der Homographie Aufnahmen aus einer Kamera über eine Raumebene in eine andere Kamera transformiert werden. Dies ermöglicht das Überlagern der Aufnahmen und einen direkten Vergleich verschiedener Blickwinkel der Szene. Damit kann direkt nach Punktkorrespondenzen in den verschieden Aufnahmen gesucht werden ohne zuerst Epipolarlinien bestimmen zu müssen. Die Transformationen und damit auch die jeweiligen

Bildvergleiche sind dabei bekanntlich von der Parametrisierung der Ebene abhängig. Es ergibt sich somit für jede Orientierung und jede Positionierung der Ebene im $\mathbb{R}^3$ eine andere Transformation.

Um mittels des Plane-Sweep-Verfahrens die Tiefeninformation der Szene zu rekonstruieren, wird die Szene mit zahlreichen Ebenen verschiedener Parametrisierung abgetastet. Für jede dieser Ebenen wird dabei die Aufnahme aus der einen Kamera, mittels der entsprechenden homographischen Abbildung H, auf die Ebene und dann in die andere Kamera transformiert. Durch die Konstruktion der Homographie wird dabei impliziert, dass Bildpunkte, deren dazugehöriger Szenenpunkt auf der entsprechenden Raumebene liegt, bei der Überlagerung des transformierten und nicht-transformierten Bildes deckungsgleich sein müssen. Somit wird für jedes Pixel eine Ebene im $\mathbb{R}^3$ gesucht, die dem entsprechenden Szenenpunkt am nächsten liegt. Die Ebenen werden je nach Orientierung in Klassen, sogenannten *Ebenenfamilien*, gruppiert. Das heißt, dass alle Ebenen einer Klasse denselben Normalenvektor $\vec{n}$ haben und sich lediglich in der Entfernung d zum Ursprung unterscheiden. Für jede Ebenenfamilie ist ein Normalen-Vektor $\vec{n}$ sowie Entfernungsminimum d_{min} und -maximum d_{max} der Ebenen gegeben.

Das Vorgehen des Ebenen-spezifischen Abtastens wird im Folgenden zunächst für eine Ebenenfamilie erläutert, und anschließend auf eine beliebige Anzahl von Normalenvektoren erweitert.

---

1. Beginnend bei der maximalen Entfernung wird die Ebene entlang ihres Normalenvektors $\vec{n}$ in bestimmten Abständen durch den Raum verschoben. Solange bis diese die minimale Entfernung erreicht.

2. Für jede Ebenenposition wird die Aufnahme der Referenzkamera mittels der homographischen Abbildung H (vgl. Gl. 3.7) über die Ebene Π_k in die zweite Kamera projiziert und mit deren Aufnahme überlagert.

3. Die beiden Bilder werden nun mit einander verglichen. Dies erfolgt durch die Berechnung von pixelbasierten Kosten zwischen den überlagerten Aufnahmen gemäß einer beliebigen Kostenfunktion (siehe Kapitel 2.3). Die Kosten werden in einem sogen Kostenvolumen C_{vol} der Größe $w \times h \times m$[1] abgespeichert. Hierbei enthalten die einzelnen Schichten der dritten Dimension des Kostenvolumens die Kosten der jeweiligen Ebenen Π_k.

1 w = Bildbreite, h = Bildhöhe, m = Anzahl der Ebenen

4. Anschließend wird für jedes Pixel *die* Ebene ausgewählt, die die geringsten Kosten aufweist. Dabei werden die pixelbasierten Minimalkosten des Kostenvolumens gemäß

$$\widetilde{C}_{x,y} = \min_{k} \{C_{vol}(x, y, k)\} \tag{3.8}$$

ermittelt und dem entsprechenden Pixel die dazugehörige Ebene

$$\widetilde{\Pi}_{x,y} = \operatorname*{argmin}_{k} \{C_{vol}(x, y, k)\} \tag{3.9}$$

zugeordnet.

5. Zuletzt wird die Tiefenkarte aufgebaut. Dabei wird der Sichtstrahl durch das jeweilige Pixel mit der entsprechend ausgewählten Ebene $\widetilde{\Pi}_{x,y}$ geschnitten (vgl. Gl. 3.4). Die entsprechende Tiefe ergibt sich dann aus den Koordinaten des Schnittpunktes. Bei einer frontoparallelen Orientierung der Ebenen entspricht die Tiefe dem Abstand der Ebene zum optischen Zentrum der Referenzkamera.

In ihren Abhandlungen [8] und [9] haben Gallup *et al.* sowie Pollefeys *et al.* Plane-Sweep-Verfahren vorgestellt, die die Ebenen in verschiedenen Richtungen und Orientierungen durch den Raum bewegen und somit das grundlegende Verfahren für die Verwendung mehrerer Ebenen-Familien erweitert. Diese Erweiterung ist dabei eingängig und lässt sich auf eine beliebige Anzahl von Orientierungen bzw. Verschiebungsrichtungen anwenden:

---

1. Zunächst wird für jede Familie von Ebenen, die durch den Normalenvektor $\vec{n}$, sowie dem Entfernungsminimum d_{min} und -maximum d_{max} klassifiziert sind, gemäß der oben vorgestellten Methode pixelbasiert die kostengünstigste Ebene $\widetilde{\Pi}^{n}_{x,y}$ ermittelt.

2. Aus dieser Gruppe von optimalen Ebenen wird anschließend für jedes Pixel die Ebene bestimmt die insgesamt die geringsten Kosten hat. Diese stellt die sogenannte „Winner-Takes-It-All“ Lösung $\hat{\Pi}_{x,y}$ dar.

3. Auch hier werden die Parametrisierungen der Ebene $\hat{\Pi}_{x,y}$ abschließend dazu verwendet die Tiefenkarte zu bestimmen.

Auf die in [8] zusätzlich durchgeführte energiebasierte Optimierung zur Reduzierung von Ausreißern und zur Minimierung von sprunghaften Änderungen wird im Rahmen dieser Arbeit verzichtet, da das Plane-Sweep-Verfahren ohnehin mit einer *Total-Variations-Optimierung* (vgl. Kapitel 3.3) in Verbindung gesetzt wird. Mehr zu dem umgesetzten Verfahren ist in 4.1 zu finden.

3.2.3 *Verwendung mehrerer Aufnahmen*

Die Erläuterung der Vorgehensweise des Plane-Sweep-Verfahrens basiert in den vorherigen Kapiteln auf einem stereoskopischen Aufbau und berücksichtigte daher lediglich zwei Aufnahmen einer Szene. Die Verwendung eines SfM-Ansatzes zur Rekonstruktion einer Szene erlaubt es aber mehr als zwei Bilder für die Tiefenschätzung in Betracht zu ziehen. Die erhöhte Anzahl an betrachteten Aufnahmen führt dabei zu einer Qualitätssteigerung des Modells, da die Positionen der Ebenen durch eine größere Anzahl an Transformationen und Vergleiche getestet und validiert werden. Die Erweiterung des Plane-Sweep-Verfahren auf eine Mehrzahl an Aufnahmen ist ebenso eingängig wie die Erweiterung auf mehrere Ebenenfamilien.

Zunächst wird die mittlere der vorhandenen Aufnahmen als Referenzaufnahme ausgewählt und das Koordinatensystem im Brennpunkt der dazugehörigen Kameraposition zentriert. Die darauffolgende Abtastung der Szene durch die verschiedenen Ebenen und Ebenenfamilien verläuft zum größten Teil analog zum stereoskopischen Fall. Lediglich das Vorgehen bei der Transformation der Aufnahmen und der Berechnung der Kosten für die einzelnen Ebenen verläuft anders.

Anstelle einer einzelnen Bildprojektion, müssen nun pro Ebene $k-1$ verschiedene Homographien berechnet und Aufnahmen projiziert werden. Dabei ist k die Anzahl der in Betracht gezogenen Bildern. Pro Ebene werden alle umliegenden Aufnahmen I_k transformiert und mit dem Referenzbild I_{ref} überlagert. Ähnlich wie bei der Verwendung von zwei Aufnahmen werden die pixelbasierten Kosten für die Ebene gemäß einer gegebenen Kostenfunktion bestimmt. Dabei ergeben sich die Gesamtkosten für die Ebene aus der Summe der Teilkosten, die aus den einzelnen Transformationen entstehen:

$$Cost_{total} = \sum_{k \setminus \{ref\}} Cost(I_{ref}, I_k, H_{I_{ref}, I_k}). \tag{3.10}$$

Die Auswahl der bestmöglichen Ebene für jedes Pixel erfolgt dann erneut über die Suche nach dem pixelbasierten Kostenminimum innerhalb des Kostenvolumens.

3.3 DAS VERALLGEMEINERTE-VARIATIONS-VERFAHREN

Mittels des vorgestellten Plane-Sweep-Verfahrens lässt sich ein dreidimensionales Modell einer Szene aus einer Sequenz aus Einzelbildern mit dazugehörigen Kameraposen berechnen. Die Qualität des dabei entstehenden Modells hängt jedoch stark von den Eingangsdaten ab. Je besser die Qualität der Einzelbilder, sowie der Genauigkeit der Kameraposen und der damit verbundenen Homographie, je zuverlässiger können Strukturen und Objekte zwischen den Aufnahmen einander zugeordnet und damit die Tiefe bestimmt werden. Sinkt die Qualität der Eingangsdaten, so kann es häufiger zu falschen Zuordnungen und Fehlern im Modell kommen. Nicht nur die Fehlberechnungen, sondern auch die Unstetigkeit, die mit dem Plane-Sweep-Verfahren einhergeht, kann starke Schwankungen und häufige Diskontinuitäten im resultierenden Modell hervorrufen. Das im Folgenden beschriebene *Verallgemeinerte-Variations*-Verfahren (TGV), aus dem Englischen *Total-Generalized-Variation*, setzt eine Optimierung mit Nebenbedingungen um und erlaubt es somit, dem zu berechnenden Modell zusätzliche Einschränkungen wie beispielsweise Stetigkeit aufzuerlegen.

3.3.1 *Total-Variation*

Die Mathematik unterscheidet zwischen *direkten* und *inversen* Problemstellungen. Dabei wird eine Aufgabenstellung als direktes Problem bzw. Vorwärtsproblem bezeichnet, wenn aus gegebener Ursache die Wirkung eines Systems ermittelt werden soll. Betrachtet man die Aufnahme einer Szene durch eine Kamera, so wäre die perspektivische Projektion der Szene in ein zweidimensionales Bild, unter Kenntnis der Szenen- und der Kamerageometrie ein solches Vorwärtsproblem. Denn aus den Koordinaten der Szenenpunkte und den intrinsischen Kameraparameter lässt sich problemlos die zweidimensionale Abbildung berechnen (vgl. Kapitel 2.1.2). Die Rücktransformation aus der Aufnahme in eine dreidimensionale Szene hingegen ist charakteristisch für inverse Problemstellungen. Dabei soll aus einer bekannten Wirkung eines Systems auf die Ursache geschlossen werden. In der 3D-Rekonstruktion lässt sich trotz Kenntnis über die Bildinformation und der Kameraparametrisierung lediglich ein möglicher Aufenthaltsort des Szenenpunktes bestimmen und damit kein eindeutiges Ergebnis erzielen. Inverse Probleme sind meist sehr schwierig oder gar nicht lösbar. Mehr zum Umgang mit dieser Art Problemstellungen ist in [30] zu finden.

Gerade in der Computer Vision werden solche *inverse* Problemstellungen oft in ein Minimierungsproblem einer globalen Energiefunktion transformiert. Die Energiefunktion setzt sich dabei aus der Summe eines *Daten-* und *Regularisierungsterms* zusammen. Als Vorreiter gilt hierbei das von Rudin *et al.* vorgestellte *ROF-Rauschreduzierungs*-Verfahren [31]. Dies wendet zum ersten Mal einen *Total-Variations*-basierten (TV) Ansatz zur Rauschreduzierung einer Aufnahme an. Der Begriff Total-Variation bezeichnet die Summe der Beträge aller Bildgradienten, welche im Falle eines verrauschten Bildes durch die starken Sprünge sehr hoch ist. Bei der ROF-Rauschreduzierung wird diese TV in Abhängigkeit einer Nebenbedingung minimiert. Diese Nebenbedingung erzwingt dabei, dass das gesuchte, verbesserte Bild dem gegebenen,

Abbildung 3.3: Von links nach rechts: **(a)** Verrauschte Aufnahme. **(b)** Rekonstruierte Aufnahme mittels ROF-Rauschreduzierung. Deutliche Glättung des Bildes unter Erhalt der Bildstrukturen. Quelle: http://www.mathworks.com/examples/matlab/3636-total-variation-denoising (Zugriff: 29.03.2015)

verrauschten Bild ähnelt und somit die Bildstrukturen erhält. Die formale Beschreibung des ROF-Denoising-Verfahrens ist wie folgt gegeben:

$$\boldsymbol{u} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \left\{ \underbrace{|\nabla \boldsymbol{u}|}_{R(\boldsymbol{u})} + \lambda \underbrace{|\boldsymbol{f} - \boldsymbol{u}|}_{C(\boldsymbol{u})} \right\} \tag{3.11}$$

Hierbei ist $\boldsymbol{u} = u(x, y)$ das gesucht, verbesserte Bild und $\boldsymbol{f} = f(x, y)$ das vorhandene, verrauschte Bild. Der Regularisierungsterm, welcher die Reduzierung der Total-Variation und damit Rauschreduzierung bewirkt, ist mit $R(\boldsymbol{u})$ gekennzeichnet. Durch ihn werden die Beträge der Bildgradienten minimiert. Die Nebenbedingung ist durch den Datenterm oder Kostenterm $C(\boldsymbol{u})$ gegeben. Er minimiert die Differenz zwischen dem verrauschten und dem verbesserten Bild und erzwingt damit deren Ähnlichkeit. Mit dem zusätzlichen Gewichtungsterm λ kann der Einfluss der beiden Terme reguliert werden.

In Abbildung 3.3 ist das Ergebnis einer durchgeführten Rauschreduzierung mittels dem ROF-Verfahren abgebildet. Dabei zeigt *(a)* die verrauschte Aufnahme und *(b)* das rekonstruierte Bild. Darin ist deutlich die Glättung des Bildes unter Erhalt der Strukturen zu erkennen.

3.3.2 *Verallgemeinerte-Total-Variation zweiter Ordnung*

Eine große Einschränkung der TV-basierten Formulierung eines inversen Problems ist, dass der Regularisierungsterm $R(\boldsymbol{u})$ lediglich auf der ersten Ableitung von $\boldsymbol{u}$ basiert. Dies impliziert die Annahme, dass das gesuchte Bild nur aus flachen, frontoparallelen Strukturen aufgebaut ist. Die Optimierung nach der ersten Ableitung begünstigt damit stückweise konstante

Funktionen, was bei abgeschrägten Strukturen zu sogenannten „Staircasing"-Artefakten führt. Solche Artefakte sind auch in Abbildung 3.3 *(b)* zu erkennen. Farbverläufe werden durch eine Treppenfunktion rekonstruiert, wodurch deutliche Abstufungen in den entsprechenden auftreten.

Die von Bredies *et al.* vorgestellte *Total-Generalized-Variation (TGV)* [32] verallgemeinert die Total-Variation und erlaubt bei der Regularisierung die Berücksichtigung von Ableitungen höheren Grades. Anders als bei der TV wird das rekonstruierte Bild dabei nicht nur aus abschnittsweise konstanten Funktionen zusammengesetzt, sondern aus abschnittsweise Polynomfunktionen. Dies erlaubt eine genauere Rekonstruktion der im Bild befindlichen Strukturen. Je nach Grad k der angewendeten TGV werden dabei Funktionen des Grades $k-1$ den lokalen Strukturen angepasst. Die TGV erster Ordnung (TGV^1) entspricht der einfachen TV. Sie passt Funktionen nullter Ordnung, d.h. konstante Funktionen, der Bildstrukturen an. Wird beispielsweise die TGV zweiter Ordnung (TGV^2) zur Rauschreduzierung angewendet, so können Farbverläufe durch lineare Funktionen approximiert werden. Dies entfernt was die „Staircasing"-Artefakte in Bereichen von Farbverläufen (vgl. Abb. 3.4 *(c)*). Bei der Verwendung eines zu hohen Grades der *TGV* kann es jedoch passieren, dass die rekonstruierte Funktion an das Rauschen angepasst wird, was nicht zu einer Rauschreduzierung führen würde. Die formale Definition und der mathematische Beweis, dass TGV^1 der einfachen TV entspricht und damit die TGV formal eine Verallgemeinerung der TV darstellt, ist in der entsprechenden Abhandlung [32] gegeben. Diese Arbeit wird sich lediglich mit der TGV^2-basierten Regularisierung befassen, welche im Folgenden näher erläutert wird. Da die TGV eine Verallgemeinerung der TV darstellt, wird zudem im Folgenden statt *TV* der Ausdruck TGV^1 verwendet.

Der TGV^2-basierte Regularisierungsterm ist durch

$$R(\boldsymbol{u})_{TGV^2} := \alpha_1 \underbrace{|\nabla \boldsymbol{u} - \boldsymbol{v}|}_{R_1} + \alpha_2 \underbrace{|\nabla \boldsymbol{v}|}_{R_2} \tag{3.12}$$

definiert. Bei der Regularisierung durch TGV^2 werden nicht nur allein die Schwankungen in $\boldsymbol{u}$ minimiert, sondern auch die eines zusätzlichen Vektorfeldes $\boldsymbol{v}$. In Term R_1 wird dieses zusätzliche Vektorfeld vom Gradienten von $\boldsymbol{u}$ abgezogen und der Betrag dieser Differenz minimiert. Ähnlich wie bei der Kostenfunktion $C(\boldsymbol{u})$ in Gleichung 3.11 wird dadurch eine Ähnlichkeit des Vektorfeldes $\boldsymbol{v}$ zu $\nabla \boldsymbol{u}$ erzwungen. Der Term R_2 sorgt dafür, dass die Variation in diesem zusätzlichen Vektorfeld ebenfalls minimiert wird. Durch die Kombination aus R_1 und R_2 werden lineare Funktionen begünstigt. Denn lineare Funktionen der Form $y = px + c$ haben konstante Gradienten p, deren Ableitung gleich 0 ist. Die Steigung der affinen Funktionen wird dabei durch R_1 an die Gradienten der lokalen Strukturen von $\boldsymbol{u}$ angepasst. Die Nullwertigkeit der Ableitungen wird durch R_2 erzwungen. Die zusätzlichen Multiplikatoren α_i gewichten dabei den Einfluss der beiden Terme R_1 und R_2.

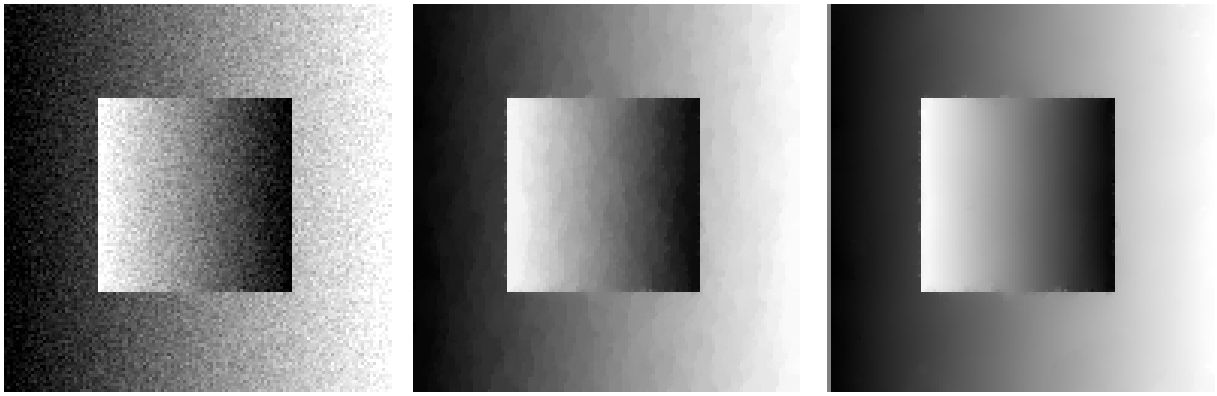

Abbildung 3.4: Von links nach rechts: **(a)** Verrauschtes Bild mit Farbverläufen zur Demonstration der Unterschiede zwischen *TV* und *TGV*. **(b)** Rekonstruktion mittels *TV*-basierter Rauschreduzierung. Deutliche „Staircasing"-Artefakte in Bereichen von Farbverläufen zu erkennen. **(c)** Rekonstruktion mittels *TGV*-basierter Rauschreduzierung. Fehlerfreie Rekonstruktion der Farbverläufe. Quelle: http://www.uni-graz.at/imawww/optcon/projects/bredies/tgv.html (Zugriff: 29.03.2015) [32]

3.3.3 *TGV²-gestützte 3D-Rekonstruktion*

In den beiden vorhergehenden Kapiteln wird die Verallgemeinerte-Total-Variation erster und zweiter Ordnung im Rahmen der Rauschreduzierung von Bildern vorgestellt. Da es sich bei der TGV um Ansätze zur Regularisierung handelt, lassen diese sich auf beliebige Anwendungsgebiete und zur Verfeinerung verschiedener Verfahren anwenden. So auch im Rahmen der Rekonstruktion eines dreidimensionalen Modells einer Szene. Dabei tragen sie zur Glättung des resultierenden Modells bei, denn bekanntlich kommt es bei der Tiefenschätzung von Objekten des Öfteren zu Mehrdeutigkeiten, wodurch es zu Schwankungen in der errechneten Tiefe kommen kann. Diese Schwankungen treten dabei auch innerhalb eines ebenen stehenden Objektes auf, was sich negativ auf die Genauigkeit des Modells auswirkt. Mittels einer TGV-basierten Regularisierung können diese Schwankungen minimiert und dabei, vor allem innerhalb der Szenenobjekte, ein glatteres Modell erzeugt werden. Anders ausgedrückt wird dabei eine TGV-basierte Rauschreduzierung auf die Tiefenkarte des Modells angewendet.

Ein solches TGV²-gestütztes Verfahren zur 3D-Rekonstruktion wurde von Kuschk und Cremers in [19] vorgestellt. Die in diesem Verfahren zu minimierenden Energiefunktion sieht dabei wie folgt aus:

$$\boldsymbol{u} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \left\{ \underbrace{\lambda_s \ |G(\nabla \boldsymbol{u} - \boldsymbol{v})| \ + \ \lambda_a \ |\nabla \boldsymbol{v}|}_{R(\boldsymbol{u})} + \lambda_d \ C(\boldsymbol{u}) \right\}. \tag{3.13}$$

Analog zur Vorstellung der TGV¹ und TGV² Regularisierung stellt $\boldsymbol{u} = u(x, y) \in \Gamma$ die gesuchte Tiefenkarte des Modells dar. Hierbei ist Γ der Definitionsbereich der Tiefenkarte und

enthält alle möglichen Tiefen der Szenenobjekte. $R(\boldsymbol{u})$ repräsentiert die TGV^2-basierte Regularisierungsterm, wobei die zusätzliche Variable G ein zusätzlicher Gewichtsmultiplikator ist, der an die lokalen Strukturen des Bildes anpasst. Er hat die Form eines 2×2 großen Tensors und ist für jedes Pixel im Referenzbild unterschiedlich. Bekannt aus der Diffusion wird er aus den Bildgradienten bestimmt und erlaubt die richtungsabhängige Gewichtung der Regularisierung in Abhängigkeit der Bildstrukturen. Damit kann die Regularisierung und damit die Glättung innerhalb eines Objektes angehoben und zugleich an den Objektgrenzen, an denen der Bildgradient hoch ist, gesenkt werden. Dies erlaubt die Erhaltung von Tiefendiskontinuitäten an Objektgrenzen was für die Abgrenzung von einzelnen Szenenobjekten wichtig ist. Details zur Konstruktion und Verwendung dieses Gewichtstensors sind in Kapitel 4.2 zu finden.

Mittels des Kosten- bzw. Datenterms $C(\boldsymbol{u})$ werden pixelbasiert jeder Tiefe innerhalb des Suchbereichs Γ bestimmte Kosten zugeordnet und damit die Plausibilität der entsprechenden Tiefe für das jeweilige Pixel angegeben. Diese Kosten können dabei nach Belieben ermittelt und realisiert werden. Beispielsweise durch ein Kostenvolumen, welches mittels des Plane-Sweep-Verfahrens aufgebaut wird. Hierbei ist zu beachten, dass die Kosten und die Plausibilität einer Tiefe invers proportional zueinander sind. Je höher die Kosten, desto geringer ist die Plausibilität der Tiefe für das entsprechende Pixel.

3.3.4 *Lösungsstrategie*

Nach der Definition des Energiefunktional zur TGV^2-gestützten Rekonstruktion wird nun eine Lösungsstrategie zur globalen Optimierung vorgestellt. Zudem wird erläutert, wie diese Strategie auf das Funktional aus Gleichung 3.13 angewendet werden kann.

Zur Berechnung der Lösung eines Energiefunktionals, wie es in der TGV-gestützten Optimierung vorkommt, muss das Minimum dieser Funktion gefunden werden. Aus der allgemeinen Kurvendiskussion ist bekannt, dass die notwendige Bedingung für ein Minimum oder Maximum einer Funktion die Nullstelle der ersten Ableitungen ist. Bei einer einfachen Funktion mit einem eindimensionalen Definitionsbereich ist dies lediglich die Ableitung nach der einzigen Funktionsvariable. Bei einem zweidimensionalen Definitionsbereich, wie es in der Bildverarbeitung der Fall ist, müssen dazu zwei partielle Ableitungen berechnet werden. In der globalen Optimierung eines Energiefunktionals spielen die *Euler-Lagrange-Gleichungen* eine wichtige Rolle. Sie setzen sich aus den partiellen Ableitungen des Funktionals zusammen und führen zu einer Differentialgleichung, welche durch verschiedene Iterationsverfahren, wie beispielsweise das *Jacobi-* oder das *Gauß-Seidel-Iterationsverfahren*, gelöst werden kann. Diese Methode ist weit verbreitet und wird häufig zur Lösung von Energiefunktionalen heran gezogen. So auch bei der von Rudin *et al.* vorgestellten TGV^1-basierten Rauschreduzierung [31]. Eine weitere erfolgreiche Strategie bei der Minimierung von Funktionen unter Nebenbedingungen ist das Konzept der *Primal-Dual* Optimierung. Dieses wird auch von Kuschk und Cremers in [19] zur Lösungsfindung angewendet und soll im Rahmen dieser Arbeit adaptiert

werden. Aus diesem Grund wird das Vorgehen zur Primal-Dual Optimierung im Folgenden näher vorgestellt und die Anwendung auf Gleichung 3.13 erläutert.

3.3.4.1 *Primal-Dual Optimierung*

Bei einem Primal-Dual Verfahren wird ein mathematisches Problem aus zwei Sichtweisen betrachtet. Zum einen aus der primären Sichtweise, welche der originalen Formulierung entspricht, und zum anderen aus Sicht einer dualen Formulierung. Diese duale Formulierung der Problemstellung ist das Ergebnis einer Transformation des originalen Problems in einen zweiten Funktionsraum, der es erlaubt weitere Aussagen über die Problemstellung zu machen, die innerhalb des primären Funktionsraums nicht möglich sind. Solche Aussagen können beispielsweise die Abgrenzungen des Wertebereiches der primären Funktion betreffen und damit die möglichen Ergebnisse eingrenzen (vgl. [33]). So ist beispielsweise die duale Formulierung eines Minimierungsproblems eine Maximierungsaufgabe. Dabei gibt die Lösung der dualen Maximierung die untere Grenze der möglichen Lösungen der primären Minimierung an. Sind die primären und dualen Funktionen konvex, so besitzen diese globale Optima, welche durch Gradientenabstieg bzw. -anstieg ermittelt werden können.

Während es verschiedene Transformationsfunktionen gibt (z. B. Fourier- oder Laplace-Transformation), die eine primäre Formulierung in einen dualen Funktionsraum abbilden, haben Kuschk und Cremers in [19] die *Legendre-Fenchel* Transformation (vgl. [33]) gewählt, welche wie folgt definiert ist:

$$f^*(p) = \sup_{x \in \mathbb{R}} \{ px - f(x) \} \quad , \quad p = f'(x). \tag{3.14}$$

Sie transformiert den Funktionsraum $(x, f(x))$ in $(p, f^*(p))$, wobei p die Steigung und $f^*(p)$ die konjugierte Funktion zu $f(x)$ sind. Während x in der primären Formulierung auftritt und dadurch als *Primärvariable* bezeichnet wird, ist p als *Dualvariable* gekennzeichnet. Die konjugierte Funktion $f^*(p)$ wird gemäß Gleichung 3.14 mittels der *Supremum*-Funktion und der Steigung p gebildet. Dies bedeutet, dass ein Punkt x auf $f(x)$ gesucht ist, dessen Tangente p den maximalen Schnittpunkt auf der y-Achse hat. Per Definition ist diese Legendre-Fenchel konjugierte Funktion $f^*(p)$ immer konvex (vgl. [33]). In Bezug auf das Energiefunktional der TGV^2-gestützten Rekonstruktion soll diese Konvexität Abhilfe bei der vorhandenen L_1-Norm im Regularisierungsterm $R(\boldsymbol{u})$ (vgl. Gl. 3.13) schaffen. Denn aufgrund der Tatsache, dass $|x|$ an der Stelle $x = 0$ nicht stetig und damit nicht differenzierbar ist, erschwert die L_1-Norm eine Minimierung mittels Gradientenabstiegs.

3.3.4.2 *Quadratische Lockerung*

Der TGV^2 basierte Regularisierungsterm $R(\boldsymbol{u})$, des von Kuschk und Cremers vorgestellten Energiefunktional aus Gleichung 3.13, ist zwar konvex, jedoch ist es der Kostenterm im Allgemeinen nicht. Dies wirkt sich auf das gesamte Energiefunktionals aus, was eine direkte Anwendung der Primal-Dual Optimierung nicht ermöglicht. Die nicht konvexe Eigenschaft

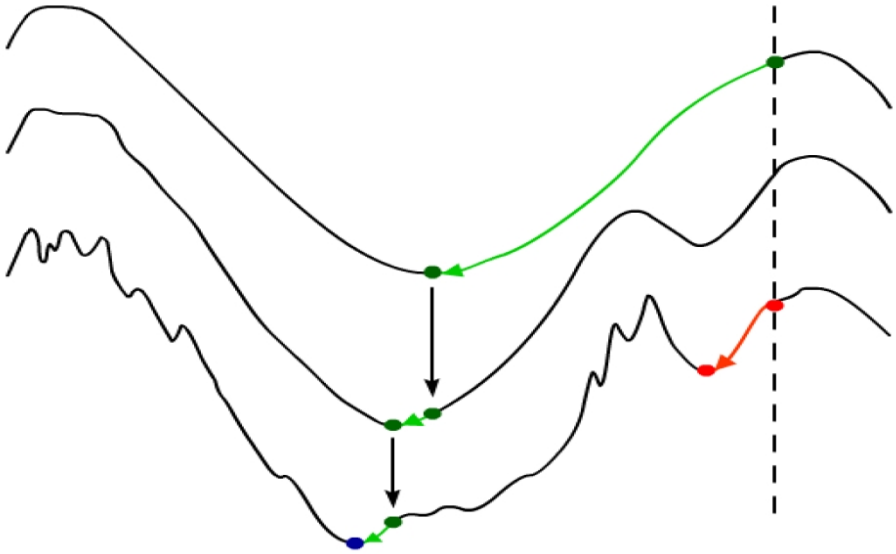

Abbildung 3.5: *Coarse-To-Fine-Warping*: Beginnend mit einer groben Auflösung (oben) wird sukzessiv nach dem Optimum gesucht. Dabei werden lokale Minima in der feinen Auflösung übersprungen. Der Grüne Pfad findet das global Optimum, während der Rote Pfad in einem lokalen Minimum stecken bleibt. Quelle: Vorlesung zu *Correspondence Problems in Computer Vision*, gehalten von Prof. Andrés Bruhn, Sommersemester 2014, Universität Stuttgart.

einer Funktion bedeutet, dass diese lokale Optima besitzt. Ein gradientenbasiertes Schrittverfahren könnte dabei in ein solches lokales Optimum laufen, dort festsitzen und damit nicht das globale Optimum finden.

Um dies zu umgehen wird oft ein sogenanntes *Coarse-to-Fine-Warping* angewendet. Dabei wird das Gradientenabstiegs-Verfahren auf verschiedenen Auflösungsstufen der Funktion angewendet. Wie der Name es bereits andeutet wird hierbei mit einer groben Auflösung begonnen und diese immer weiter verfeinert. Dies glättet die Funktion, wodurch lokale Optima übersprungen werden (vgl. Abb. 3.5). Je gröber die Auflösung eines Bildes ist, desto weniger Details sind darauf zu erkennen. Die Anwendung eines Coarse-to-Fine Warping Ansatzes zur Optimierung der nicht konvexen Funktion würde somit feine Strukturen im Bild glätten und damit aus dem Modell entfernen.

Abhilfe hierfür schafft das in [34] vorgestellte Verfahren. Hierbei wird der konvexe Regularisierungsterm durch *Quadratische Lockerung* von dem nicht-konvexen Kostenterm getrennt. Somit können diese beiden Terme unabhängig voneinander gelöst werden. Ein zusätzlicher Kopplungsterm sorgt dabei für die nötigen Abhängigkeiten zwischen den beiden Termen. Steinbrücker *et al.* haben den Algorithmus zwar im Rahmen der Berechnung des optischen Flusses vorgestellt, jedoch lässt er sich zur Optimierung jeglicher nicht-konvexen Energiefunktionals anwenden. So auch durch Kuschk und Cremers auf Gleichung 3.13 (vgl. [19]):

Zunächst werden die zwei Terme durch die Einführung einer zusätzlichen Variable $\boldsymbol{a} = a(x, y)$ entkoppelt (vgl. Gl. 3.15). Hierbei hat $\boldsymbol{a}$ dieselben Dimensionen wie $\boldsymbol{u}$. Der Kopplungsterm ε erzwingt eine Ähnlichkeit zwischen $\boldsymbol{u}$ und $\boldsymbol{a}$, wodurch verhindert wird, dass $R(\boldsymbol{u})$ und $C(\boldsymbol{a})$ komplett unabhängig voneinander gelöst werden. Diese Entkopplung ergibt:

$$\boldsymbol{u} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \left\{ R(\boldsymbol{u}) \; + \; \lambda_d \, C(\boldsymbol{a}) \; + \; \underbrace{\frac{1}{2\theta} \, (\boldsymbol{u} - \boldsymbol{a})^2}_{\varepsilon} \right\}. \tag{3.15}$$

Gleichung 3.15 wird nun iterativ gelöst, wobei $\theta \; \rightarrow \; 0$ bei jeder Iteration weiter reduziert wird. Dies bewirkt, dass die Lösungen von $\boldsymbol{u}$ und $\boldsymbol{a}$ bei jeder Iteration näher zusammen gezogen werden, was die Gleichheitsbedingung $\boldsymbol{u} = \boldsymbol{a}$ erzwingt. Zusätzlich zum quadratischen Kopplungsterm ϵ erzwingen Kuschk und Cremers in [19] die Gleichheitsbedingung durch einen erweiterten *Lagrange-Multiplikator* $\boldsymbol{L} = L(x, y)$, wodurch sich Gleichung 3.15 weiter zu

$$\begin{aligned} \boldsymbol{u} &= \underset{\boldsymbol{u}}{\operatorname{argmin}} \left\{ R(\boldsymbol{u}) \; + \; \lambda_3 \, C(\boldsymbol{u}) \; + \; \boldsymbol{L} \, (\boldsymbol{u} - \boldsymbol{a}) \; + \; \frac{1}{2\theta} \, (\boldsymbol{u} - \boldsymbol{a})^2 \right\} \\ &= \underset{\boldsymbol{u}}{\operatorname{argmin}} \left\{ \lambda_1 \, |G(\nabla \boldsymbol{u} - \boldsymbol{v})| \; + \; \lambda_2 \, |\nabla \boldsymbol{v}| \; + \; \lambda_3 C(\boldsymbol{u}) \; + \; \boldsymbol{L}(\boldsymbol{u} - \boldsymbol{a}) \; + \; \frac{1}{2\theta} (\boldsymbol{u} - \boldsymbol{a})^2 \right\} \end{aligned} \tag{3.16}$$

umschreiben lässt. Ein solcher Lagrange-Multiplikator wird häufig zur Optimierung unter Nebenbedingungen eingesetzt. Durch die Berechnung der partiellen Ableitungen (jeweils eine pro Funktionsvariable und eine bezüglich dem Lagrange-Multiplikator) kann ein Gleichungssystem aufgestellt und damit eine Lösung ermittelt werden. Nach [19] verbessert dies die Robustheit des Algorithmus bzgl. der Wahl von θ und die Lösungsfindung um einen Faktor zwei.

In Gleichung 3.16 ist nun das endgültig zu lösende Energiefunktional für die TGV^2 gestützte 3D-Rekonstruktion, wie in [19] vorgestellt, dargestellt. Es wird iterativ gelöst, wobei in jeder Iteration der konvexe Regularisierungsterm und der nicht-konvexe Kostenterm unabhängig voneinander berechnet werden. Der erweiterte Lagrange-Multiplikator und der zusätzliche Kopplungsterm binden dabei die beiden Terme immer weiter aneinander. Während $R(\boldsymbol{u})$ nun mittels einem *Primal-Dual* Verfahren gelöst werden kann, werden der Kostenterm $C(\boldsymbol{a})$, sowie die zwei Kopplungsterme mittels einer erschöpfenden Suche im Suchbereich gelöst. Dies scheint ineffizient zu sein, aber aufgrund der lokal begrenzten Eigenschaft der Strukturvergleiche in Bildern, kann diese Suche wirkungsvoll parallelisiert werden.

3.3.4.3 *Der Lösungsalgorithmus*

Mit der endgültigen Aufstellung des zu lösenden Funktionals für die TGV^2-gestützte 3D-Rekonstruktion, wird nun den entsprechende Algorithmus zur Lösung dieses Funktionals vorgestellt. Nach [19] ergibt eine Anwendung der Legender-Fenchel Transformation auf die zwei Teile des Regularisierungsterms die Gleichung

$$\lambda_s \left| G(\nabla \boldsymbol{u} - \boldsymbol{v}) \right| = \underset{\boldsymbol{p} \in P}{\operatorname{argmax}} \left\{ \left\langle G(\nabla \boldsymbol{u} - \boldsymbol{v}) \, , \, \boldsymbol{p} \right\rangle \right\}, \tag{3.17}$$

sowie die Gleichung

$$\lambda_a \left| \nabla \boldsymbol{v} \right| = \underset{\boldsymbol{q} \in Q}{\operatorname{argmax}} \left\{ \left\langle \nabla \boldsymbol{v} \, , \, \boldsymbol{q} \right\rangle \right\}. \tag{3.18}$$

Hierbei werden aufgrund deren Geltungsbereiche $\boldsymbol{u}$ und $\boldsymbol{v}$ als Primärvariablen und $\boldsymbol{p}$ und $\boldsymbol{q}$ als Dualvariablen bezeichnet. Der Ausdruck der Form $\langle a, b \rangle$ steht für das Skalarprodukt zwischen Vektor a und b. Hierfür werden die Variablen in Gleichungen 3.17 & 3.18 als Spaltenvektoren geschrieben. Damit gilt $\boldsymbol{u} \in \mathbb{R}^{WH \times 1}$, $\boldsymbol{v} \in \mathbb{R}^{2WH \times 1}$ sowie $P = \left\{ \boldsymbol{p} \in \mathbb{R}^{2WH \times 1} : \|\boldsymbol{p}\|_\infty \leq \lambda_s \right\}$ und $Q = \left\{ \boldsymbol{q} \in \mathbb{R}^{4WH \times 1} : \|\boldsymbol{q}\|_\infty \leq \lambda_d \right\}$.

Die Formel für das Skalarprodukt $\langle a, b \rangle = |a| \cdot |b| \cdot \cos \varphi$ zwischen zwei Vektoren zeigt, dass dieses maximal ist, wenn der Winkel φ zwischen den zwei Vektoren 0 ist. Somit wird durch das Skalarprodukt in Gleichungen 3.17 & 3.18 bewirkt, dass die Primär- und Dualvariablen zueinander gezogen werden. Die Einschränkungen, dass die *Maximumsnorm*, definiert durch $\|\mathbf{q}\|_\infty = \max_{i=,\cdots n} \{|x_i|\}$, der Dualvariablen kleiner gleich des entsprechenden Gewichtungsmultiplikator λ_i sein muss, erwirkt eine Eingrenzung der primären Lösung und damit die Gewichtung durch den Multiplikator in der primären Problemformulierung. Durch die Transformation ergibt sich für die *Primal-Dual* Formulierung zu Optimierung des Regularisierungsterms,

$$\max_{\boldsymbol{p},\boldsymbol{q}} \min_{\boldsymbol{u},\boldsymbol{v}} \left\{ \left\langle G(\nabla \boldsymbol{u} - \boldsymbol{v}) \, , \, \boldsymbol{p} \right\rangle + \left\langle \nabla \boldsymbol{v} \, , \, \boldsymbol{q} \right\rangle \right\}. \tag{3.19}$$

Danach wird zunächst nach den Dualvariablen maximiert und anschließend nach den Primärvariablen minimiert. Hierbei werden die Maximierung und Minimierung iterativ mittels Gradientenanstieg bzw. -abstieg gelöst.

Der in [19] vorgestellte Algorithmus zur Minimierung des Energiefunktionals aus Gleichung 3.16 und damit zur Berechnung der Tiefenkarte $\boldsymbol{u}$ basierend auf einer TGV^2-gestützten Rekonstruktion ist wie folgt:

1. Für eine Anzahl an Glättungsiterationen i wird der Regularisierungsterm durch das Primal-Dual Verfahren gelöst. Dabei werden die Variablen wie folgt berechnet:
   $\boldsymbol{p}^{i+1} = \Psi_p \left[\boldsymbol{p}^i + \tau_p G\left(\nabla \hat{\boldsymbol{u}}^i - \hat{\boldsymbol{v}}^i\right)\right]$,
   $\boldsymbol{q}^{i+1} = \Psi_q \left[\boldsymbol{q}^i + \tau_q \nabla \hat{\boldsymbol{v}}^i\right]$,
   $\boldsymbol{u}^{i+1} = \Psi_u \left[\frac{\boldsymbol{u}^i + \tau_u \operatorname{div}\left(G\boldsymbol{p}^{i+1}\right) - \tau_u \boldsymbol{L}^n + \frac{\tau_u}{\theta^n}\boldsymbol{a}^n}{1+\frac{\tau_u}{\theta^n}}\right]$,
   $\boldsymbol{v}^{i+1} = \boldsymbol{v}^i + \tau_v \left(\boldsymbol{p}^{i+1} + \operatorname{div}\left(\boldsymbol{q}^{i+1}\right)\right)$,
   $\hat{\boldsymbol{u}}^{i+1} = 2\boldsymbol{u}^{i+1} - \boldsymbol{u}^i$,
   $\hat{\boldsymbol{v}}^{i+1} = 2\boldsymbol{v}^{i+1} - \boldsymbol{v}^i$.
2. Im Anschluss wird das Ergebnis $\boldsymbol{u}^{i+1} = \boldsymbol{u}^n$ dazu verwendet den Datenterm mittels einer erschöpfenden Suche zu lösen:
   $\boldsymbol{a}^{n+1} = \operatorname{argmin}_{\boldsymbol{a}\in\Gamma}\left\{\lambda_d C(\boldsymbol{a}) + \boldsymbol{L}^n(\boldsymbol{u}^n - \boldsymbol{a}) + \frac{(\boldsymbol{u}^n - \boldsymbol{a})^2}{2\theta^n}\right\}$.
3. Der *Lagrange-Multiplikator* wird wie folgt aktualisiert:
   $\boldsymbol{L}^{n+1} = \boldsymbol{L}^n + \frac{(\boldsymbol{u}^n - \boldsymbol{a})^2}{2\theta^n}$.
4. Aktualisierung von θ erfolgt nach
   $\theta^{n+1} = \theta^n(1 - \beta n)$.
5. Für eine Reihe von n globaler Iterationen werden Schritte 1 bis 4 wiederholt.

In den Primal-Dual Iterationen aus Schritt 1 werden die Primär- und Dualvariablen entsprechen durch Gradientenabstieg bzw. -anstieg gelöst. Da eine Positive Divergenz einem negative Gradienten entspricht wird in dem Gradientenabstieg für $\boldsymbol{u}$ und $\boldsymbol{v}$ eine Addition durchgeführt. Für den beschriebenen Algorithmus werden die Primärvariable $\boldsymbol{u}^0$ sowie $\boldsymbol{a}^0$ mit dem einer Tiefen-/Disparitätskarte initialisiert. Die Dualvariablen $\boldsymbol{p}^0$ und $\boldsymbol{q}^0$, sowie $\boldsymbol{v}$ werden zu Anfang auf 0 gesetzt. Für $\hat{\boldsymbol{u}}^0$ und $\hat{\boldsymbol{v}}^0$ gilt das Gleiche wie für $\boldsymbol{u}^0$ und $\boldsymbol{v}^0$. Die Multiplikatoren τ_i stehen für die Schrittweite der Gradienten-Anstiege bzw. Abstiege. Für das Konvergieren des Primal-Dual Schemas müsse die Gradienten- und Divergenzoperatoren negativ adjungiert sein, sodass gilt $\langle \nabla \boldsymbol{u}, \boldsymbol{p}\rangle = \langle -\boldsymbol{u}, \operatorname{div}(\boldsymbol{p})\rangle$ bzw. $\langle \nabla \boldsymbol{v}, \boldsymbol{q}\rangle = \langle -\boldsymbol{v}, \operatorname{div}(\boldsymbol{p})\rangle$. Aus diesem Grund wird der Gradient mittels Vorwärtsdifferenzen und die Divergenz durch Rückwärtsdifferenzen gebildet (vgl. Kapitel 2.4.3). Um die Einschränkungen $\|\boldsymbol{p}\|_\infty \leq \lambda_s$ und $\|\boldsymbol{q}\|_\infty \leq \lambda_a$ zu erfüllen, gelten für $\Psi_p = \frac{\boldsymbol{p}}{(\max\{1, \|\boldsymbol{p}\|_\infty/\lambda_s\})}$ und $\Psi_q = \frac{\boldsymbol{q}}{(\max\{1, \|\boldsymbol{q}\|_\infty/\lambda_a\})}$. Allgemein gilt zudem, dass $\boldsymbol{u}$ in den Gültigkeitsbereich $[0, 1]$ normiert ist. Mittels Ψ_u wird das Ergebnis von $\boldsymbol{u}$ in diesem Gültigkeitsbereich gehalten. Dabei werden die Werte, die die Grenzen überschreiten auf 0 bzw. 1 gesetzt (vgl. [19]).

3.3.4.4 *Subtiefen-Verfeinerung*

Aufgrund der Verwendung der kontinuierlichen TGV^2 Regularisierung wird in der Berechnung von $\boldsymbol{u}$ über die diskreten Tiefenwerte hinweg interpoliert. Um diese in $\boldsymbol{u}$ enthaltenen Zwischenwerte zu bewahren muss auch das Ergebnis von $\boldsymbol{a}$ solche Zwischenwerte enthalten. Um zwischen den diskreten Werten zu interpolieren wird dabei zunächst wie gehabt die diskrete Lösung von Gleichung

$$\boldsymbol{a} = \underset{\boldsymbol{a}\in\Gamma}{\operatorname{argmin}} \left\{ \lambda_d \, C(\boldsymbol{a}) \; + \; \boldsymbol{L}^n \, (\boldsymbol{u}^n - \boldsymbol{a}) \; + \; \frac{(\boldsymbol{u}^n - \boldsymbol{a})^2}{2\theta^n} \right\} \tag{3.20}$$

berechnet. Anschließend wird eine Parabel der Form $at^2 + bt + c$ durch das berechnete Kostenminimum und die links und rechts angrenzenden Kosten gelegt (vgl. Abb. 3.6). Mit diesen Kosten und dem entsprechendem $t \in \{-1, 0, 1\}$ können die Parameter a,b und c der Parabel bestimmt werden. Die Verfeinerung zum diskreten Ergebnis ergibt sich dann aus dem Minimum der Parabel.

Während in [19] hierfür die Parabelgleichung für $C(\mathbf{a})$ in Gleichung 3.20 eingesetzt wird und das Minimum durch die Nullstellung der ersten Ableitung nach t dieser modifizierten Gleichung 3.20 berechnet wird, reicht es, das Minimum lediglich mittels der ersten Ableitung der Parabelgleichung zu errechnen. Wichtig hierbei ist zu beachten, dass $\boldsymbol{u}$ und $\boldsymbol{a}$ in den Bereich $[0, 1]$ normiert sind, wodurch $t \in [\frac{-1}{m}, \frac{1}{m}]$ gelten muss, wobei $m = |\Gamma|$ die Anzahl der Tiefen/Disparitäten ist, über die im Suchraum gesucht wird. Somit ergibt sich für die Verfeinerung $t = \frac{-b}{2a} \cdot \frac{1}{m}$.

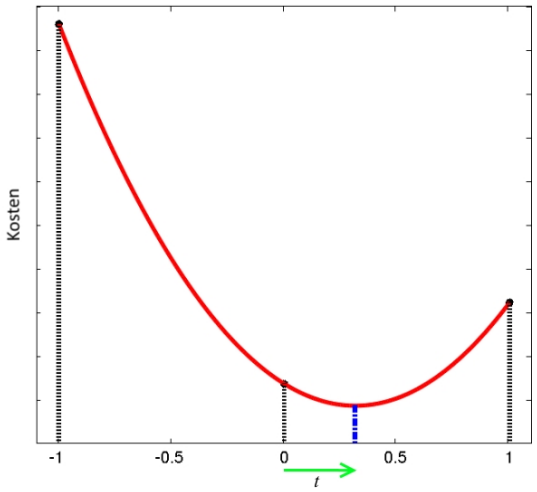

Abbildung 3.6: Berechnung der Subtiefen-Verfeinerung: Hierzu wird eine Parabel durch das Kostenoptimum und den beiden umliegenden Kosten gelegt. Durch Berechnung des Minimums der Parabel wird dabei die Subtiefen-Verfeinerung t zur kostenoptimalen Tiefe berechnet. [19]

4. UMSETZUNG UND ERWEITERUNGEN

Im folgenden Kapiteln wird erläutert, wie als Teil der Masterarbeit die vorgestellten Verfahren miteinander kombiniert, und damit das System für eine echtzeitnahe Rekonstruktion für Structure-from-Motion umgesetzt wurde. Zudem werden verschiedene Erweiterungen vorgestellt, die ebenfalls im Rahmen dieser Arbeit angewendet wurden. Dabei wird zunächst das umgesetzte Framework vorgestellt. Im Anschluss wird auf verschiedene Gewichtstensoren für die TGV^2-gestützte Rekonstruktion eingegangen. Danach werden die Behandlung von Verdeckungen und die Verwendung einer adaptiven Aggregationsnachbarschaft erläutert. Zum Abschluss des Kapitels wird noch kurz die Umsetzung für eine adaptive Ebenenwahl und Parallelisierung diskutiert.

4.1 FRAMEWORK FÜR DIE ECHTZEITNAHE 3D-REKONSTRUKTION

Eine Aufgabe dieser Arbeit war es, ein geeignetes Framework zur echtzeitnahen 3D-Rekonstruktion mittels Structure-from-Motion umzusetzen. Dieses soll dabei aus einer Reihe von Einzelbildern mit dazugehörigen Posen dichte Tiefenkarten berechnen, die im Anschluss in ein dichtes dreidimensionales Modell projiziert werden können. Abbildung 4.1 zeigt die umgesetzte Verarbeitungspipeline. Sie ist aus vier Einzelmodulen (Vorverarbeitung, Plane-Sweep, TGV und Nachverarbeitung) aufgebaut. Jedes dieser Module führt eine Reihe an Berechnungen durch und leitet das Ergebnis an das nächste Modul weiter. Die Eingabedaten für die Rekonstruktion setzten sich aus der Einzelbildsequenz und den dazugehörigen Posen zusammen. Diese Eingabedaten werden in Gruppen von k aufeinanderfolgenden Frames unterteilt, welche nacheinander verarbeitet werden. In jeder dieser Gruppen wird das mittlere Einzelbild als Referenzaufnahme (Keyframe) festgelegt.

Als Ausgabe liefert das Framework nach und nach Tiefenkarten, zusammen mit entsprechenden dreidimensionalen Punktwolken zu den Keyframes der einzelnen Gruppen zurück. Liegen dem Framework zusätzlich Groundtruth-Daten vor, so wird ebenfalls eine Fehlerberechnung für die ermittelte Tiefenkarte durchgeführt, welche zur Analyse und Verbesserung des Verfahrens verwendet werden kann.

Viele der Berechnungen, die in den Pipelinemodulen durchgeführten werden, sind sehr gut parallelisierbar. Um dies auszunutzen wird ein Großteil der Berechnungen auf die Grafikkarte ausgelagert, da diese, aufgrund ihrer hohen Anzahl an Prozessoreinheiten, für eine parallele Ausführung bestens geeignet ist. Mehr zur parallelisierten Umsetzung ist in Kapitel 4.6 zu finden.

Eine der Anforderungen (vgl. Kapitel 1.1) an das Framework ist, dass es die 3D-Rekonstruktion echtzeitnah durchführen kann. Um dies zu gewährleisten ist die Pipeline in zwei Bereiche unterteilt: die On-the-fly-Berechnung und die Offline-Berechnung. Die On-the-fly-

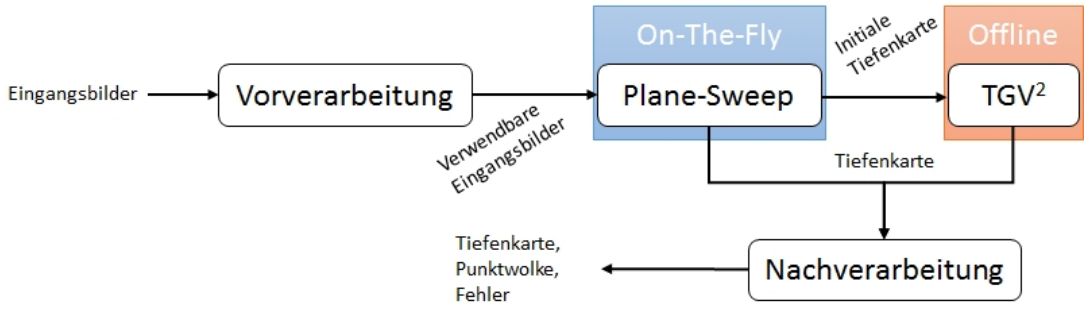

Abbildung 4.1: Verarbeitungsfolge des umgesetzten Frameworks.

Berechnung der Tiefenkarte wird durch ein isoliertes Plane-Sweep-Verfahren realisiert. Dies folgt direkt nach dem Vorverarbeitungsschritt und berechnet in kürzester Zeit eine initiale Tiefenkarte. Diese wird nachbearbeitet und von dem System zurückgegeben. Zusätzlich werden die Tiefenkarten des Plane-Sweep-Verfahrens für die Offline-Verfeinerung mittels des TGV^2-gestützten Verfahrens zwischengespeichert. Diese Verfeinerung wird erst nach Abschluss aller durch das Plane-Sweep-Modul durchgeführten Berechnungen gestartet. Die hohe Laufzeit der Verfeinerung würde bei einer gleichzeitigen Ausführung die Berechnungen des Plane-Sweep-Verfahren verdrängen. Dies würde wiederum zum Verlust der Echtzeitfähigkeit des Systems führen. Im Folgenden werden die einzelnen Pipelinemodule des Frameworks sowie deren Zusammenspiel näher erläutert:

4.1.1 *Schritt 1: Vorverarbeitung*

Im ersten Verarbeitungsschritt werden die eingehenden Einzelbilder für die weitere Verwendung vorbereitet sowie nötige Vorberechnungen durchgeführt. Einer dieser vorbereitenden Schritte ist das *zeitliche Subsampling*, welches nicht relevante Einzelbilder verwirft. Nicht jedes einzelne Frame des Videos ist für die 3D-Rekonstruktion von Bedeutung. So können Einzelbilder verworfen werden, wenn sie sich nicht genug vom vorherigen Frame unterscheiden, oder die Baseline, d. h. die Distanz zwischen zwei aufeinanderfolgenden Aufnahmepositionen, nicht groß genug ist. Bei einer zu geringen Baseline existiert nicht genügend Parallaxe zwischen den Einzelbildern, um daraus die Tiefe zu schätzen. Bei einer späteren Eingliederung des Systems in eine SfM-Pipeline (vgl. Kapitel 2.2.4), ist diese Filterung nicht nötig, da bei SfM die Eingangsbilder bereits durch das Tracking nach Brauchbarkeit selektiert werden. Derzeit wird das System jedoch eigenständig verwendet und ausgewertet. Dadurch ist dieser Schritt notwendig, um nicht-informative Aufnahmen der Eingangssequenz von der Rekonstruktion auszuschließen.

Ein weiterer Schritt dieser Vorverarbeitung ist die eventuelle Vorglättung der zu verwendenden Einzelbilder um mögliches Rauschen zu reduzieren. Diese Vorglättung erlaubt zudem, eine zusätzliche Unterabtastung der Bilder durchzuführen ohne dabei sogenannte *Aliasing-*

Artefakte hervorzurufen. Die Verwendung einer solchen Unterabtastung verringert die Auflösung der Bilder. Dies hat zwar positive Auswirkungen auf die Rechenzeit der Verfahren, jedoch gehen dadurch auch detaillierten Bildstrukturen verloren.

Des Weiteren sind in diesem Verarbeitungsschritt die Berechnung der lokalen Gewichtstensoren (vgl. Kapitel 4.2) sowie die Durchführung der Census-Transformation (vgl. Kapitel 2.3.2) enthalten. Die Gewichtstensoren, die für die spätere TGV^2-gestützte Rekonstruktion wichtig sind, werden für jedes Referenzbildes ermittelt. Die Census-Transformation hingegen wird auf jedes Einzelbild angewendet, das für die 3D-Rekonstruktion berücksichtigt wird.

Wie bereits erwähnt ist das Ergebnis des Vorverarbeitungsschritts eine Gruppe aus k Einzelbildern. Dabei muss $k > 3$ ungerade sein. Diese kann später für das Matching in zwei Teilgruppen unterteilt werden, wobei die Referenzaufnahme in der Mitte der beiden Teile angeordnet ist. Hierdurch kann besser mit Verdeckung von Strukturen umgegangen werden. Diese Thematik wird in Kapitel 4.3 behandelt. Eine Untersuchung über die Größe von k ist in Kapitel 5.3.1.3 zu finden.

4.1.2 *Schritt 2: Plane-Sweep*

In diesem Schritt des Frameworks wird die Gruppe der Einzelbilder aus Schritt 1, zusammen mit den dazugehörigen Posen dazu verwendet, ein gewöhnliches Plane-Sweep-Verfahren (vgl. Kapitel 3.2) und damit eine erste Tiefenschätzung der Szene durchzuführen. Die Tiefenkarte wird dabei für die Referenzaufnahme (das mittlere der fünf Einzelbilder) aufgebaut. Wie in Kapitel 3.2 erläutert können Ebenen mit verschiedenen Normalenvektoren verwendet und entlang diesen durch den Raum verschoben werden. Dies hilft dabei verschiedene Orientierungen der Szenenobjekte zu berücksichtigen und die Ebenen diesen anzupassen. Die durch diesen Verarbeitungsschritt berechnete Tiefenkarte wird zum einen an den Nachverarbeitungsschritt zur Projektion und Fehlerberechnung weitergeleitet, sowie für Verfeinerung mittels des TGV^2-gestützte Verfahrens eingereiht.

4.1.3 *Schritt 3: TGV^2-gestützte Verfeinerung*

Der dritte Verarbeitungsschritt verfeinert mit Hilfe des TGV^2-gestützten Verfahrens (vgl. Kapitel 3.3) die aus dem vorherigen Verarbeitungsschritt resultierende Tiefenkarte. Hierbei werden nur drei der k Einzelbilder zur Tiefenschätzung verwendet um Rechenzeit zu sparen. Die Dreiergruppe setzt sich dabei aus dem Referenzbild, sowie dem ersten und dem letzten der fünf Einzelbilder zusammen. Die nicht verwendeten Aufnahmen werden verworfen. Die Reduktion der Eingangsdaten wird durch die Glattheitsbedingung, die dem Modell in dem TGV^2-gestützten Verfahren auferlegt wird, kompensiert. Obwohl das in [19] vorgestellte Verfahren zur TGV^2-gestützten 3D-Rekonstruktion lediglich zwei Bilder für die Tiefenschätzung verwendet, soll in dieser Umsetzung durch den Abgleich von insgesamt drei Aufnahmen das Auftreten von Verdeckungen ausgeglichen werden (vgl. Kapitel 4.3).

Wie in Kapitel 3.3.4.3 beschrieben wird in der Berechnung der TGV^2-gestützten Rekonstruktion die Variablen $\boldsymbol{u}$ und $\boldsymbol{a}$ mit einer Tiefenkarte initialisiert. Für diese Initialisierung wird hierbei die berechnete Tiefenkarte aus Schritt 2 (Plane-Sweep) verwendet. Das Kostenvolumen, welches bei der Lösung des Kostenterms eine Rolle spielt, wird mit Hilfe eines weiteren Plane-Sweep-Verfahren aufgebaut. Hierbei werden jedoch lediglich die Ebenen verwendet, die frontoparallel zur Referenzaufnahme stehen (Normalenvektor $\vec{n}$ parallel zur optischen Achse). Nach der Berechnung wird die resultierende Tiefenkarte ebenfalls zur Nachverarbeitung (Projektion und Fehlerberechnung) an den letzten Verarbeitungsschritt des Frameworks weitergeleitet.

4.1.4 *Schritt 4: Nachverarbeitung*

Wie aus Abbildung 4.1 hervorgeht, erhält der Nachverarbeitungsschritt des Frameworks dessen Eingangsdaten sowohl von dem gewöhnlichen Plane-Sweep-Verfahren aus Schritt 2, als auch von der verfeinernden TGV^2-gestützten Rekonstruktion aus Schritt 3. Dieses letzte Pipelinemodul führt eine Reihe von Berechnungen durch, die zur Analyse und Visualisierung notwendig sind. Zum einen werden hierbei die berechneten Tiefenkarten in dreidimensionale Punktwolken projiziert. Wie aus Kapitel 2.2.3 bekannt ist, enthalten Tiefenkarten für jedes Pixel die Tiefeninformation des dazugehörigen Szenenpunktes. Mittels dieser Information, der Kenntnis über die Brennweite und der Lokalität des Pixels können durch die in Gleichung 2.1 beschriebene Relation die Koordinaten des dreidimensionalen Szenenpunktes berechnet werden. Des Weiteren wird in diesem Schritt eine Fehlerberechnung für die jeweilige Tiefenkarte durchgeführt. Hierbei werden die berechneten Ergebnisse mit der Groundtruth (falls vorhanden) verglichen und eine Abweichung berechnet. Details zum Qualitätsmaß und der Berechnung des Fehlers ist in Kapitel 5.2 zu finden. Am Ende der Nachverarbeitung werden die berechneten Tiefenkarten zusammen mit den daraus projizierten Punktwolken als Ergebnis des Frameworks zurückgegeben.

4.2 GEWICHTSTENSOREN

In Kapitel 3.3.3 wird das Energiefunktional vorgestellt, welches als Teil der TGV^2-gestützten Rekonstruktion minimiert wird. Dieses besteht bekanntlich aus einem Regularisierungs- und einem Datenterm. Während der Datenterm den Abgleich der Bildstrukturen enthält, sorgt der Regularisierungsterm in der Tiefenschätzung dazu, dass das rekonstruierte Modell lokal konsistent ist. Dabei wird die Tiefenkarte geglättet, sodass das Modell keine starken Sprünge aufweist. Während diese Glättung innerhalb eines Szenenobjektes einen hohen Einfluss haben sollte, gilt für Bereiche an Objektgrenzen das Gegenteil. Um im Modell Tiefendiskontinuitäten zu erhalten, sollten in solchen Bereichen die Glättung ausgesetzt werden, sodass das Energiefunktional Hauptsächlich durch den Datenterm bestimmt wird.

Um dies zu erreichen enthält Gleichung 3.13 ein zusätzlichen Gewichtungsterm G, der sich den Bildstrukturen anpasst und damit den Einfluss des Regularisierungsterms entsprechend der lokalen Bildgegebenheiten steuert. Dieser Term G wird als Gewichtstensor bezeichnet und hat die Form einer 2×2 Matrix. Durch die Multiplikation des Bildgradienten mit dieser Matrix kann der Gradient richtungsabhängig skaliert werden. Denn ein Vektor, der mit einer Matrix multipliziert wird, wird auf einen neuen Vektor abgebildet. Dabei hängt die Abbildung von den Eigenvektoren und Eigenwerten der Matrix ab. Der multiplizierte Vektor wird dabei in Richtung der Eigenvektoren gemäß der Eigenwerte skaliert. Das heißt, dass ein Vektor, der parallel zu einem der Eigenvektoren ist, durch die Multiplikation mit dem entsprechenden Eigenwert skaliert wird. Somit kann, je nach Aufbau des Gewichtstensors, der Bildgradient für die Regularisierung richtungsunabhängig oder richtungsabhängig skaliert werden. Dies ermöglicht eine lokale Gewichtung der Regularisierung lokal. Bekannt aus der nicht-linearen Diffusion (vgl. [35]) wird hierbei von *isotropen* bzw. *anisotropen* Tensoren gesprochen.

4.2.1 *Isotrope Gewichtung*

Eine Gewichtung mittels eines isotropen Tensors erfolgt richtungsunabhängig und bleibt somit für jede Orientierung des Vektors gleich. Sie entspricht einer einfachen Skalierung des Vektors durch einen Skalierungsfaktor. Ein solcher isotroper Gewichtstensors der Größe 2×2 mit dem Skalierungsfaktor $g(\nabla I)$ ist wie folgt definiert:

$$G_{iso} := \begin{pmatrix} g(\nabla I) & 0 \\ 0 & g(\nabla I) \end{pmatrix}. \tag{4.1}$$

Bei der Multiplikation eines Vektors mit diesem Tensor werden die beiden Komponenten des Vektors gleichermaßen mit dem Faktor $g(\nabla I)$ multipliziert, was einer allgemeinen Skalierung des Vektors entspricht. Hierbei wird der Gewichtungsfaktor in Abhängigkeit des Bildgradienten (∇I) gewählt. Dieser ergibt sich aus einer fallenden Gewichtsfunktion, wodurch die Gewichtung mit zunehmenden Gradienten reduziert wird. Da angenommen wird, dass Objektgrenzen hohe Bildgradienten aufweisen, wird durch einen solchen Tensor an Objektgren-

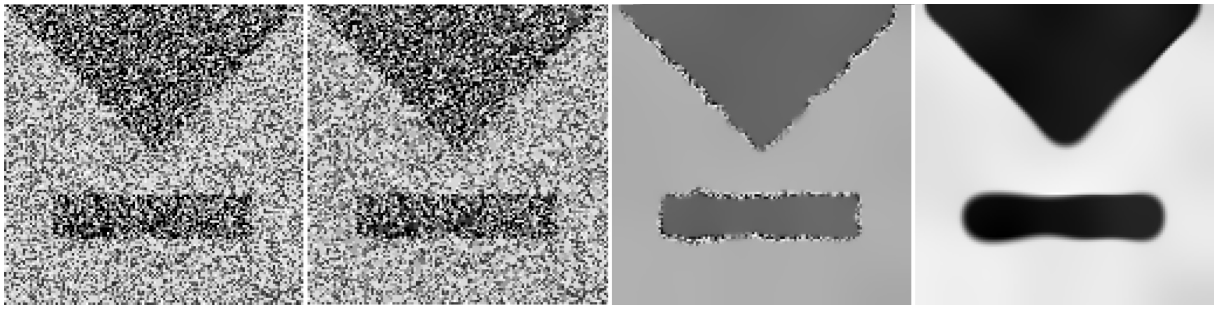

Abbildung 4.2: Von links nach rechts: **(a)** Verrauschtes Testbild. **(b)** Nicht-lineare Diffusion mittels isotropem Tensor. Perona-Malik Term mit $\lambda_{pm} = 2,5$, $t = 80$. **(c)** Nicht-lineare Diffusion mittels isotropem Tensor. Perona-Malik Term mit $\lambda_{pm} = 3,5$, $t = 80$. Vorglättung des Testbildes mit $\sigma = 3$. **(d)** Nicht-lineare Diffusion mittels anisotropem Tensor. Perona-Malik Term mit $\lambda_{pm} = 3,5$, $t = 80$. Vorglättung mit $\sigma = 3$. Quelle: [35]

zen eine geringere Regularisierung durchgeführt. Details zur Bestimmung der Bildgradienten sind in Kapitel 4.2.3 zu finden.

Die Verwendung eines isotropen Gewichtstensors in Gleichung 3.13, bei dem der Gewichtungsfaktor unabhängig vom Bildgradienten mit $g = 1$ gewählt wird, entspricht einer Regularisierung ohne lokale Gewichtungsanpassung. Der Einfluss des Regularisierungsterms aus Gleichung 3.13 würde hierbei lediglich durch die globale Gewichtung λ_s gesteuert werden.

4.2.2 *Anisotrope Gewichtung*

Durch den Gewichtungsfaktor, der das Gewicht in Abhängigkeit des Bildgradienten bestimmt, ermöglicht die Verwendung eines isotropen Tensors bereits eine adaptive Gewichtung. Jedoch hat diese Art von Tensoren auch ihre Grenzen. Wie in Abbildung 4.2 *(b)* dargestellt kann eine isotrope Gewichtung schlecht mit Rauschen umgehen. Aufgrund des betraglich großen Gradienten, der an verrauschten Pixeln auftritt wird der Gewichtungsfaktor stark verkleinert. Aufgrund der Richtungsunabhängigkeit des isotropen Tensors, wird die Diffusion an den entsprechenden Stellen für alle Richtungen ausgesetzt. Dadurch wird selbst in orthogonaler Richtung zum Gradienten nicht diffundiert. Eine einfache Abhilfe schafft eine Vorglättung des Eingangsbildes mittels eines Gaußkerns. Hierbei wird das Rauschen reduziert, wodurch die isotrope Diffusion nicht mehr so stark gehemmt wird (vgl. Abb. 4.2 *(c)*). In Ausschnitt *(d)* der Abbildung 4.2 wird das Ergebnis einer nicht-linearen Diffusion mittels einem anisotropen Tensor dargestellt. Hierbei wird die Diffusion durch die Existenz von Rauschen nicht mehr richtungsunabhängig gehemmt, sondern nur noch in Richtung des Bildgradienten. Dies erlaubt einen weitaus besseren Umgang mit den lokalen Bildstrukturen, was unmittelbar zu einer besseren Glättung führt. Mehr zu anisotropen nicht-linearen Diffusion ist in [35] zu finden.

Ein anisotroper Gewichtstensor erlaubt eine richtungsabhängige Skalierung des mit dem Tensor multiplizierten Vektors. Wie bereits erwähnt wird dabei der Vektor gemäß der Eigenwerte der Matrix in Richtung der dazugehörigen Eigenvektoren skaliert. Das heißt, um eine Gewichtung der Regularisierung zu erzielen, die von den lokalen Bildstrukturen abhängt, müssen die Eigenwerte und Eigenvektoren des Gewichtstensors entsprechend der auftretenden Strukturen gewählt werden. Zunächst wird hierzu festgelegt, wie sich die durch den Tensor hervorgerufene Abbildung, in Bezug auf die lokalen Strukturen im Bild verhalten soll:

1. Damit Diskontinuitäten zwischen verschiedenen Objekten innerhalb der Tiefenkarte erhalten werden, soll die Regularisierung über die Objektgrenzen hinweg ausgesetzt werden. Unter der Annahme, dass Objektgrenzen in Bereichen auftreten wo der Bildgradient hoch ist, sollen somit Vektoren, die in Richtung des Gradienten zeigen und damit parallel zu ihm stehen, durch die Multiplikation mit dem Gewichtstensor geschwächt werden.

2. Um die Regularisierung innerhalb von Szenenobjekten und entlang Objektgrenzen zu begünstigen, sollen Vektoren, die orthogonal zu den Objektgrenzen und damit zum Bildgradienten stehen, nicht verändert werden.

Aus diesen zwei Bedingungen können die Eigenvektoren des gesuchten Gewichtstensors wie folgt definiert werden, sodass diese parallel bzw. orthogonal zum Bildgradienten (∇I) stehen:

$$\vec{v}_1 \parallel \nabla I\,, \quad \vec{v}_2 \perp \nabla I. \tag{4.2}$$

Hierbei gilt, dass die Eigenvektoren $(\vec{v}_1, \vec{v}_2)$ normalisiert sein müssen. Zudem gibt die erste Bedingung an, dass Vektoren in Richtung des ersten Eigenvektors gemäß des Betrags des Bildgradienten skaliert werden sollen. Daraus ergibt sich für den ersten Eigenwert ein Gewichtungsfaktor, der vom Betrag des Bildgradienten abhängt:

$$\lambda_1 = g\left(|\nabla I|\right). \tag{4.3}$$

Da Vektoren parallel zur Objektgrenze nicht skaliert werden sollen, ergibt sich für den zweiten Eigenwert

$$\lambda_2 = 1. \tag{4.4}$$

Die Definition des *anisotropen* Gewichtstensors auf Basis der definierten Eigenwerte und Eigenvektoren ist wie folgt gegeben:

$$\begin{aligned} G_{aniso} \;:= \quad & \lambda_1 \left(v_1 \cdot\; v_1^T\right) \;+\; \lambda_2 \left(v_2 \cdot v_2^T\right) \\ = \; & g\left(\nabla I\right) \left(v_1 \cdot\; v_1^T\right) \;+\; \left(v_2 \cdot v_2^T\right). \end{aligned} \tag{4.5}$$

Durch einen solchen anisotropen Gewichtstensor wird die Regularisierung durch Vektoren, die parallel zum Bildgradienten stehen, mit dem Gewichtungsfaktor $g\left(\nabla I\right)$ gesteuert.

Gleichzeitig wird die Regularisierung entlang von Objektkanten aufgrund von $\lambda_2 = 1$ nicht gehemmt. Aufgrund dieser Eigenschaft werden bei der Diffusion bzw. Regularisierung Kanten von Bildstrukturen hervorgehoben, was solchen Tensoren den Namen „Edge Enhancing Diffusion Tensor“ gibt (vgl. [35]). Die Verwendung einer *anisotropen* Gewichtung zur Regularisierung, wurde bereits in [36] von Nagel und Enkelmann vorgeschlagen.

4.2.3 *Wahl der Gewichtungsfaktoren*

Eine wichtige Rolle, sowohl beim isotropen als auch beim anisotropen Gewichtstensor spielt der Gewichtungsfaktor $g\left(|\nabla I|\right)$. Dieser passt die Stärke der Gewichtung an den Betrag des Bildgradienten an und reguliert damit die Skalierung in Abhängigkeit des auftretenden Bildgradienten. Dieser gibt dabei Aufschluss über die im Bild vorkommenden Objektgrenzen. Dabei wird in der Regel ein betraglicher Schwellenwert des Gradienten festgelegt, ab dem der Gradient als Objektkante identifiziert wird.

In Ihrer Abhandlung [37] von 1990 stellen Perona und Malik einen Gewichtungsfaktor für die Kantenerkennung mittels isotroper Diffusion vor. Dieser sogenannte *Perona-Malik*-Term lässt sich wie folgt definieren:

$$g_{pm}\left(\nabla I\right) := \frac{1}{1 + \left(|\nabla I|^2 \,/\, \lambda_{pm}^2\right)} \,. \tag{4.6}$$

Die violette Kurve in Abbildung 4.3 zeigt den Verlauf dieser Funktion in Abhängigkeit vom Kontrastparameter λ. Ist der Bildgradient betraglich größer als λ, so nimmt das Gewicht ab, während für Bildgradienten die kleiner sind als λ die Gewichtung zunimmt. Im Zusammenhang der nicht-linearen Diffusion spricht man dabei von einer Rückwärts- ($|\nabla I| > \lambda$) bzw. einer Vorwärtsdiffusion ($|\nabla I| < \lambda$). Dabei werden bei einer Rückwärtsdiffusion die Kanten hervorgehoben und bei einer Vorwärtsdiffusion die Kanten entsprechend geglättet. Dadurch kann λ als Schwellenwert interpretiert werden, ab dem ein Gradient als Kante identifiziert und damit die Regularisierung gedämpft wird.

Wie in [35] erläutert, ist dieser Perona-Malik-Term jedoch schlecht gestellt und kann bei der nicht-linearen Diffusion zu theoretischen Problemen führen. Um dies zu umgehen wird in [38] eine Verbesserung von schlecht gestellten Problemen im Zusammenhang mit nicht-linearer Diffusion vorgestellt. Diese verbesserte Gewichtsfunktion, in Form von

$$g_{ch}\left(\nabla I\right) := \frac{1}{\sqrt{1 + \left(|\nabla I|^2 \,/\, \lambda_{ch}^2\right)}} \,, \tag{4.7}$$

wird in dieser Arbeit als *Charbonnier*-Term bezeichnet. Die dazugehörige Kurve ist in Abbildung 4.3 mit grün gekennzeichnet. Aus Abbildung 4.3 geht hervor, dass die Kurve des Charbonnier-Terms wesentlich langsamer abfällt als die des Perona-Malik-Terms. Für einen ähnlichen Kurvenverlauf muss dabei $\lambda_{ch} \approx \frac{1}{2}\lambda_{pm}$ gewählt werden (vgl. blaue und gelbe Kurve).

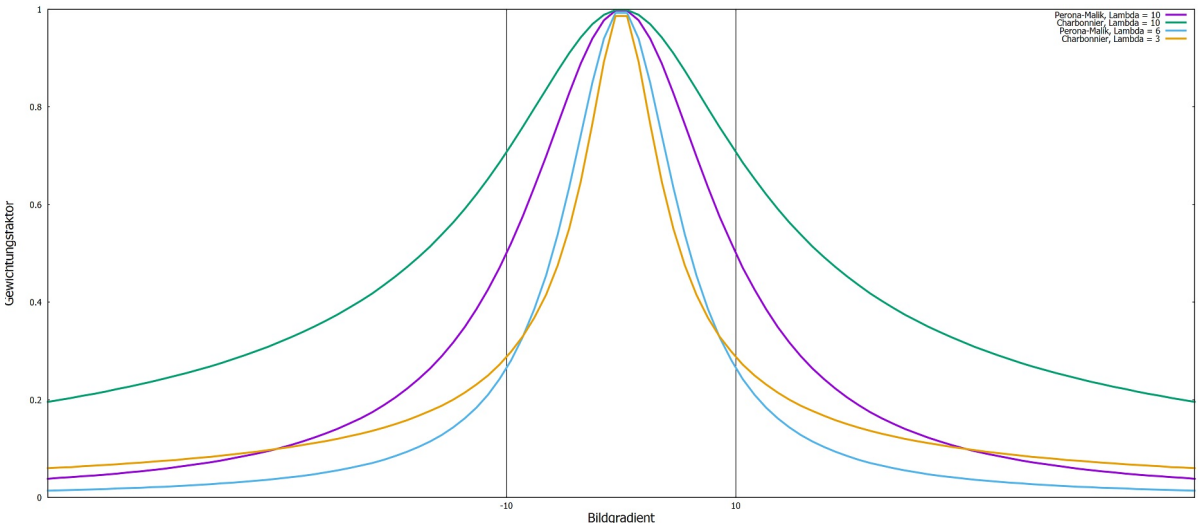

Abbildung 4.3: **Violett:** Kurve des Perona-Malik-Gewichtungsfaktor mit $\lambda_{pm} = 10$. **Grün:** Charbonnier Gewichtungsfaktor mit $\lambda_{ch} = 10$. **Blau:** Kurve des Perona-Malik-Gewichtungsfaktor mit $\lambda_{pm} = 6$. **Gelb:** Charbonnier Gewichtungsfaktor mit $\lambda_{ch} = 3$

.

In ihrer Arbeit [19] verwenden Kuschk und Cremers eine Exponentialfunktion als Gewichtungsfaktor für den anisotropen Gewichtstensor G, welche von Alvarez *et al.* in [39] vorgestellt wurde. Darin wird der Gewichtsfaktor durch

$$g_{exp}\left(\nabla I\right) := \exp(-\alpha \left|\nabla I\right|^{\beta}) \tag{4.8}$$

definiert. Abbildung 4.4 zeigt exemplarisch verschiedene Konfigurationen der Exponentialfunktion. Je nach Wahl der Parameter α und β kann dabei die Gewichtsfunktion nach Belieben angepasst werden. Die Kurven in Violett, Grün und Blau laufen dabei sehr spitz nach oben zusammen. Die gelbe und orangene Kurve haben eine etwas flachere Spitze. Dabei fällt die Kurve in Gelb steiler ab als die Kurve in Orange. An dieser Stelle sollte erwähnt werden, dass die Verwendung der Exponentialfunktion als Gewichtsfaktor eine Anpassung des Wertebereichs der Eingangsbilder oder des Definitionsbereiches der Exponentialfunktion erfordert. Da der Haupteinfluss der Funktionen im Bereich von $[-1, 1]$ ist, sollte entweder α entsprechend gewählt oder der Wertebereich des Eingangsbildes in den Beriech $[0, 1]$ normiert werden.

4.2.4 *Verbesserung der Kantendetektion*

Mit Hinblick auf die Rekonstruktion von Gebäuden mittels Luftaufnahmen, haben Kuschk und Cremers in ihrer Abhandlung [19] für die Bestimmung des lokalen Gewichtstensors ein *Liniensegment-Detektor* (LSD) integriert, um die Tiefendiskontinuitäten an Gebäudekanten besser hervorzuheben. Der dabei verwendete LSD wurde in [40] vorgestellte. Im Folgenden wird in Kürze erläutert, wie das Verfahren von Gioi *et al.* zur Detektion von Liniensegmenten funk-

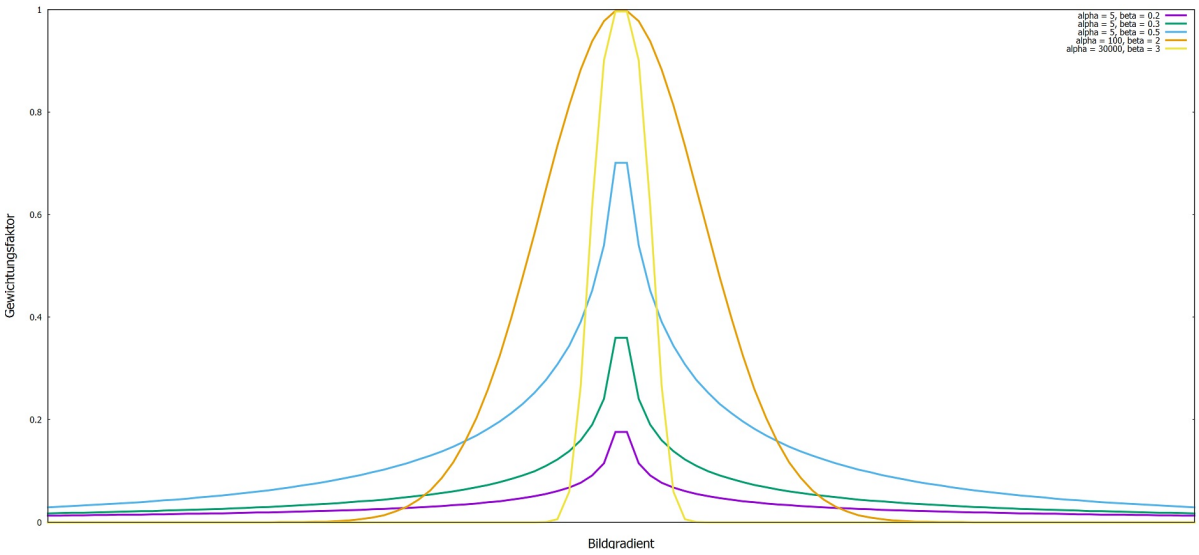

Abbildung 4.4: Verschieden Konfigurationen der exponentiellen Gewichtungsfunktion aus [39]. Haupteinfluss der Funktion im Intervall $[-1, 1]$. Die Abgeschnittenen Kurven laufen sehr Spitz nach oben zusammen. Alle Kurven haben ihr Maximum bei 1.

tioniert, und wie es genutzt werden kann, um den anisotropen Gewichtstensor noch besser an die Objektgrenzen anzupassen. Der LSD extrahiert in kurzer Berechnungszeit aussagekräftige Liniensegmente aus einem gegebenen Graustufenbild. Hierbei werden folgende Schritte ausgeführt:

---

1. Für jedes Pixel werden mittels finiter Differenzen sogenannte „Level-Lines“ ermittelt (vgl. Abb. 4.5 *(b)*). Diese geben pro Pixel mögliche Kantenverläufe an. Zusammen ergeben die einzelnen Linien ein „Level-Line“-Feld.

2. Als nächstes werden durch eine „Region-Growing“-Methode die „Level-Lines“ gruppiert. Dabei werden die „Level-Lines“, die bis zu einer Toleranz (τ) die gleiche Orientierung haben, zusammengefasst. Diese Gruppierungen ergeben die sogenannten „Line-Support-Regions“ (vgl. Abb. 4.5 *(c)*).

3. Jede dieser „Line-Support-Regions“ wird nun mit einem Rechteck umrahmt, welches als mögliches Liniensegment in Frage kommt (vgl. Abb. 4.5 *(d)*).

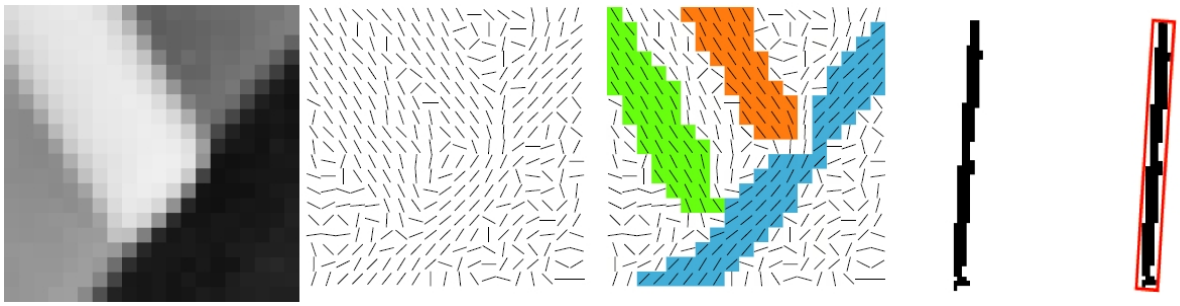

Abbildung 4.5: Einzelne Zwischenergebnisse des Liniensegment-Detektors. Von links nach rechts: **(a)** Testbild. **(b)** „Level-Line"-Feld. **(c)** „Line-Support-Regions" eingefärbt in Grün, Orange und Blau. **(d)** „Line-Support-Regions" umrahmt von einem Rechteck. Validiertes Rechteck ergibt mögliches Liniensegment. Quelle: [40]

4. Abschließend werden die Rechtecke validiert. Diese Validierung erfolgt aufgrund des Verhältnisses von Pixeln, die zu den „Line-Support-Regions" gehören, und der Gesamtanzahl der Pixel, die in dem Rechteck enthalten sind.

5. Besteht ein Rechteck die Validierung, so wird es als Liniensegment markiert. Hierbei bilden die Liniensegmente das Ergebnis dieses Algorithmus. Abbildung 4.6 zeigt das Ergebnis des Liniensegment-Detektors in Standardkonfiguration. Ausgeführt auf einer Aufnahme aus dem neuen Tsukuba-Stereo-Datensatz (vgl. Kap. 5.1).

Um die anisotropen Gewichtstensoren mittels des erkannten Liniensegmente zu verbessern, werden zunächst die Gewichtstensoren G basierend auf dem Eingangsbild, sowie die Gewichtstensoren G' basierend auf dem Ergebnisbild des LSDs berechnet. Anschließend werden die Tensoren G an den Stellen, an denen die Liniensegmente detektiert wurden, durch die Gewichtstensoren G' ersetzt. Aufgrund des hohen Gradienten zwischen den Liniensegmenten und dem Hintergrund, bewirken die Tensoren in G' eine niedrige Glättung über die Liniensegmente hinweg. Gleichzeitig wird aber entlang der Liniensegmente und damit entlang der Kanten regularisiert. Dadurch werden stückweise gerade Objektkanten präziser hervorgehoben, was bei Häuserfassaden zu einem Qualitativ schöneren Ergebnis führen kann. Eine Evaluation über die Auswirkung des LSDs als Teil der Bestimmung der Gewichtstensoren ist in den experimentellen Ergebnissen zu finden.

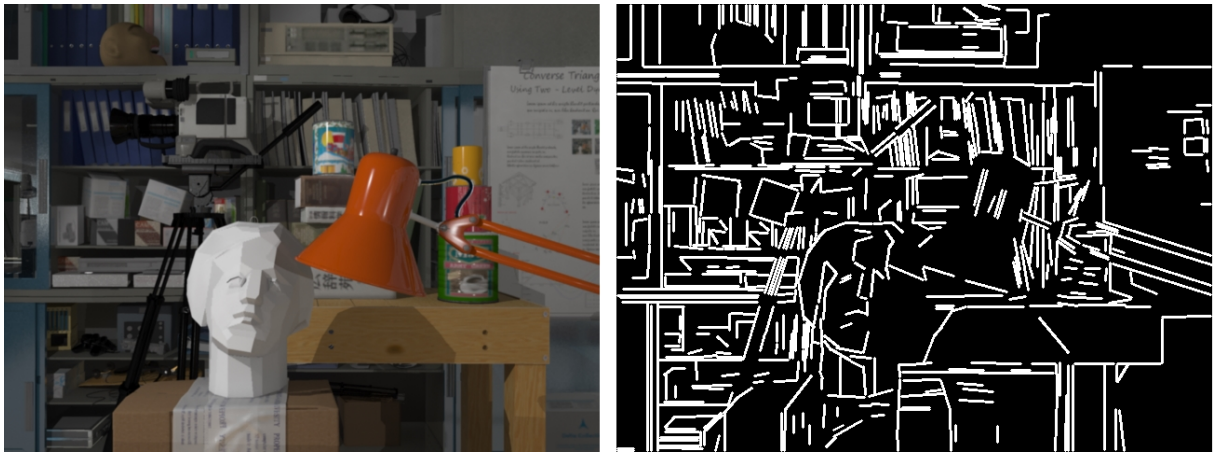

Abbildung 4.6: Von links nach rechts: **(a)** Aufnahme des neuen Tsukuba-Stereo-Datensatzes (vgl. Kap. 5.1). **(b)** Ergebnis des Liniensegment-Detektors in Standardkonfiguration ausgeführt auf (a).

4.3 UMGANG UND BEHANDLUNG VON VERDECKUNGEN

Sowohl im Plane-Sweep-Verfahren, als auch in der TGV^2-gestützten Rekonstruktion, werden mehr als zwei Eingangsbilder verwendet. Die Verwendung mehrerer Aufnahmen ermöglicht aufgrund der Mittelung der Kosten eine robustere Aussage über die Plausibilität der entsprechenden Tiefe. Des Weiteren kann durch einen Vergleich mit mehreren Bildern besser mit Verdeckungen umgegangen werden. Hierbei wird angenommen, dass die Objekte, welche in der Referenzkamera zu sehen sind, meist nur in einer Teilgruppe der Vergleichsbilder verdeckt sind. Werden nur zwei Aufnahmen verwendet, dann können Hintergrundobjekte nicht wieder gefunden werden, wenn sie in der einen Kamera zu sehen sind und in der Anderen verdeckt werden. Dies führt bekanntlich zu Mehrdeutigkeiten in der Tiefenschätzung. Die Abhandlung [41] von Kang *et al.* stellt simple aber durchaus effektive Methoden vor, wie mehrere Aufnahmen dazu verwendet werden können mit der *Verdeckungsproblematik* umzugehen.

Eine dieser Methoden wird von Kang *et al.* als *Temporal Selection* bezeichnet. Anstelle alle Einzelbilder zum Abgleich mit der Referenzaufnahme zu verwenden, verfolgt diese Methode das Ziel nur diejenigen Bilder zu verwenden, in denen die abzugleichenden Pixel auch sichtbar sind. Dazu wird angenommen, dass Bildbereiche, die in der Referenzaufnahme teilweise verdeckt sind, häufig entweder in den zeitlichen Vorgängern oder Nachfolgern nicht verdeckt sind. In der Umsetzung werden hierbei die Kosten nicht aus der Gesamtsumme der Vergleichskosten aller Aufnahmen berechnet (vgl. Gl. 3.10). Vielmehr werden die Kosten zunächst für die Vergleiche der linken und rechten Teilmengen der Aufnahmen separat berechnet und auf-

summiert. Die Gesamtkosten ergeben sich dann aus dem Minimum der beiden Teilsummen durch:

$$\begin{aligned} Cost_{total} &= \min\left\{Cost_{left},\ Cost_{right}\right\} \quad \text{mit } Cost_{left} \ = \sum_{k<ref} Cost(I_{ref}, I_k, H_{I_{ref},I_k}) \ , \\ Cost_{right} &= \sum_{k>ref} Cost(I_{ref}, I_k, H_{I_{ref},I_k}) \ . \end{aligned} \tag{4.9}$$

Diese Methodik hat jedoch ihre Grenzen und funktioniert nicht bei Objekten, die in den Aufnahmen abwechselnd sichtbar oder verdeckt sind. Solch ein schneller Wechsel der Sichtbarkeit kann beispielsweise durch einen Lattenzaun hervorgerufen werden. Um dies zu umgehen verallgemeinern Kang *et al.* ihre Temporal Selection Methode und verwenden aus den Aufnahmen zum Abgleich nur die Hälfte, die die geringsten Vergleichskosten haben und somit die besten 50% aller Vergleiche repräsentieren. Umgesetzt wird diese Methode in dem alle Vergleichskosten separat berechnet werden und nur die niedrigsten 50% zu den Gesamtkosten aufsummiert werden.

Im Rahmen dieser Arbeit wurde für die Berechnung der initialen Tiefenkarte mittels dem alleinigen Plane-Sweep-Verfahrens, bei dem $k > 3$ Aufnahmen zur Ermittlung der Kosten verwendet werden, die verallgemeinerte Methode des Temporal Selection angewendet. Da bei der Verfeinerung mittels des TGV^2-gestützten Verfahrens ohnehin nur drei Aufnahmen verwendet werden, und somit die niedrigsten 50% der Kosten sich entweder aus dem Vergleich mit dem linken oder rechten Bild ergeben, wird direkt das Minimum der beiden Teilkosten verwendet. Die Methodik der Temporal Selection wird ebenfalls von Pollefeys *et al.* in [9] verwendet.

4.4 ADAPTIVE AGGREGATION

In Kapitel 2.3 wird beschrieben wie Bildstrukturen abgeglichen und damit Korrespondenzen zwischen verschiedenen Bildern gefunden werden. Dabei werden die Vergleichskosten über eine Nachbarschaft fester Größe aggregiert. Da nicht nur einzelne Pixel verglichen werden, führt das zur Reduzierung von Fehlzuordnungen, was die Suche nach Korrespondenzen robuster macht (vgl. Abb. 2.10). Gleichzeitig werden durch die Aufsummierung jedoch Tiefendiskontinuitäten an Objektgrenzen verfälscht. Denn bei der Aggregation über eine Nachbarschaft wird implizit angenommen, dass alle Pixel innerhalb des Aggregationsbereiches die gleiche Tiefe haben. Während dies für Bildbereiche die komplett innerhalb eines Szenenobjektes liegen keine negativen Auswirkungen hat, führt es an Objektgrenzen mit Tiefendiskontinuitäten dazu, dass die Nachbarschaftspixel des entfernteren Objektes als Teil des näheren Objektes angenommen werden. Dadurch werden die Grenzen der Vordergrundobjekte weiter nach außen verschoben, wodurch diese Objekte größer werden. Dieses Phänomen ist auch unter dem Namen „foreground-fattening“ bekannt. Eine eingängige Methode um das Auftreten dieses Phänomens zu reduzieren ist, die Aggregationsnachbarschaft so anzupassen, dass nur Pixel eines Szenenobjektes miteinander aufsummiert werden. Dabei sollte die Größe der Nachbarschaft nicht kleiner werden, da dies wiederum zu Mehrdeutigkeiten in der Suche nach Punktkorrespondenzen führt.

Eine Methode die Aggregationsnachbarschaft den Objekten anzupassen ist die Verwendung der sogenannten *spatially-shiftable-windows*, welche von Kang *et al.* in [41] verwendet werden. Hierbei werden die Nachbarschaften um das Referenzpixel so verschoben, dass zumindest ein Großteil der Nachbarschaft innerhalb demselben Szenenobjekt liegt. So können für Referenzpixel, die in der Nähe einer Objektgrenze liegen, die Nachbarschaften vollständig innerhalb des Szenenobjektes gehalten werden. Dies gilt natürlich nur für Objekte deren Größe mindestens die Maße der Aggregationsnachbarschaft umfassen.

Eine weitere Strategie, die zur Anpassung der Nachbarschaft verfolgt werden kann, ordnet den umliegenden Pixeln Gewichte entsprechend einer Heuristik zu. Hierbei bleibt die Größe und Orientierung (zentral positioniertes Referenzpixel) der Nachbarschaft konstant. Lediglich der Einfluss der umliegenden Pixel auf die Gesamtkosten wird mittels der Gewichte den Szenenobjekten angepasst. In [42] wird ein solches gewichtsbasiertes Verfahren für die adaptive Aggregation vorgestellt. Dies wird auch von Kuschk und Cremers in [19] verwendet. Die Heuristik, die Yoon und Kweon dabei zur Bestimmung der Gewichte verwenden, basiert auf den *Gastaltprinzipien*. Diese stellen verschiedene Faktoren auf, die zur Wahrnehmung von Objekten ausschlaggebend sind (vgl. [43], [44]). Für die Bestimmung der Gewichte werden in [42] zwei dieser Faktoren herangezogen, die allgemein wie folgt definiert werden können:

- **Faktor der Ähnlichkeit:** Dieser ordnet ähnlich aussehende Objekte derselben Gruppe zu.
- **Faktor der Nähe:** Objekte der gleichen Gruppe liegen in unmittelbarer Nähe zueinander.

Gemäß [42] wird der *Ähnlichkeitsfaktor* zwischen zwei Pixel anhand der Differenz derer Farbwerte aufgestellt. Dabei wird die Differenz der Farben im CIELAB-Farbraum gemessen. Dieser umfasst alle, durch das menschliche Auge wahrnehmbare Farben und stellt diese in einen dreidimensionalen Raum dar. Je weiter zwei Farben im CIELAB-Farbraum voneinander entfernt sind, desto unterschiedlicher ist die Wahrnehmung für das menschliche Auge. Gleichzeitig kann angenommen werden, dass eng beieinander liegende Farben sehr ähnlich wahrgenommen werden. Daraus folgt, dass die Distanz zwischen zwei Punkten im CIELAB-Farbraum mit der wahrgenommenen Ähnlichkeit korreliert. Daraus lässt sich nach [42] der Ähnlichkeitsfaktor zwischen zwei Pixel p und q durch

$$f_c(p,q) = \exp\left(-\frac{\Delta c_{p,q}}{\gamma_c}\right) \tag{4.10}$$

bestimmen. Hierbei ist $\Delta c_{p,q}$ die euklidische Distanz zwischen den Farbwerten der beiden Pixel im CIELAB-Farbraum. Nach [42] ist $\gamma_c = 7$ gewählt.

Analog zum Ähnlichkeitsfaktor ergibt sich der Faktor der Nähe aus der euklidischen Distanz zwischen zwei Punkten. Dabei gilt, dass Punkte die näher beieinander liegen eher als gemeinsames Objekt wahrgenommen werden als weiter entfernte Punkte. Dadurch berechnet sich der *Nachbarschaftsfaktor* zweier Pixel gemäß [42] wie folgt:

$$f_d(p,q) = \exp\left(-\frac{\Delta d_{p,q}}{\gamma_d}\right). \tag{4.11}$$

Hierbei ist $\Delta d_{p,q}$ der euklidische Abstand zwischen den Pixeln. Zudem gilt, dass γ_d proportional zur Nachbarschaftsgröße ist und empirisch ermittelt wird. Im Rahmen dieser Arbeit wird für γ_d der Radius der Aggregationsnachbarschaft gewählt.

Aus dem Ähnlichkeitsfaktor und dem Nachbarschaftsfaktor setzt sich das Gewicht zwischen Pixel p und q in der Aggregationsnachbarschaft, wobei p der zentrale Referenzpixel und q das Nachbarschaftspixel ist, wie folgt zusammen:

$$w(p,q) = f_s(p,q) \cdot f_d(p,q). \tag{4.12}$$

Daraus ergibt sich für die aggregierten Vergleichskosten zwischen Pixel p_1 in Bild 1 und Pixel p_2 in Bild 2

$$Cost_{Aggr}(p_1,p_2) = \frac{\sum_{q_1 \in N_{p_1}, q_2 \in N_{p_2}} w(p_1,q_1) \cdot w(p_2,q_2) \cdot Cost(q_1,q_2)}{\sum_{q_1 \in N_{p_1}, q_2 \in N_{p_2}} w(p_1,q_1) \cdot w(p_2,q_2)}. \tag{4.13}$$

Dies bedeutet, dass für jedes Pixel in der Nachbarschaft um p_1 und p_2 die Vergleichskosten zwischen q_1 und q_2 mit den jeweiligen Gewichtungen, bezogen auf den entsprechenden zentralen Referenzpixel, multipliziert und zu den Gesamtkosten aufsummiert werden. Um die verschiedenen aggregierten Kosten vergleichen zu können, werden die Gesamtkosten durch die Summe der Gewichte normiert.

4.5 ADAPTIVE ABTASTUNG

Wie in Kapitel 3.2.2 erläutert, wird beim Plane-Sweep-Verfahren eine Ebene entlang ihres Normalenvektors durch den Raum $\mathbb{R}^3$ geschoben. Dabei wird versucht die Ebene lokalen Strukturen anzupassen um dadurch ein dreidimensionale Modell zu erhalten. Hierbei spielt neben der Wahl des Normalenvektors n, die Schrittweite Δd zwischen den einzelnen Ebenen eine wichtige Rolle. Eine Möglichkeit ist es die Ebenen äquidistanten zu verschieden. Nach [45] ist es jedoch bei Aufnahmen mit hochfrequenten Strukturen wichtig, die Ebenen entsprechend der Abtastung des Bildes zu wählen. Die Bildabtastung ist dabei durch die Pixel gegeben ist. Ähnlich wie in [9] besteht im umgesetzten Framework die Möglichkeit, die Szene in adaptiven Abständen abzutasten. Dabei soll der maximale Unterschied zwischen den Pixelverschiebungen, die durch die homographische Abbildung zweier aufeinanderfolgenden Ebenen induziert werden, kleiner-gleich 1 sein. Dies soll den Abglich von Bildern mit häufig wechselnder Struktur begünstigen.

Im Folgenden ist das Vorgehen zur Bestimmung der entsprechenden Ebenen beschrieben. Hierbei kann, um die maximale relative Pixelverschiebung zwischen zwei Ebenen zu berechnen, die Menge der zu berechnenden Daten stark eingegrenzt werden:

---

1. Die größte Pixeldisparität zwischen dem Referenzbild und den Vergleichsbildern wird durch diejenige Aufnahme hervorgerufen, die am weitesten von der Referenzaufnahme entfernt ist. Somit wird zunächst die Aufnahme gesucht, die die längste Baseline zum Referenzbild aufweist. Zur Bestimmung der einzelnen Ebenen wird anschließend nur zwischen diesen beiden Aufnahmen transformiert.
2. Die Pixelverschiebungen aller Pixel innerhalb eines Bildes, die bei einer homographischen Transformation auftreten, werden durch die Verschiebungen der vier Eckpunkte des Bildes begrenzt. Somit reicht es die Disparitätsänderungen der Eckpunkte zwischen zwei aufeinanderfolgenden Ebenen zu vergleichen.
3. Daraus folgt das Vorgehen zur Bestimmung der adaptiven Schrittweite: Für eine Reihe an Ebenenkonfigurationen werden nach und nach die Eckpunkte des Referenzbildes in die Aufnahme, die am weitesten von der Referenzaufnahme entfernt ist, transformiert. Wenn der Unterschied der Verschiebungen zwischen dieser und der letzten ausgewählten Ebene den betraglichen Wert 1 überschreitet, wird diese Ebene zu den ausgewählten Ebenen hinzugefügt. Ansonsten wird sie verworfen.

Die hierbei hervorgerufene unregelmäßige Abtastung hat zusätzliche Auswirkungen auf die Berechnung der Subtiefen-Verfeinerung (vgl. Kapitel 3.3.4.4). Während Δt_{links} und Δt_{rechts} bei einer äquidistanten Abtastung betraglich gleich sind, können sie nun gänzlich andere Werte annehmen. Um dies zu berücksichtigen muss die Berechnung der Parabelparameter a und b durch die Hinzunahme von Δt_{links} und Δt_{rechts} verallgemeinert werden.

4.6 PARALLELISIERUNG

Viele der Berechnungen, die als Teil der vorgestellten Verfahren durchgeführt werden, sind äußerst zeitintensiv. Gleichzeitig können diese Berechnungen sehr gut parallelisiert werden, da in den jeweiligen Berechnungen immer nur eine lokale Teilgruppe der Daten benötigt wird. Um eine echtzeitnahe Laufzeit zu ermöglichen werden die Verfahren für eine Berechnung auf der Grafikkarte umgesetzt.

Der Hauptprozessor (CPU) eines Computers ist in erster Linie für die Ausführung und Steuerung von Programmen gedacht und dafür entsprechend ausgelegt. Hierbei wird der Umgang mit komplexen Programmstrukturen optimiert, um dadurch die Reaktionszeit auf Veränderungen, die durch den Benutzer hervorgerufen wurden, zu minimieren. Zudem erlaubt die geringe Anzahl an einzelnen Prozessorkernen nur eine geringe Parallelisierung. Der Grafikprozessor (GPU) andererseits ist für die Ausführung kleinerer, vordefinierter Operationen, wie beispielsweise Transformationen, für eine hohe Anzahl an Pixel ausgelegt. Aus diesem Grund besitzt die Grafikkarte eine große Menge an Prozessoreinheiten, die pro Pixel parallel dieselben Operationen ausführen. Hierbei wird die Grafikkarte vom Hauptprozessor mit den Daten gefüttert und kann diese in möglichst effizienter Art und Weise bearbeiten. Während früher die Benutzung des Grafikprozessors lediglich für Grafikberechnungen bestimmt war, wird von den Herstellern inzwischen Schnittstellen angeboten, durch die die Prozessoreinheiten der Grafikkarte auch für andere Berechnungen genutzt werden können. Durch diese sogenannte „*General Purpose Computation on Graphics Processing Unit*“, kurz *GPGPU*, können Algorithmen und Berechnungen aufgrund der hohen Anzahl von separaten Prozessoreinheiten gut parallelisiert und damit stark beschleunigt werden.

Diese Möglichkeit soll auch im Rahmen dieser Arbeit ausgenutzt werden. Alle Berechnungen, die in den vorgestellten Algorithmen durchgeführt werden, verwenden nur einen geringen und lokal begrenzten Teil der vorhandenen Daten, wie beispielsweise einzelne Pixelvergleiche. Dies erlaubt eine Verteilung und damit eine Parallelisierung der Berechnungen auf die einzelnen Prozessoreinheiten der Grafikkarte. Dabei wird auf jeder Einheit parallel die gleiche Operation für jeweils eine andere Teilgruppe der Daten ausgeführt. Des Weiteren eignet sich die Verwendung der Grafikkarte gerade für das Plane-Sweep-Verfahren besonders gut, denn ein großer Teil der Berechnungen des Plane-Sweep-Verfahrens macht die homographische Projektion der Bilder aus der einen in die andere Kamera aus. Bei dem Aufbau einer computergenerierten Welt spielt die perspektivische Projektion von Texturen auf die entsprechenden Ebenen ebenfalls eine große Rolle. Aus diesem Grund enthalten Grafikkarten eine äußerst effiziente Implementierung der perspektivischen Projektion, welche für die homographische Abbildungen ausgenutzt werden kann.

Zur Umsetzung der Algorithmen auf der Grafikkarte wird im Rahmen dieser Arbeit die *OpenCL*-Bibliothek[1] der Khronos Group verwendet. Diese Bibliothek ist dabei plattformunabhängig, wodurch die Implementierungen auf verschiedene Hardware zur Parallelisierung portiert werden kann. Die Algorithmen der Verfahren werden dabei zunächst in einzelne, lo-

1 https://www.khronos.org/opencl/

kal begrenzte Berechnungen unterteilt. Diese Berechnungen werden in sogenannten „Kernel"-Programmen umgesetzt. Dieses sind kleine Programme die simultan auf den einzelnen Prozessoreinheiten durchgeführt werden und dabei immer eine andere Teilgruppe der Daten bearbeiten. Für die Berechnung der Algorithmen werden die Daten anschließend auf die Grafikkarte geladen und die einzelnen „Kernel"-Programme der Reihe nach ausgeführt. Dabei wird auf jeder Prozessoreinheit die Operation für eine andere Teilgruppe der Eingangsdaten (z. B. einzelne Pixel) durchgeführt. Nach Beendigung der Berechnungen werden die Ergebnisse wieder aus der Grafikkarte ausgelesen.

Mehr zu GPGPU ist auf den Herstellerseiten, sowie beispielsweise unter gpgpu.org, oder in der entsprechenden Literatur zu finden.

5. AUSWERTUNG

Als letzter Teil dieser Arbeit sollen die Ergebnisse des umgesetzten Systems anhand geeigneter Testdatensätze evaluiert werden. Das folgende Kapitel widmet sich dieser Auswertung und stellt die endgültigen Ergebnisse der Arbeit vor. Hierzu werden zunächst die verwendeten Testdatensätze sowie ein geeignetes Qualitätsmaß für die quantitative Evaluation erläutert. Anschließend werden einige experimentelle Ergebnisse des Systems vorgestellt. Dabei werden die Ergebnisse der beiden Verfahren zunächst unabhängig voneinander betrachtet. In einer anschließenden Diskussion werden die einzelnen Ergebnisse erörtert und mit einander in Verbindung gesetzt. Dies führt zu einer abschließenden Vorstellung der Ergebnisse des umgesetzten Systems zur echtzeitnahen 3D-Rekonstruktion.

5.1 DIE TESTDATENSÄTZE

Für die Auswertung des umgesetzten Systems zur echtzeitnahen 3D-Rekonstruktion werden im Rahmen dieser Arbeit zwei verschiedene Testdatensätze herangezogen:

- Der neue *Tsukuba-Stereo-Datensatz*[1] [46] [47] aus dem Jahr 2012,
- und der *Middlebury-Stereo-Datensatz*[2] [48] [49] von 2001 und 2003.

Während primär die Daten aus dem neuen Tsukuba-Stereo-Datensatz zur Auswertung verwendet werden, dient der Middlebury-Stereo-Datensatz zum Abgleich der Ergebnisse mit denen von Kuschk und Cremers aus [19]. Im Folgenden werden die Datensätze in Kürze vorgestellt sowie deren Verwendung erläutert.

5.1.1 *Neuer Tsukuba-Stereo-Datensatz*

Der neue Tsukuba-Stereo-Datensatz von 2012 (Tsukuba-2012) ist ein umfangreicher synthetischer Stereo-Datensatz. Er besteht aus 1800 Einzelbildern eines stereoskopischen Kameraaufbaus, der sich durch einen computergenerierten Raum bewegt. Dabei sind die Einzelbilder jeweils für die linke und die rechte Kamera gegeben. Die Auflösung der Einzelbilder liegt bei 640×480 Pixeln. Der Stereoaufbau ist orthoparallel und bleibt durch die ganze Sequenz hindurch gleich. Da es sich im Fall von Tsukuba-2012 um einen synthetischen Datensatz handelt, sind zudem verschiedene Groundtruth-Daten gegeben. So sind unter anderem für jedes Einzelbild die einzelnen Kameraposen, sowie genaue Tiefenkarten für die linke und rechte Kamera vorhanden. Die in den Tiefenkarten enthaltenen Daten beziehen sich dabei auf das

1 http://cvlab-home.blogspot.de/2012/05/h2fecha-2581457116665894170-displaynone.html
2 http://vision.middlebury.edu/stereo/

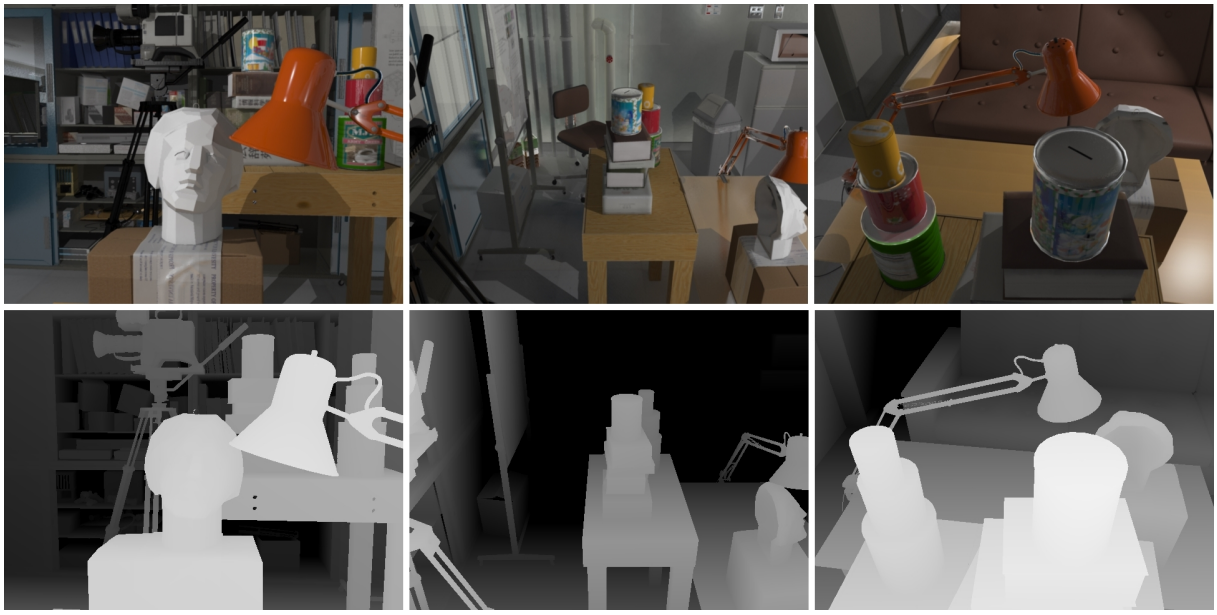

Abbildung 5.1: Referenzbilder der drei Teilsequenzen mit entsprechender Groundtruth-Tiefenkarten, die zur Auswertung mittels des neuen Tsukuba-Stereo-Datensatz verwendet werden. Von links nach Recht: **(a)** Frame 75, **(b)** Frame 300, **(c)** Frame 380.

Koordinatensystem der linken, bzw. rechten Kamera. Während in den entsprechenden Abhandlungen und im Datensatz beschrieben wird, dass sich die Kameraposen auf den Mittelpunkt des Stereoaufbaus beziehen, ist auf Nachfrage bei den Autoren bestätigt worden, dass sich die Posen auf das optische Zentrum der linken Kamera beziehen. Diese sind dabei in einem Weltkoordinatensystem, mit positiver y-Richtung nach Oben und negativer z-Richtung entlang der optischen Achse der Kameras, gegeben. Dies muss für Berechnung der Homographie berücksichtigt und die Koordinaten entsprechend um 180° um die x-Achse rotiert werden.

Der Testdatensatz eignet sich gut zur quantitativen Auswertung des Systems. So können zum einen die nicht-rektifizierten Bilder und die gegebenen Kameraposen zur Auswertung eines Plane-Sweep Verfahrens genutzt werden. Wären die Bilder bereits rektifiziert und die Kameraposen nicht gegeben, wie es beim Middlebury-Datensatz der Fall ist, könnte das Plane-Sweep-Verfahren nicht evaluiert werden. Ein weiterer Punkt, der für Tsukuba-2012 spricht ist die Existenz der genauen Tiefenkarten, womit sich die Ergebnisse gut evaluieren lassen.

Da das System für eine Rekonstruktion mittels Structure-from-Motion gedacht ist, werden nur die Aufnahmen einer einzelnen Kamera verwendet. Dies führt zu einem Nachteil dieses Datensatzes: Die Kamerafahrt ist aus vielen Rotationsbewegungen aufgebaut, welche für die Rekonstruktion nur bedingt geeignet sind, da durch sie keine Parallaxe in aufeinanderfolgenden Aufnahmen entstehen. Zur Auswertung wurden aus diesem Grund drei Teilsequenzen herausgenommen, die hauptsächlich aus translativen Bewegungen bestehen. Die Referenzbil-

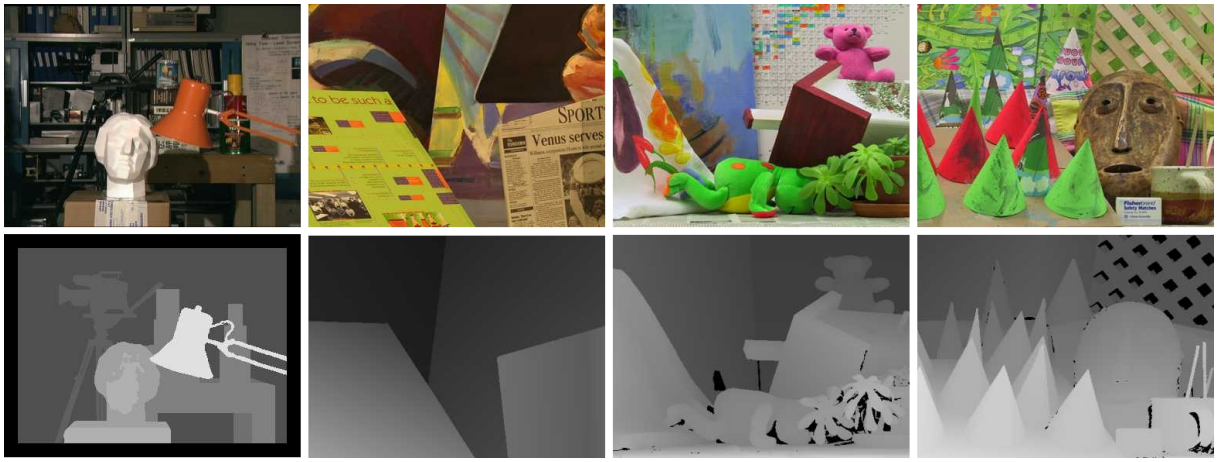

Abbildung 5.2: Verwendete Datensätze des Middlebury-Stereo-Datensatz mit dazugehörigen Groundtruth-Disparitätskarten. Von links nach rechts: **(a)** Tsukuba, **(b)** Venus, **(c)** Teddy, **(d)** Cones.

der dieser drei Teilsequenzen, sowie die dazugehörigen Groundtruth-Tiefenkarten, sind in Abbildung 5.1 dargestellt. Die durchschnittliche Tiefe der Szene zu den einzelnen Teilsequenzen ist in Tabelle 5.1 aufgelistet.

	Frame 75	Frame 300	Frame 380
Durchschnittliche Tiefe $[cm]$	186,52	256,89	161,90

Tabelle 5.1: Durchschnittliche Szenentiefe der drei Teilsequenzen des Tsukuba-2012 Datensatzes.

5.1.2 *Middlebury-Stereo-Datensatz*

Ein weiterer Datensatz, der zur Auswertung verwendet wird, ist der Middlebury-Stereo-Datensatz (Middlebury). Er enthält einen der ersten Datensätze, die zur quantitativen Evaluation und Gegenüberstellung verschiedener Verfahren genutzt wurden. Darunter auch der ursprüngliche Tsukuba-Datensatz. Die Daten, die im Middlebury-Benchmark enthalten sind, bestehen aus rektifizierten Einzelbildern, deren Posen nicht bekannt sind. Zusätzlich sind nur zu einzelnen Bildern jeder Sequenz Groundtruth-Disparitätskarten gegeben.

Da keine Kameraposen vorhanden sind, eignet sich dieser Testdatensatz nicht zur Auswertung des Plane-Sweep-Verfahrens. Er wird lediglich dazu verwendet die Ergebnisse der TGV^2-gestützten Rekonstruktion mit denen aus [19] zu vergleichen. Unter der Verwendung von Daten aus dem Middlebury-Benchmarks wird der Kostenterm in Gleichung 3.20 durch eine horizontalen Disparitätssuche zwischen den Aufnahmen gelöst. Auch hier werden nicht alle Datensätze verwendet. Lediglich die „Tsukuba“, „Venus“, „Teddy“ und „Cones“ Daten-

sätze werden herangezogen (vgl. Abb. 5.2). Die entsprechende durchschnittliche Disparität findet sich in Tabelle 5.2.

	Tsukuba	Venus	Teddy	Cones
Durchschnittliche Disparität $[px]$	10,76	8,88	13,41	16,12

Tabelle 5.2: Durchschnittliche Disparität der Szenen des Middlebury Becnhmarks.

5.2 QUALITÄTSMASS

Um die Verfahren quantitativ auswerten zu können, wird neben geeigneten Testdatensätzen mit Groundtruth-Daten auch ein geeignetes Qualitätsmaß benötigt. Damit lässt sich das Ergebnis des auszuwertenden Verfahrens mit der Groundtruth vergleichen und der Fehler der Berechnung ermitteln. Für eine solche Fehlerberechnung wird im Rahmen dieser Arbeit die *Durchschnittliche-Absolute-Abweichung* (DAA) des Ergebnisbildes ($\boldsymbol{u}^e$) zur Groundtruth ($\boldsymbol{u}^t$) verwendet. Diese errechnet sich gemäß:

$$DAA(\boldsymbol{u}^t, \boldsymbol{u}^e) = \frac{1}{w\,h} \sum_{i}^{w} \sum_{j}^{h} \left| \boldsymbol{u}^t(i,j) - \boldsymbol{u}^e(i,j) \right|. \tag{5.1}$$

Hierbei werden zunächst pixel-weise die absoluten Differenzen (Abweichung) zwischen dem Groundtruth- und dem Ergebnisbild aufsummiert. Für die Vergleichbarkeit zwischen verschiedenen Datensätzen wird die Summe im Anschluss gemittelt. Hierbei stehen w und h für die Breite bzw. Höhe der Tiefenkarte. In der Berechnung des optischen Flusses entspricht die DAA dem *Durchschnittlichen-Endpunkt-Fehler*, aus dem Englischen *Average-Endpoint-Error* (AEE), der die durchschnittliche Abweichung zwischen den Endpunkten der Vektoren des optischen Flusses angibt.

5.3 EXPERIMENTELLE ERGEBNISSE

Nach der Vorstellung der verwendeten Testdatensätze und dem Qualitätsmaß zur quantitativen Evaluation werden nun die experimentellen Ergebnisse der beiden Verfahren vorgestellt. Die Ergebnisse wurden dabei auf einer leistungsstarken Desktop-Hardware erzielt. Darin sind ein Intel Core i7 − 5820K mit 3,3 GHz, sowie 16GB Arbeitsspeicher (RAM) verbaut. Zur parallelen Berechnung wurde eine NVIDIA GeForce GTX 980 GPU mit 2048 Recheneinheiten und 4GB Gesamtspeicher verwendet.

5.3.1 *On-the-fly-Berechnung mittels Plane-Sweep*

Für die Auswertung werden zunächst die Ergebnisse des isolierten Plane-Sweep-Verfahrens zur On-the-fly-Berechnung untersucht. Wie in Kapitel 4.1 erläutert wird dieses Verfahren dazu verwendet, möglichst schnell erste Tiefenkarten berechnen zu können. Hierbei kann die Szene mit verschieden orientierten Ebenen abgetastet werden. Der Einfluss verschiedener Orientierungen auf die Qualität der Tiefenkarten wird im ersten Abschnitt dieses Kapitels evaluiert. Danach werden Ergebnisse, die durch verschiedene Kostenfunktionen und Größen der Aggregationsnachbarschaft berechnet wurden, gegenübergestellt. Als letztes wird geprüft, wie sich die Qualität der Tiefenkarten in Abhängigkeit der Anzahl von verwendeten Ebenen verhält.

5.3.1.1 *Verschiedene Ebenenorientierungen*

Die einfachste Konfiguration des Plane-Sweep-Verfahrens, gerade auch bei einem fehlenden Vorwissen über den Aufbau der zu rekonstruierenden Szene, ist die Verwendung einer einzigen frontoparallelen Ebenenorientierung bzw. Verschiebungsrichtung. Hierbei stehen die Ebenen orthogonal zur optischen Achse der Referenzkamera und damit parallel zu der entsprechenden Bildebene. Abbildung 5.3 zeigt die Ergebnisse eines solchen Plane-Sweeps mit frontoparalleler Abtastung. Die Tiefenkarten wurden für die drei verwendeten Teilsequenzen des Tsukuba-2012 Datensatzes berechnet. Jede dieser Teilsequenz besteht aus fünf Einzelbildern. Die Größe der Aggregationsnachbarschaft aller drei Berechnungen ist 11×11. Für die Parametrisierungen der Ebenen gilt: $\vec{n} = (0, 0, 1)^T$, sowie $d_{max} = 300cm$, $d_{min} = 30cm$ bei einer Schrittweite von $4cm$.

Ist die Lage und Orientierung von Objekten in der Szene bekannt, kann versucht werden die Orientierungen der Ebenen des Plane-Sweeps an die Objekte anzupassen um damit eine bessere Abtastung und Rekonstruktion zu erzielen. So ist zum Beispiel die Blickrichtung der Kamera bei Frame 300 und 380 des Datensatzes ein wenig nach unten geneigt. Unter der Annahme, dass die Platte des in der Szene vorhandenen Tisches horizontal zum Boden liegt, und mit der Kenntnis über die Orientierung der Kamera kann eine zusätzliche Ebenenorientierung hinzugenommen werden, bei der die zusätzlichen Ebenen horizontal zum Boden der Szene liegen. Um den Normalenvektor dieser Orientierung zu berechnen, wird der Vektor $(0, 1, 0)^T$, der im Weltkoordinatensystem orthogonal zum Boden steht, in das Koordinatensys-

Abbildung 5.3: Ergebnisse des isolierten Plane-Sweep-Verfahrens mit frontoparalleler Ebenenorientierung. Angewendet auf Teilsequenzen des neuen Tsukuba-Stereo-Datensatz. SAD mit Nachbarschaftsgröße: 11×11. Ebenen: $d_{max} = 300cm$, $d_{min} = 30cm$, Schrittweite $= 4cm$. Von links nach rechts: **(a)** Frame 75 mit $DAA = 15,22$; **(b)** Frame 300 mit $DAA = 32,03$; **(c)** Frame 380 mit $DAA = 25,8$.

tem der geneigten Kamera transformiert. Dadurch ergeben sich für die zusätzliche Orientierung in den Teilsequenzen 300 und 380 die Normalenvektoren $\vec{n}'_{300} = (0, 0.914, 0.405)^T$ bzw. $\vec{n}'_{380} = (0, 0.76, 0.65)^T$.

Da aufgrund des großen Wertebereichs der Tiefenkarten und der beschränkten Anzahl an Graustufen in der Visualisierung eine Auswertung von feinen Änderungen bei Betrachtung der gesamten Tiefenkarte schwierig ist, wird zur qualitativen Evaluation des zusätzlichen Normalenvektors nur ein Teil der Tiefenkarte betrachtet. Durch die in diesem Bereich geringere Anzahl an vorhandenen Tiefen kann der betrachtete Ausschnitt anders eingefärbt werden, wodurch auch kleinere Änderungen in der Tiefe besser sichtbar werden. In Abbildung 5.4 werden die Ergebnisse der Abtastung mit einem zum Boden senkrecht stehenden Normalenvektors den Ergebnissen einer frontoparallelen Abtastung gegenübergestellt. Während Ausschnitte *(a)* und *(c)* die Ergebnisse der frontoparallelen Abtastung zeigen, wird in den Ausschnitten *(b)* und *(d)* jeweils das entsprechende Ergebnis zur horizontalen Abtastung dargestellt. In dieser Gegenüberstellung ist deutlich sichtbar, dass die Tischplatte in *(b)* und *(d)* durch eine wesentlich gleichmäßigere Fläche rekonstruiert wurde. Die horizontale Abtastung führt zudem zu einem schichtweisen Aufbau der Objekte. Eine Kombination des frontoparallelen und des senkrechten Vektors würde diese Artefakte verschwinden lassen. Denn der Algorithmus des Plane-Sweep-Verfahrens verwendet immer die „Winner-Takes-It-All“ Lösung, was in den vertikalen Bereichen (Dosen und Bücher) der Tiefenkarte eher die frontoparallele Ebene bevorzugen würde.

Tabelle 5.3 enthält eine quantitative Gegenüberstellung verschiedenen Konfigurationen zur Abtastung einer Szene. Die erste Spalte enthält dabei die Ergebnisse einer frontoparallelen Abtastung. In der zweiten Spalte sind die Ergebnisse einer Kombination aus frontoparalleler und horizontaler Abtastung aufgelistet. Die dritte Spalte zeigt die Ergebnisse einer Abtastung mit fünf willkürlich gewählten Ebenenorientierungen: $\vec{n}_1 = (0,0,1)^T$, $\vec{n}_2 = (1,0,1)^T$, $\vec{n}_3 = (-1,0,1)^T$, $\vec{n}_4 = (0,1,1)^T$ und $\vec{n}_5 = (0,-1,1)^T$. Zusätzlich ist für jede Konfiguration die Laufzeit des Plane-Sweep-Verfahrens aufgelistet. Da die Tischplatte nur einen geringen

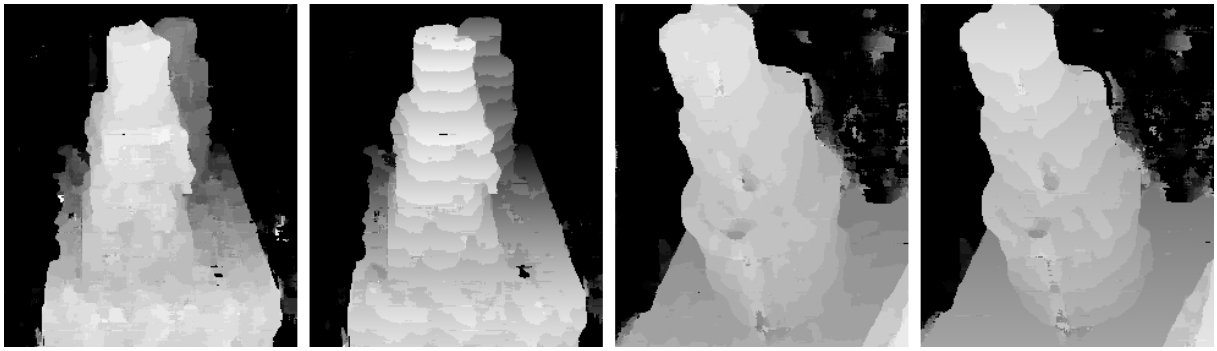

Abbildung 5.4: Gegenüberstellung zweier Ausschnitte aus den Frames 300 und 380 des Tsukuba-2012 Datensatzes, die mit jeweils einer frontoparallelen und horizontalen Ebenenorientierung abgetastet wurde. Von links nach rechts: **(a)** Frontoparallele Abtastung des Ausschnittes aus Frame 300. **(b)** Horizontale Abtastung des gleichen Ausschnittes. **(c)** Frontoparallele Abtastung des Ausschnittes aus Frame 380. **(d)** Horizontale Abtastung des Ausschnittes aus Frame 380. Eine horizontale Abtastung führt zu einer besseren Rekonstruktion der Tischplatte.

Teil der Tiefenkarte ausmacht, wirkt sich die Verbesserung jedoch nicht merklich auf die berechnete Abweichung zur Groundtruth aus. Es ist jedoch zu erkennen, dass eine zusätzliche Abtastung mit willkürlich gewählten Orientierungen (vgl. Tab. 5.3 Spalte 3) zu keiner Verbesserung in den quantitativen Ergebnissen führt. Es führt vielmehr zu einer Erhöhung der Laufzeit, da die Szenen mit weitaus mehr Ebenen abzutasten ist. Es sollte somit abgewogen werden, ob es sich lohnt verschiedene Ebenenorientierungen zu verwenden, da lediglich eine genaue Ausrichtung der Ebenen an die in der Szene enthaltenen Objekte eine Verbesserung bewirkt.

	Frame	Frontoparallel	Frontoparallel + Vertikal	Fünf Orientierungen
DAA [cm]	75	**15,22**	16,45	15,55
	300	32,03	**31,11**	31,22
	380	**25,80**	25,81	25,89
Laufzeit [s]	75	0,066	0,109	0,250
	300	0,054	0,117	0,229
	380	0,063	0,105	0,234

Tabelle 5.3: Quantitative Gegenüberstellung verschiedener Konfigurationen zu Abtastung mittels des Plane-Sweeps.

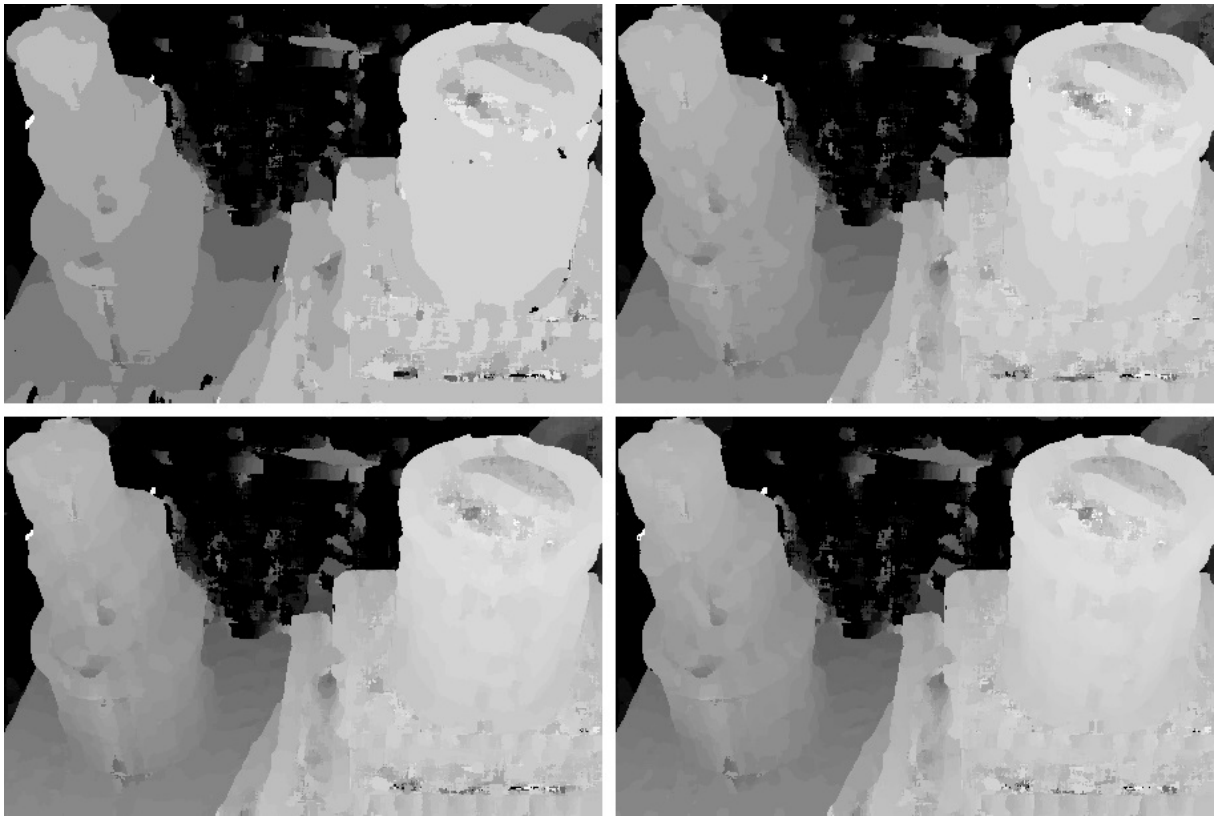

Abbildung 5.5: Gegenüberstellung der Ergebnisse verschiedener Abtastfrequenzen. Angewendet auf die Teilsequenz um Frame 380 des Tsukuba-2012 Datensatzes. Frontoparallele Abtastung. SAD mit Nachbarschaftsgröße 11×11. Von links nach rechts, von oben nach unten: **(a)** Schrittweite zwischen den einzelnen Ebenen von $10cm$. **(b)** Schrittweite von $4cm$. **(c)** Schrittweite von $2cm$. **(d)** Adaptive Schrittweite mit einer maximalen Disparitätsänderung zwischen zwei aufeinanderfolgenden Ebenen von 1.

5.3.1.2 *Auswertung zur Abtastfrequenz*

Neben Orientierung und Verschiebungsrichtung der Ebenen spielt die Abtastfrequenz bei der Rekonstruktion der Szene eine große Rolle. Werden zu wenige Ebenen verwendet, so kann die Szene möglicherweise nicht genau genug rekonstruiert werden. Bei zu vielen Ebenen wird die Berechnung deutlich langsamer. Somit ist es wichtig ein gutes Mittelmaß zwischen Qualität und Geschwindigkeit zu finden. Abbildung 5.5 zeigt zum Vergleich die Ergebnisse vier verschiedener Abtastfrequenzen angewendet auf die Teilsequenz um Frame 380. Die Tiefenkarten wurden mit einer frontoparallelen Ebenenorientierung berechnet. Als Kostenfunktion wurde die Summe der Absoluten Differenzen (SAD) mit einer Aggregationsnachbarschaft von 11×11 verwendet. Ähnlich wie beim Vergleich der unterschiedlichen Orientierungen wird nur ein Ausschnitt der Tiefenkarten betrachtet um dadurch auch geringere Unterschiede besser Visualisieren zu können.

Ausschnitt *(a)* zeigt das Ergebnis einer Abtastung, bei der die Ebenen jeweils 10*cm* voneinander entfernt sind. Deutlich zu erkennen sind hierbei die großen Abstufungen zwischen den verschiedenen Tiefen. Gerade im Bereich des linken Dosenturms sowie an der Tischplatte, kann der Tiefenverlauf nur schlecht rekonstruiert werden, da die Häufigkeit der abgetasteten Tiefen zu gering ist. Ausschnitte *(b)* und *(c)*, sind mit einer Schrittweite von 4*cm* bzw. 2*cm* zwischen den Ebenen rekonstruiert wurden. Mit zunehmender Abtastfrequenz zeigt sich, dass die Tiefenverläufe besser rekonstruiert werden können. In Abschnitt *(d)* wird die Verwendung einer adaptiven Ebenenwahl gezeigt. Wie in Kapitel 4.5 erläutert, kann je nach Anwendung und Eingangsdaten die Wahl der Ebenen in Abhängigkeit der Disparitätsänderung zwischen aufeinanderfolgenden Ebenen gewählt werden. Zwar zeigt Ausschnitt *(d)* im Vergleich zu *(c)* qualitativ keine wirklich besseren Ergebnisse, jedoch ist gerade im Verlauf der Tischplatte bei *(d)* die nicht äquidistante Abtastung zu erkennen.

Tabelle 5.4 zeigt die quantitativen Ergebnisse dieses Vergleiches. Darin zeigt sich, dass eine adaptive Ebenenwahl in Zusammenhang mit diesem Datensatz keine Verbesserung in der Qualität der Tiefenkarte bringt. Durch eine hohe Abtastfrequenz wird eine hohe Anzahl an perspektivischen Transformationen durchgeführt, was die Berechnung deutlich verlangsamt. Auch hier gilt wieder, dass mittels Kenntnis über den groben Aufbau der Szene die Ebenen und die Abtastung besser gewählt werden können. Aus den aufgelisteten Ergebnissen geht hervor, dass im Zusammenhang mit dem Tsukuba-2012 Datensatz eine Schrittweite von 4*cm* zwischen den einzelnen Ebenen ein gutes Verhältnis zwischen Genauigkeit und Laufzeit bietet.

	Frame	Schrittw. = 10*cm*	Schrittw. = 4*cm*	Schrittw. = 2*cm*	adaptiv
DAA [cm]	75	15,53	15,22	**15,10**	15,23
	300	32,06	**32,03**	32,11	32,11
	380	26,44	**25,79**	**25,79**	25,93
#Ebenen	75	27	68	136	210
	300	27	68	136	245
	380	27	68	136	158
Laufzeit[s]	75	0,031	0,063	0,094	0,156
	300	0,031	0,047	0,094	0,203
	380	0,031	0,062	0,094	0,140

Tabelle 5.4: Quantitativer Vergleich verschiedener Abtastfrequenzen bei einer frontoparallelen Abtastung, angewendet auf den Tsukuba-2012 Datensatz.

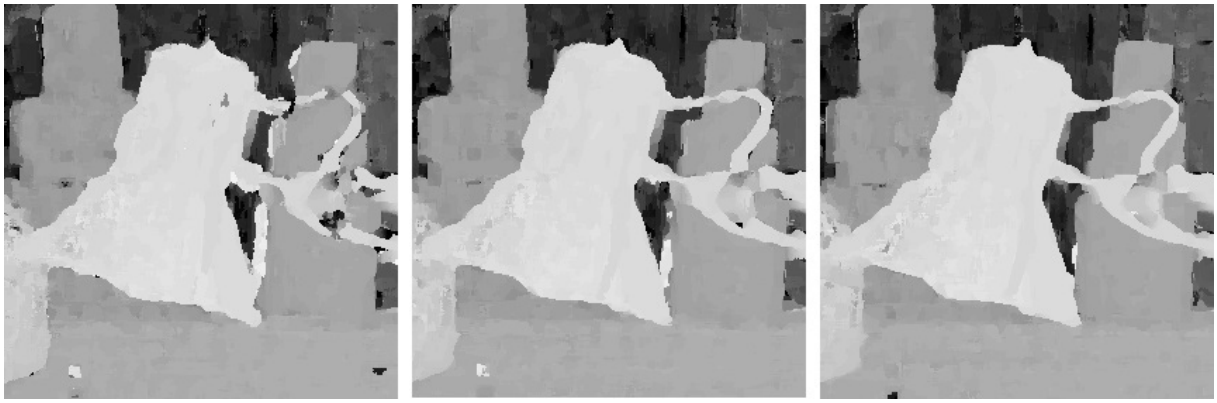

Abbildung 5.6: Auswirkungen verschiedener Anzahl an Eingangsbildern. Erkennbare Verbesserung der Tiefenschätzung mit zunehmender Anzahl an Eingangsdaten. Von links nach rechts: **(a)** Verwendung einer Sequenz bestehend aus 3 Einzelbildern. **(b)** Verwendung von 5 Eingangsbilder. **(c)** Betrachtung von 11 Einzelbildern zur Rekonstruktion.

5.3.1.3 *Behandlung von Verdeckungen durch mehrere Aufnahmen*

Wie in Kapitel 4.3 beschrieben kann die Verwendung von mehreren Aufnahmen bei der 3D-Rekonstruktion genutzt werden um besser mit der Verdeckungsproblematik umzugehen. Dabei wird für jedes Pixel immer nur eine Teilgruppe der Aufnahmen verwendet, um die Tiefe zu schätzen. Dies beruht auf der Annahme, dass Pixel, die in einem Teil der Vergleichsaufnahmen verdeckt sind, in der anderen Teilgruppe sichtbar sind. Um die Auswirkung der Anzahl von verwendeten Aufnahmen zu untersuchen, wird in Abbildung 5.6 ein Ausschnitt verglichen, der mit drei verschiedenen Konfigurationen an Eingangsbildern rekonstruiert wurde. Der betrachtete Ausschnitt ist dabei Teil der Tiefenkarte zu Frame 75 des Tsukuba-2012 Datensatzes. Er enthält drei Szenenobjekte, die in verschiedener Tiefe liegen und dabei die dahinterliegenden Objekte verdecken.

Ausschnitt *(a)* ist mit 3 Eingangsbildern rekonstruiert worden. Diese setzen sich aus der Referenzaufnahme und jeweils einem zeitlichen Vorgänger und Nachfolger zusammen. Der zweite Ausschnitt *(b)* wurde aus 5 Eingangsbilder berechnet, während bei *(c)* 11 Aufnahmen verwendet wurden. Alle drei Ausschnitte wurden dabei mit einer frontoparallelen Ebenenorientierung und einer Schrittweite von $4cm$ zwischen den einzelnen Ebenen aufgebaut. Auch hier wurde die Summe der Absoluten Differenzen (SAD) mit einer Aggregationsnachbarschaft von 11×11 zur Berechnung der Kosten verwendet. In den Gegenüberstellungen aus Abbildung 5.6 ist zu erkennen, dass die Rekonstruktion mit zunehmender Anzahl von Eingangsbildern an Qualität gewinnt. Gerade im Bereich des Lampenkabels und des Lampenarms sind deutliche Verbesserungen zu erkennen. Zudem verschwinden die weißen Bereiche im Hintergrund, welche durch Verdeckungen hervorgerufen werden. Diese Verbesserung geht auch aus den Zahlen in Tabelle 5.5 hervor. Dabei ist hervorzuheben, dass die Verwendung einer höheren Anzahl an Eingangsbilder einen nicht so hohen Einfluss auf die Laufzeit des Verfahrens hat wie die Erhöhung der Abtastfrequenz (vgl. Tab. 5.4). Gleichzeitig wird jedoch eine

größere Qualitätssteigerung erzielt. Die Verwendung von 11 Aufnahmen liefert in allen drei Teilsequenzen den bisher geringste Fehler zu einer durchaus passablen Laufzeit.

	Frame	#Frames = 3	#Frames = 5	#Frames = 11
DAA [*cm*]	75	16,00	14,89	**14,36**
	300	38,34	32,31	**31,24**
	380	26,08	26,23	**23,02**
Laufzeit [*s*]	75	0,031	0,047	0,078
	300	0,031	0,047	0,078
	380	0,044	0,053	0,094

Tabelle 5.5: Quantitativer Vergleich der Anzahl an verwendeten Eingangsbildern.

5.3.1.4 *Kostenfunktionen und Aggregationsnachbarschaften*

In Kapitel 2.3 werden zwei verschiede Kostenfunktionen vorgestellt, die zum Abgleich von Bildstrukturen genutzt und mit deren Hilfe Punktkorrespondenzen zwischen Bildern gefunden werden können. Die simpelste der beiden Kostenfunktionen ist die Summe der absoluten Differenzen (SAD), bei der die Intensitäten der einzelnen Pixel direkt verglichen werden. Die zweite vorgestellte Kostenfunktion ist die Hammingdistanz der Census-Transformation (CT). Durch die CT werden die Pixel in eine Bitfolge konvertiert die das Verhältnis der Intensitäten innerhalb einer Nachbarschaft beschreibt. Die eigentlichen Kosten ergeben sich im Zusammenhang der CT durch die Hammingdistanz, welche die Anzahl der unterschiedlichen Bits zweier zu vergleichenden Bitfolgen angibt. Zur Robustifizierung der Kostenfunktionen werden die Kosten innerhalb einer Nachbarschaft um den betrachteten Pixel aufsummiert. Dies führt zur Reduktion von Mehrdeutigkeiten in der Suche nach Punktkorrespondenzen, da der Vergleich einzelner Pixelnachbarschaften aussagekräftiger ist, als der direkte Vergleich zwischen den Pixeln.

In Abbildung 5.7 sind die Ergebnisse verschiedener Größen der Aggregationsnachbarschaft für die beiden Kostenfunktionen gegenübergestellt. Aufgrund der Erkenntnisse aus den vorherigen Kapiteln werden die Tiefenkarten mit einer frontoparallelen Ebenenorientierung, einer Schrittweite zwischen den Ebenen von $4cm$ und einer Anzahl von 11 Eingangsbilder berechnet. Die Ergebnisse der oberen Reihe in Abbildung 5.7 wurden mit der SAD berechnet, während in der zweiten Reihe die Hammingdistanz der CT als Kostenfunktion verwendet wurde. Für den Aufbau der CT ist hierbei ein Größe von 9×7 Pixel gewählt. Dies ergibt für die Länge der jeweiligen Bitfolgen 63, was der maximalen Größe entspricht, die in einem 64-Bit großen `long`-Datentyp gespeichert werden kann. Die Tiefenkarten in der ersten Spalte wurden mit einer Nachbarschaftsgröße von 7×7 erstellt, die zweite Spalte enthält die Ergebnisse einer 11×11 großen Nachbarschaft, und die dritte Spalte zeigt die Resultate einer

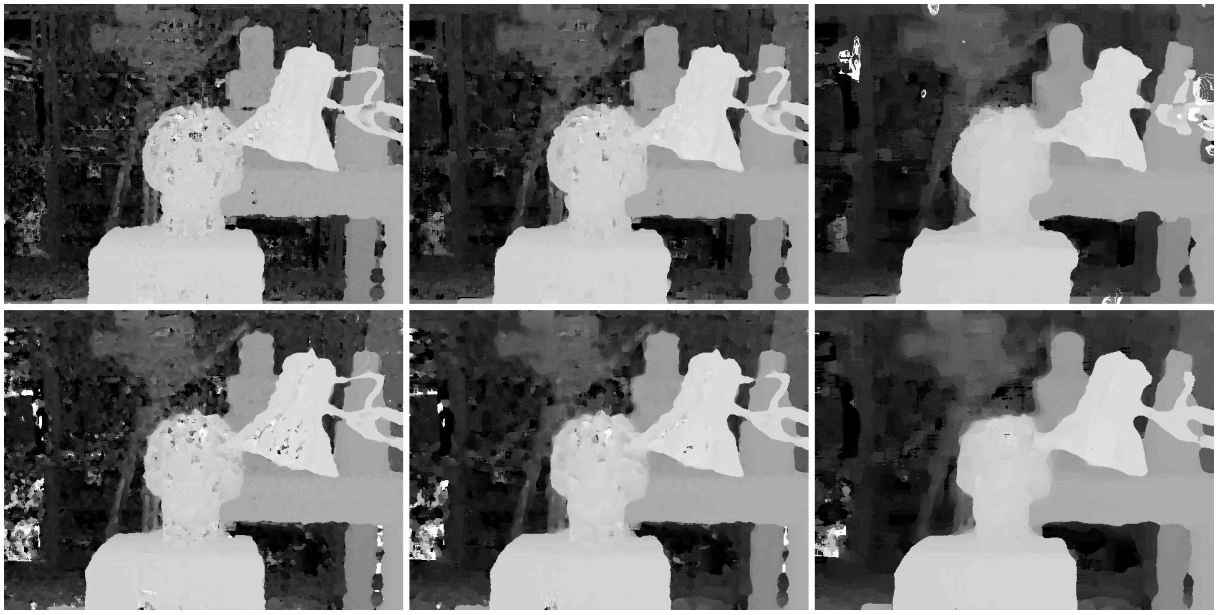

Abbildung 5.7: Vergleich verschiedener Nachbarschaftsgrößen unter Verwendung der zwei Kostenfunktionen. Von links nach rechts, von oben nach unten: **(a)** SAD bei einer Nachbarschaft von 7×7, **(b)** SAD bei einer Nachbarschaft von 11×11, **(c)** SAD bei einer Nachbarschaft von 21×21, **(d)** CT bei einer Nachbarschaft von 7×7, **(e)** CT bei einer Nachbarschaft von 11×11, **(f)** CT bei einer Nachbarschaft von 21×21.

Aggregationsnachbarschaft der Größe 21×21. Zusätzlich listet Tabelle 5.6 die Abweichungen zu den Groundtruth-Daten auf.

Aus diesen Ergebnissen können verschiedene Schlüsse gezogen werden. Zum einen wird deutlich, dass das Rauschen in den Tiefenkarten wie erwartet mit zunehmender Nachbarschaftsgröße kleiner wird. Dies ist besonders im Hintergrund und am oberen Rand der Büste zu erkennen. Des Weiteren zeigt die Entwicklung in Abbildung 5.7, dass durch eine größere Aggregationsnachbarschaft kleinere Strukturen, die komplett innerhalb der Nachbarschaft liegen, verloren gehen. So ist z. B. das Kabel der Lampe in *(a)* und *(d)* noch vollständig, während es in den restlichen Tiefenkarten durchbrochen wird. Zusätzlich wird sichtbar, dass eine große Nachbarschaft nicht nur negative Auswirkungen auf das Vorhandensein von kleinen Strukturen hat, sondern auch auf die Genauigkeit von Objektgrenzen. So ist zwar in den Tiefenkarten *(c)* und *(f)* sehr wenig Rauschen zu erkennen, was auf wenig Mehrdeutigkeiten in der Korrespondenzsuche hinweist, jedoch wird die Form der Objekte nicht gut rekonstruiert. Gerade die Struktur des Lampenarms geht stark verloren.

Eine weitere Erkenntnis gibt der Vergleich zwischen den beiden Kostenfunktionen. Wie in Kapitel 2.3 beschrieben soll die Hammingdistanz der CT durch die alleinige Betrachtung der relativen Anordnung zwischen den Pixeln als Kostenfunktion robuster gegenüber Beleuchtungsänderungen sein. Zwar wird dies in Abbildung 5.7 nicht untersucht, jedoch geht aus den gezeigten Ergebnissen hervor, dass bei der Hammingdistanz der CT mehr Fehlzuord-

nungen auftreten, als bei der SAD. So zeigen die Ergebnisse, dass die Objektstrukturen (z. B. Lampenarm) durch die CT-basierte Kostenfunktion zwar besser rekonstruiert werden, aber im Hintergrund und teilweise auch innerhalb von Objekten (z. B. Lampenschirm) starke Ausreißer auftreten. Gleichzeitig gilt, dass während die Ergebnisse basierend auf der SAD Kostenfunktion mit zunehmender Nachbarschaftsgröße wieder schlechter werden, die Ergebnisse unter der Verwendung der CT und einer Nachbarschaftsgröße von 21×21 die geringste DAA haben. Dies wird auch durch die Zahlen in Tabelle 5.6 belegt.

Das gehäufte Auftreten von Mehrdeutigkeiten bei einer CT-basierten Kostenfunktion mit geringer Nachbarschaftsgröße ist auf die geringere Aussagekraft der CT zurückzuführen. Während die Absolute Differenz zwischen zwei Pixeln einen maximalen Wertebereich von 255 hat, entspricht dieser bei einer Hammingdistanz der CT der Länge der Pixel-gebundenen Bitfolgen. Diese beträgt bei den oben gezeigten Ergebnissen 63. Somit lassen sich durch die CT deutlich weniger Variationen in der Bildstruktur ausdrücken. Erst durch das Vergrößern der Aggregationsnachbarschaft wird die Hammingdistanz benachbarter Pixel gemittelt, wodurch weniger Ausreißer auftreten. Zwar könnte auch die Länge der Bitfolgen erhöht werden, jedoch müssten diese dann in mehreren Variablen abgespeichert werden. Aufgrund der geringeren Aussagekraft der CT wird diese häufiger in Zusammenhang mit Variationsansätzen genutzt, da die Mehrdeutigkeiten durch die zusätzliche Regularisierung weiter reduziert werden können.

Zusammenfassend geht aus den präsentierten Ergebnissen zu den Kostenfunktionen und der Aggregationsnachbarschaft hervor, dass bei der SAD die besten Ergebnisse mit einer 11×11 großen Aggregationsnachbarschaft erzielt wird. Bei der CT-basierten Kostenfunktion werden die quantitativ besten Ergebnisse bei einer Nachbarschaftsgröße von 21×21 erzielt. Der quantitative Fehler dieser Ergebnisse unterschreitet dabei sogar den Fehler der SAD mit einer 11×11 großen Aggregationsnachbarschaft. Jedoch ist die Laufzeit bei einer Berechnung mittels einer CT-basierten Kostenfunktion teilweise doppelt so hoch wie bei einer Verwendung der SAD. Zudem werden qualitativ die besten Ergebnisse im Zusammenhang des Plane-Sweep-Verfahrens, welches auf den Frame 75 des Tsukuba-2012 Datensatzes angewendet wurde, mit der SAD als Kostenfunktion erzielt.

	Kostenfunktion	7×7	11×11	21×21
DAA [cm]	SAD	15,38	**14,35**	16,14
	CT	16,40	14,86	**14,07**
Laufzeit [s]	SAD	0,078	0,094	0,11
	CT	0,172	0,187	0,203

Tabelle 5.6: Quantitativer Vergleich der beiden Kostenfunktionen mit verschiedenen Größen der Aggregationsnachbarschaft. Angewendet auf Frame 75 des Tsukuba-2012 Datensatzes.

5.3.2 *Offline-Berechnung mittels TGV^2*

Nach der Vorstellung einiger experimentellen Ergebnisse des isolierten Plane-Sweep-Verfahrens, werden nun Ergebnisse der TGV^2-gestützten Rekonstruktion betrachtet. Wie in Kapitel 4.1 erläutert, nutzt das TGV^2-gestützte Verfahren das Ergebnis der Plane-Sweep-Rekonstruktion zur Initialisierung. Es wird erhofft, dass das TGV^2-Verfahren genauere Tiefenkarten erzielt. Die Berechnung wird dabei jedoch deutlich länger brauchen als beim isolierten Plane-Sweep-Verfahren. In der Vorstellung der Ergebnisse wird zunächst auf die verschiedenen Gewichtstensoren im Regularisierungsterm eingegangen. Anschließend werden die verschiedenen Kostenfunktionen und die Adaptive Aggregationsnachbarschaft im Datenterm untersucht. Zum Schluss wird geprüft wie sich eine Veränderung der Berechnungsiterationen auf die Qualität der Ergebnisse auswirkt.

5.3.2.1 *Gewichtstensoren*

Die im Zusammenhang mit der TGV^2-gestützten Rekonstruktion auftretenden Regularisierung soll gemäß Gleichung 3.13 durch einen anisotropen Gewichtstensor an die lokale Bildstrukturen angepasst werden. Das Ziel ist dabei, die Stärke der Regularisierung an Objektgrenzen zu reduzieren. In Kapitel 4.2 werden verschiedene Tensoren zur lokalen Gewichtung der Regularisierung vorgestellt. Diese werden dabei durch eine Gewichtsfunktion in Abhängigkeit des lokalen Bildgradienten aufgebaut. Die Funktionskurven sind in Abbildungen 4.2 & 4.4 abgebildet. Die verschiedenen Parameter der Funktionen erlauben es den Gewichtungsfaktor an den Bildgradienten anzupassen. Bei dem Perona-Malik- und Charbonnier-Term gibt λ_{pm} bzw. λ_{ch} beispielsweise an, ab welchem Betrag des Gradienten eine Vorwärtsdiffusion (Glättung) oder eine Rückwärtsdiffusion (Kantenanhebung) stattfindet.

Um die Wahl der Funktionsparameter richtig zu treffen, sollte zunächst untersucht werden, wie groß der Gradienten an verschieden Kanten von Bildstrukturen ist, an denen die Regularisierung ausgesetzt werden soll. In Abbildung 5.8 werden exemplarisch drei Bildausschnitte, sowohl in Farbe als auch in Graustufen, gegenübergestellt und deren betraglich maximaler Bildgradient verglichen. Während der Gradient im Graustufenbild direkt aus den Intensitätswerten berechnet wird, ergibt sich dieser im Farbbild aus den durchschnittlichen Gradienten der einzelnen Farbkanäle. In 5.8 *(a)* wird der Bildgradient an einem Übergang von sehr hell zu sehr dunkel gemessen. Hier ist der Bildgradient sowohl im Farb- und Graustufenbild etwa gleich und beträgt ungefähr 47 bzw. 46. Bei dem in *(b)* dargestellten Bildausschnitt wird deutlich, dass es einen Unterschied macht, ob der Bildgradient basierend auf einem Farb- oder Graustufenbild aufgebaut wird. Der braune Farbton des Tisches nimmt im Graustufenbild den ähnlichen Grauwert der Büste an, wodurch die Länge des maximalen Bildgradienten im Graustufenbild von ~6 auf ~3,5 um fast die Hälfte sinkt. In Ausschnitt *(c)* ist die Länge des Gradienten basierend auf dem Graustufenbild sogar ein wenig größer. Aus diesem Vergleich geht hervor, dass die Benutzung von Farbbildern zum Aufbau der Gewichtstensoren besseren Aufschluss über lokale Bildgradienten gibt.

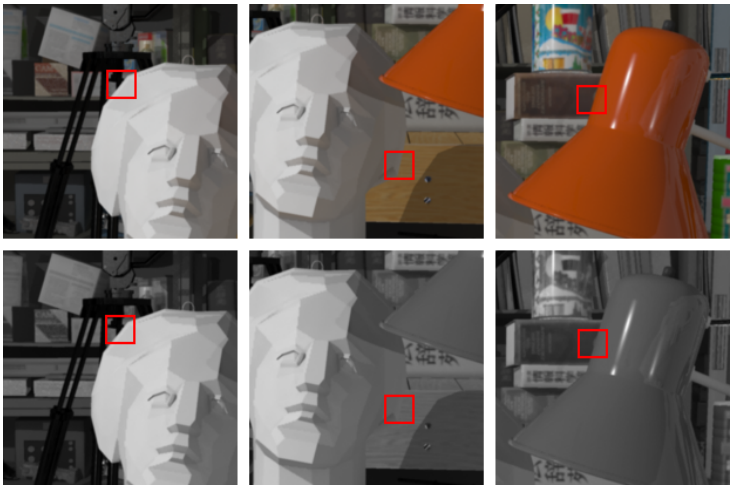

Abbildung 5.8: Drei Ausschnitte (rot umrahmt) in denen der Bildgradient an Objektkanten untersucht wird. Wertebereich der Pixelintensitäten: [0, 255]. Obere Reihe: Kombinierter Farbgradient. Untere Reihe: Einfacher Bildgradient basierend auf Graustufenbilder. Von links nach rechts: **(a)** Betraglich maximaler Bildgradient bei Farbbild ~47 und bei Graustufenbild ~46. **(b)** Maximaler Farbgradient: ~6, Maximaler Graustufengradient: ~3,5. **(c)** Maximaler Farbgradient: ~5.5, Maximaler Graustufengradient ~6

Mit dieser Erkenntnis über die Länge von auftretenden Bildgradienten können nun die Parameter der Gewichtsfunktionen entsprechend gewählt werden. Wie aus Abbildung 4.3 ersichtlich ist, fällt die Kurve des Perona-Malik-Terms schneller ab als die des Charbonnier-Terms was in der Parameterwahl der Gewichtsfunktionen berücksichtigt werden muss. Aber selbst wenn λ_{ch} des Charbonnier-Terms halb so groß gewählt wird wie λ_{pm}, fällt die Kurve des Charbonnier-Terms bei $|\nabla I| > \lambda_{ch}$ nicht so schnell ab wie die des Perona-Malik-Terms (vgl. blaue und gelbe Kurve in Abb. 4.3). Eine weitere in Kapitel 4.2.3 eingeführte Gewichtsfunktion ist die Exponentialfunktion. Zwar ähneln die Formen der Kurven des Charbonnier- und Perona-Malik-Terms der einer Exponentialfunktion, jedoch bietet die allgemeine e-Funktion durch die Parameter α und β eine größere Flexibilität in der Wahl der Kurvenform.

In Abbildung 5.9 sind Ergebnisse einer linearen Diffusion abgebildet, die mit verschiedenen bildbasierten Diffusionstensoren auf einen Ausschnitt des Frame 75 aus dem Tsukuba-2012 Datensatzes angewendet wurde. Hierbei werden die Tensoren basierend auf den Gradient des Farbbildes aufgebaut. Die Ergebnisse werden immer in Paaren angezeigt. Während in den Reihen 1, 3, und 5 die Ergebnisse einer isotropen Diffusion dargestellt sind, wurden die Ausschnitte in den anderen Reihen mit anisotropen Gewichtstensoren erzeugt. Das Paar *(a)* zeigt den Originalausschnitt, sowie eine lineare homogene Diffusion, bei der keine Gewichtsanpassung an die Bildstrukturen vorgenommen wurde. Abschnitte *(b)*, *(c)* und *(d)* zeigen die Ergebnisse einer isotropen und anisotropen Diffusion mittels des Perona-Malik-Terms mit $\lambda_{pm} = 6$, $\lambda_{pm} = 10$ und $\lambda_{pm} = 50$. In *(e)*, *(f)* und *(g)* ist ein Charbonnier-Term mit $\lambda_{ch} = 6$, $\lambda_{ch} = 10$ und $\lambda_{ch} = 50$ verwendet worden. Bei der Diffusion mittels des Charbonnier-Terms

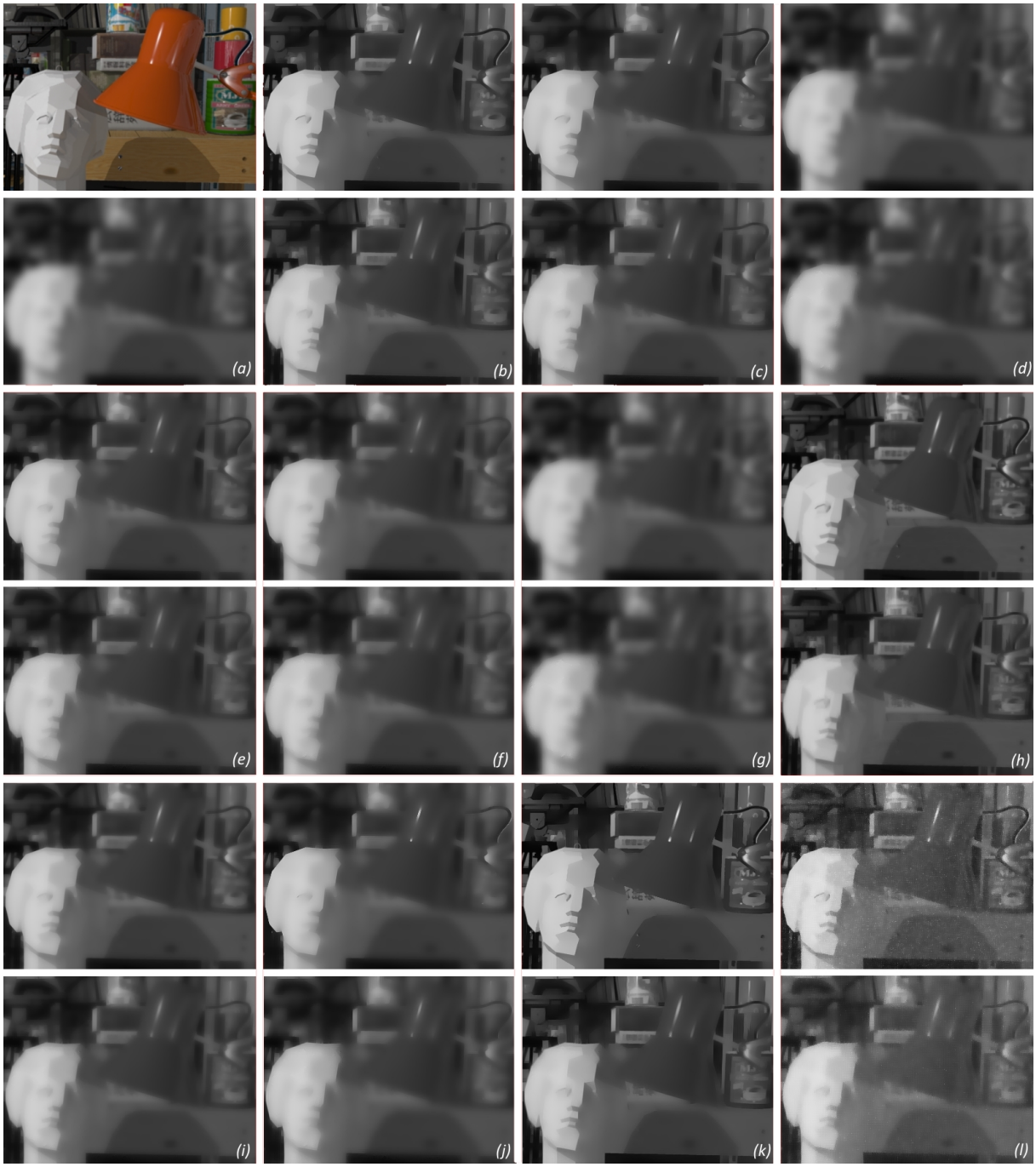

Abbildung 5.9: Lineare isotrope und anisotrope Diffusion mit verschiedenen Diffusionstensoren auf einen Ausschnitt des Frame 75 aus dem Tsukuba-2012 Datensatz. **(a)** Originalausschnitt mit homogener linearen Diffusion. **(b)** Perona-Malik-Term mit $\lambda_{pm} = 6$. **(c)** Perona-Malik-Term mit $\lambda_{pm} = 10$. **(d)** Perona-Malik-Term mit $\lambda_{pm} = 50$. **(e)** Charbonnier-Term mit $\lambda_{ch} = 6$. **(f)** Charbonnier-Term mit $\lambda_{ch} = 10$. **(g)** Charbonnier-Term mit $\lambda_{ch} = 50$. **(h)** Exponentialfunktion mit $\alpha = 5, \beta = 0,2$. **(i)** Exponentialfunktion mit $\alpha = 5, \beta = 0,5$. **(j)** Exponentialfunktion mit $\alpha = 100, \beta = 2$. **(k)** Exponentialfunktion mit $\alpha = 30000, \beta = 3$. **(l)** Anwendung einer isotropen und anisotropen linearen Diffusion auf ein verrauschtes Bild. Gaußsches Rauschen mit $\sigma = 50$. Diffusionstensor basierend auf Perona-Malik-Term mit $\lambda_{pm} = 6$.

ist im Vergleich zu den Ergebnissen des Perona-Malik-Terms eine deutlich stärkere Diffusion über die Objektkanten hinweg zu erkennen. Gerade das Kabel der Lampe wird beim Charbonnier-Term mehr geglättet als beim Perona-Malik-Term. Es ist auch ein kleiner Unterschied zwischen der isotropen und anisotropen Diffusion zu erkennen. So wird z. B. das Auge bei einem anisotropen Tensor ein wenig in horizontaler Richtung entlang der Kante geglättet. Des Weiteren ist sowohl bei der Diffusion mittels dem Perona-Malik-Terms als auch mittels dem Charbonnier-Terms deutlich zu erkennen, wie mit zunehmendem λ immer mehr Kanten geglättet werden. So wird der linke Rand der Büste aufgrund des hohen Bildgradienten selbst bei $\lambda_{pm} = \lambda_{ch} = 50$ noch ein wenig hervorgehoben.

In den Ausschnitten *(h)*, *(i)*, *(j)* und *(k)* der Abbildung 5.9 sind die Ergebnisse der isotropen und anisotropen Diffusion mit der Exponentialfunktion als Gewichtungsfaktor (vgl. Gl. 4.8) abgebildet. Hierbei enthält *(h)* das Ergebnis einer sehr spitzen *e*-Funktion mit $\alpha = 5$ und $\beta = 0,2$. Während im isotropen Fall von *(h)* wenig Glättung stattfindet, sind im anisotropen Ergebnis ein paar Kanten diffundiert. Die *e*-Funktion in *(i)* ist mit $\alpha = 5$ und $\beta = 0,5$ den Kurven des Perona-Malik- und des Charbonnier-Terms sehr ähnlich. Jedoch ist die *e*-Funktion im Vergleich zu den anderen beiden Termen deutlich steiler und glättet dadurch über weniger Kanten hinweg. Diese Parametrisierung ist in [50] verwendet worden und da Kuschk und Cremers in [19] keine Angaben zu der Parametrisierung ihrer Umsetzung machen, wird angenommen, dass diese Parameterwahl ebenfalls in [19] verwendet wird. Die Form der in *(j)* verwendeten Kurve fällt deutlich flacher ab, wodurch im Vergleich zu *(h)* über mehr Kanten hinweg diffundiert wird. Als Gegenbeispiel hierzu dienen die Ausschnitte in *(k)*, in denen sehr viele Kanten erhalten sind. Die hier verwendete *e*-Funktion hat einen sehr steileren Funktionsverlauf. Dadurch wird die Diffusion an vielen Karten schlagartig ausgesetzt und somit kommen die Kanten noch präziser hervor.

In den letzten Ausschnitten *(l)* von Abbildung 5.9 wird noch einmal der Vorteil eines anisotropen Gewichtstensors bei der Existenz von Rauschen deutlich. Während nach der isotropen linearen Diffusion noch Rauschen vorhanden ist, konnte der anisotrope lineare Diffusionstensor gut über ein Großteil des Rauschens hinweg glätten.

In der bisherigen Analyse der verschiedenen Gewichtstensoren wurde deren Wirkungsweise lediglich als Teil einer linearen Diffusion betrachtet. Im Folgenden soll nun untersucht werden, wie sich die Tensoren zur adaptiven Regularisierung als Teil der TGV^2-gestützten Rekonstruktion verhalten. Hierzu werden die globalen Gewichtungen λ_s und λ_d aus Gleichung 3.13 zunächst beide auf $1,0$ gesetzt, wodurch dem Regularisierungsterm und Datenterm der gleiche Einfluss gegeben wird. Die Gewichtung λ_a ist im Rahmen dieser Arbeit gemäß [19] fest auf $8\lambda_s$ fixiert. Die Iterationen sowie die Schrittweiten des Gradientenabstiegs bzw. -anstiegs sind ebenfalls zunächst gemäß der in [19] vorgestellten Werte gewählt: Hierbei beträgt die Anzahl der globalen Iterationen $globalItr = 80$, die Anzahl der Primal-Dual-Iterationen $smoothItr = 150$, sowie $\tau_u = \tau_p = 1/\sqrt{12}$ und $\tau_v = \tau_q = 1/\sqrt{8}$. Zur Lösung der Kostenfunktion und zum Abgleich von Bildstrukturen, wird eine frontoparalleler Plane-Sweep mit der Summe der Absoluten Differenzen (SAD) als Kostenfunktion durchgeführt. Die Schrittweite

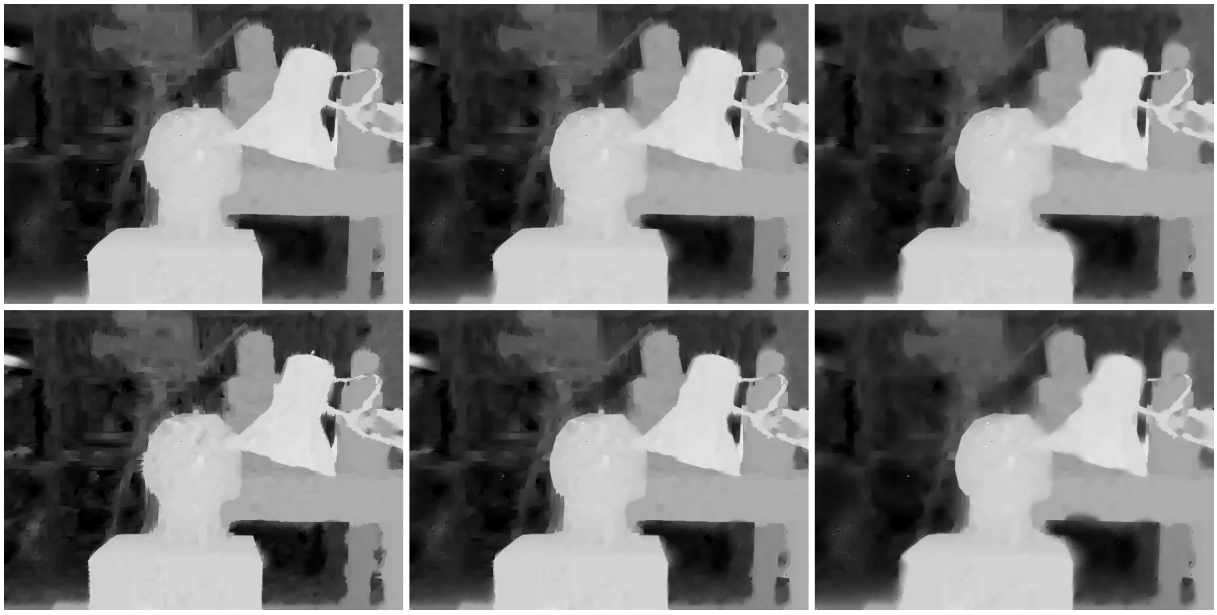

Abbildung 5.10: Vergleich verschiedener anisotroper Gewichtstensoren im Zusammenhang der TGV^2-gestützten Rekonstruktion. $\lambda_s = 1$ & $\lambda_d = 1$. Von links nach rechts, von oben nach unten: **(a)** Anisotroper Tensor basierend auf den Perona-Malik-Term mit $\lambda_{pm} = 3$, **(b)** $\lambda_{pm} = 6$, **(c)** $\lambda_{pm} = 10$. **(d)** Anisotroper Tensor basierend auf der e-Funktion mit $\alpha_{exp} = 5$ & $\beta_{exp} = 0.2$, **(e)** $\alpha_{exp} = 5$ & $\beta_{exp} = 0.3$, **(f)** $\alpha_{exp} = 5$ & $\beta_{exp} = 0.5$.

der Ebenen beträgt dabei $4cm$. Um der Kürze willen werden im Folgenden nur noch anisotrope Gewichtstensoren betrachtet, da aus der vorangegangen Diskussion und den in Abbildung 5.9 dargestellten Ergebnissen ohnehin hervorgeht, dass eine anisotrope Gewichtung bzgl. der Kantenerhaltung bessere Ergebnisse liefert.

Abbildung 5.10 sowie Tabelle 5.7 zeigen die Ergebnisse der TGV^2-gestützten Rekonstruktion unter Verwendung verschiedener Tensoren. Auch hier gilt, dass lediglich ein paar Konfigurationen evaluiert werden um die Auswirkungen der verschiedenen Tensoren zu erläutern. In Ausschnitt *(a)* von Abbildung 5.10 ist ein anisotroper Gewichtstensor basierend auf dem Perona-Malik-Term mit $\lambda_{pm} = 3$ verwendet worden. Die quantitative Auswertung in Tabelle 5.7 zeigt, dass diese Konfiguration des Tensors eine Tiefenkarte mit der geringsten Abweichung berechnet. Auch qualitativ ist zu erkennen, dass die Kanten präzise erhalten bleiben und die Objektstrukturen gut rekonstruiert werden. Mit zunehmendem λ_{pm} (vgl. Ausschnitte *(b)* und *(c)*) wird das Ergebnis insgesamt glatter. Zudem verschwimmen auch erwartungsgemäß deutlich mehr Kanten. Jedoch ist gerade der Hintergrund in *(c)* qualitativ besser rekonstruiert, da er insgesamt homogener ist. Die Zahlen aus Reihe 4 in Tabelle 5.7 bestätigen den flacheren Funktionsverlauf des Charbonnier-Terms. Die quantitativen Ergebnisse für $\lambda_{ch} = 3$ ähneln denen des Perona-Malik-Terms mit $\lambda_{pm} = 6$.

In der unteren Reihe der Abbildung 5.10 sind die Ergebnisse eines anisotropen Gewichtstensors mit der allgemeinen Exponentialfunktion abgebildet. Die e-Funktion fällt bekanntlich

stärker ab, als die Funktionen des Perona-Malik- und Charbonnier-Terms. Dadurch wird die Gewichtung schlagartiger ausgesetzt, wodurch schärferen Kanten entstehen. Während eine anisotrope Gewichtung mit $\alpha_{exp} = 5$ und $\beta_{exp} = 0,2$ vielleicht zu strikt ist, liefert die Gewichtung mit $\alpha_{exp} = 5$ und $\beta_{exp} = 0.3$ die quantitativ besten Ergebnisse der Tensoren, die auf der Exponentialfunktion basieren. Eine e-Funktion mit $\alpha_{exp} = 5$ und $\beta_{exp} = 0.5$ (vgl. Abb. *(f)*), welche vermutlich auch in [19] verwendet wurde, erzielt insgesamt schlechtere Ergebnisse. Zwar sind die Bereiche innerhalb von Szenenobjekten teilweise homogener, jedoch wird über wesentlich mehr Kanten hinweg regularisiert. So ist z. B. die Unterkannte des Tisches in *(f)* nicht präzise erhalten.

Tensoren	Frame 75	Frame 300	Frame 380
Perona-Malik $\lambda_{pm} = 3$	**12,56**	**30,34**	21,38
Perona-Malik $\lambda_{pm} = 6$	12,75	30,53	21,97
Perona-Malik $\lambda_{pm} = 10$	13,04	31,06	22,39
Charbonnier $\lambda_{ch} = 3$	12,87	30,68	21,95
Exponential $\alpha_{exp} = 5, \beta_{exp} = 0,2$	12,76	30,77	21,72
Exponential $\alpha_{exp} = 5, \beta_{exp} = 0,3$	12,66	30,59	**21,31**
Exponential $\alpha_{exp} = 5, \beta_{exp} = 0,5$	13,48	30,93	22,05

Tabelle 5.7: Quantitativer Vergleich der Auswirkung verschiedener Gewichtstensoren auf das Ergebnis der TGV^2-gestützten Rekonstruktion. Da die Veränderung der Tensoren keine Auswirkung auf die Laufzeit hat, wird diese hier nicht aufgelistet.

Während bei einem strikten anisotropen Gewichtstensor die Objektkanten gut erhalten werden, führt ein nachgiebigerer Tensor zu einer Fragmentierung mancher Kanten (vgl. Ausschnitte *(c)* und *(f)*). In [19] werden die Gewichtstensoren durch die Detektion von Liniensegmenten erweitert. Die zusätzliche Verwendung des Liniensegment-Detektors (LSD) soll dabei die Tensoren besser an gerade Objektkanten anpassen. Sie wird auch im Rahmen dieser Arbeit adaptiert und könnte dazu führen, dass bei einer nachgiebigeren Gewichtsfunktion die Tiefendiskontinuitäten an Objektkanten besser erhalten werden. Im nachfolgenden Kapitel wird die Auswirkung der zusätzlichen Integration des LSDs für auf die Ergebnisse von Tabelle 5.7 evaluiert.

5.3.2.2 *Gewichtstensoren mit LSD*

Wie in Abbildungen 5.9 und 5.10 bereits ersichtlich ist, führt die lokale Gewichtung mittels isotroper und anisotroper Gewichtstensoren dazu, dass die Regularisierung an Objektgrenzen reduziert wird. Dadurch werden Objektkanten hervorgehoben. Je nach Wahl der Tensoren sowie deren Parametrisierung werden dabei weniger oder mehr Kanten erhalten. Um die anisotropen Regularisierung noch besser an die Objektgrenzen anzupassen ist in Kapitel 4.2.4 die Hinzunahme eines in [40] vorgestellten Liniensegment-Detektors (LSD) erläutert. Dabei

werden zunächst durch den LSD Liniensegmente innerhalb des Eingangsbildes erkannt. Eine anschließende Berechnung der Gewichtstensoren basierend auf dem LSD-Bild führt zu einer zweiten Sammlung an Gewichtstensoren (G'). Zum Schluss werden die Gewichtstensoren aus G, welche auf Basis des Eingangsbildes berechnet wurden, an den Stellen der detektierten Liniensegmenten durch die Tensoren in G' ersetzt. Dies soll eine bessere Hervorhebung der Objektkanten bewirken, da die Gewichtstensoren in G' durch die Liniensegmente strenger an die Grenzen angepasst werden.

Wie bereits in 4.2.4 erwähnt, bietet die Implementierung des LSDs verschiedene Parameter um die Erkennung von Liniensegmenten zu beeinflussen. Darin unter anderem:

- Der Skalierungsfaktor s sowie die Standardabweichung σ des Gaußkerns, welche zum Aufbau einer Gaußpyramide genutzt werden. Die Verwendung einer Gaußpyramide zur Segmentdetektion soll dabei helfen mit Aliasing-Artefakten bei schrägen Kanten im Eingangsbild besser umzugehen.

- Die Winkeltoleranz τ, bis zu der benachbarten „Level-Lines“ gemeinsam als Kandidat für ein Liniensegment betrachtet werden. Es wird eine „Region-Growing“-Methode angewendet um benachbarte „Level-Lines“ zu gruppieren. Ist die Orientierung einer benachbarten „Level-Line“ größer als τ wird sie nicht zu der Gruppe hinzugenommen.

Die Standardwerte der oben genannten Parameter, welche in [40] empirisch ermittelt wurden, sind wie folgt: $s = 0.8$, $\sigma = 0.6$ und $\tau = 22.5°$. Abbildung 4.6 zeigt das Ergebnisbild des LSDs in Standardkonfiguration. Im Folgenden werden verschiedene Konfigurationen des LSDs, sowie deren Auswirkung auf die Glättung getestet und evaluiert.

In Abbildung 5.11 sind die Ergebnisse verschiedener Konfigurationen der oben genannten Parameter für einen Ausschnitt des Frame 75 aus dem Tsukuba-2012 Datensatzes abgebildet. Dabei sind in der ersten und dritten Reihe jeweils die Ergebnisbilder des LSDs enthalten. Die Reihen zwei und vier zeigen die Ergebnisse einer linearen Diffusion mittels den angepassten Tensoren. Dabei basieren die Gewichtstensoren auf der Exponentialfunktion mit $\alpha_{exp} = 5$ und $\beta_{exp} = 0.5$. Die hierbei ausgewählte e-Funktion mit entsprechender Parametrisierung spiegelt nicht die endgültige Auswahl für den anisotropen Gewichtstensor wieder. Die Auswahl wurde lediglich zu Demonstrationszwecken gewählt, um die Auswirkungen der Parameter des LSDs zu erläutert und zu vergleichen.

Abbildung 5.11 *(a)* zeigt den betrachteten Bildausschnitt im Original und in geglätteter Form. Ausschnitt *(b)* zeigt das Ergebnis des LSDs in Standardkonfiguration, sowie die Diffusion mit dem entsprechend verbesserten Diffusionstensor. Bereits hier ist eine verbesserte Anpassung der Regularisierung zu erkennen, denn die Objektgrenzen, an denen Liniensegmente erkannt wurden, werden deutlich besser hervorgehoben. So z. B. am unteren Rand der Lampe, sowie in den Gesichtsstrukturen der Büste. Die in Ausschnitt *(c)* dargestellten Ergebnisbilder sind mit einem Skalierungsfaktor $s = 0,5$ erstellt worden. Durch diesen reduzierten Faktor wurde das Eingangsbild auf die Hälfte der ursprünglichen Größe skaliert, wodurch viele Details verloren gehen. Dieser Verlust führt dazu, dass weniger Liniensegmente erkannt werden, was sich in der Struktur der Büste und in der Erkennung des Lampenkabels zeigt.

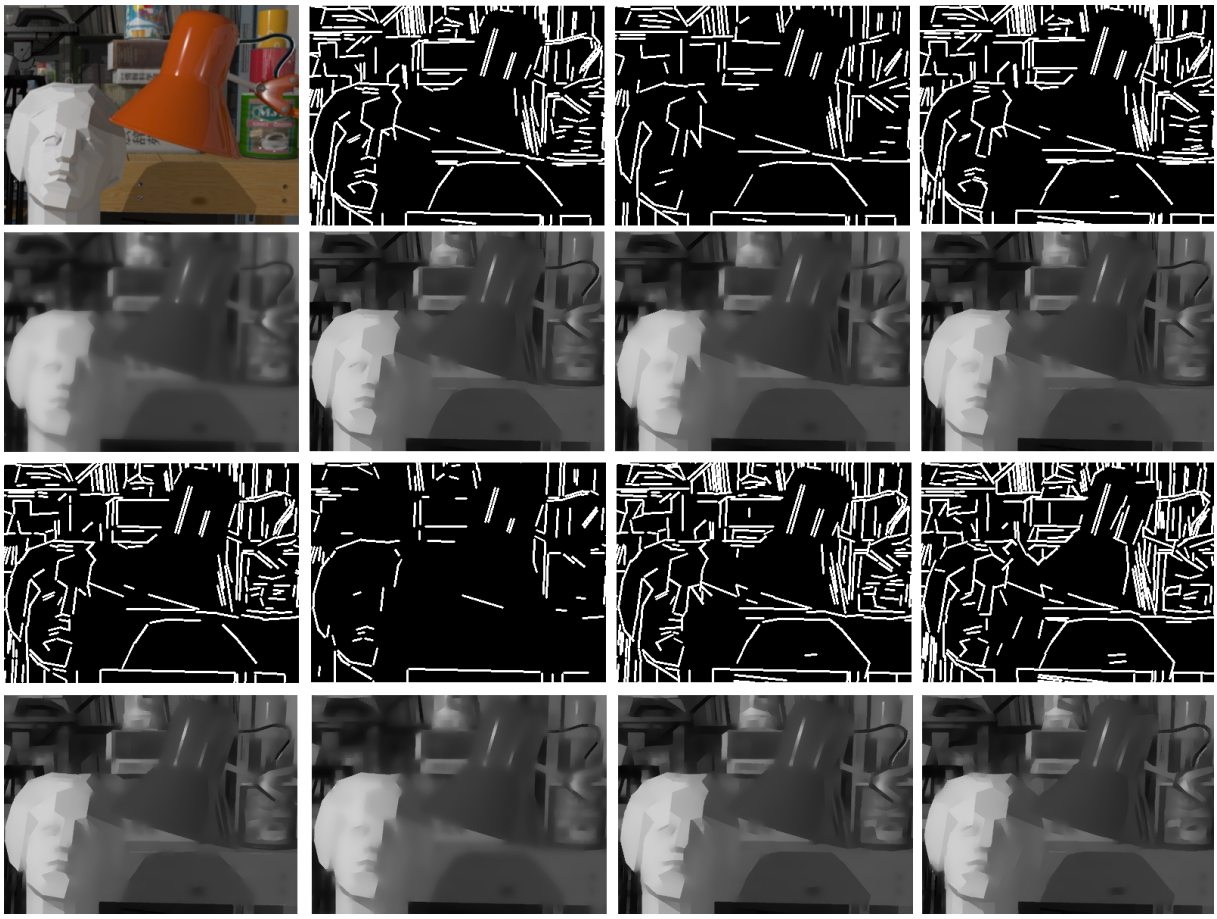

Abbildung 5.11: Vergleich verschiedener Konfigurationen des Liniensegment-Detektors (LSD) [40]. Darunter jeweils die Diffusion des Ausschnittes mit Verbesserung durch entsprechendes LSD-Ergebnis. Von links nach rechts, von oben nach unten: **(a)** Betrachteter Ausschnitt aus Frame 75 des Tsukuba-2012 Datensatzes. Diffusion ohne LSD. **(b)** LSD in Standardkonfiguration. Skalierungsfaktor $s = 0.8$, Gaußglättung mit $\sigma = 0.6$, Winkeltoleranz von $\tau = 22.5°$. **(c)** $s = 0.5$, $\sigma = 0.6$, $\tau = 22.5°$. **(d)** $s = 0.8$, $\sigma = 0.3$, $\tau = 22.5°$. **(e)** $s = 0.8$, $\sigma = 0.8$, $\tau = 22.5°$. **(f)** $s = 0.8$, $\sigma = 0.8$, $\tau = 10°$. **(g)** $s = 0.8$, $\sigma = 0.8$, $\tau = 30°$. **(h)** $s = 0.8$, $\sigma = 0.8$, $\tau = 50°$.

Folglich werden dadurch bei der Diffusion einige Objektkanten nicht erkannt und damit über sie hinweg geglättet.

Für Ausschnitte *(d)* und *(e)* ist der Skalierungsfaktor gleich geblieben, während die Standardabweichung σ des Gaußkerns verändert wurde. Eine Veränderung der Größe des Gaußkerns bewirkt eine schwächere bzw. stärkere Glättung bei der Skalierung des Eingangsbildes durch s. Durch eine Verringerung von $\sigma = 0.3$ (vgl. Ausschnitt *(d)*) können gerade bei hochfrequenten Bildstrukturen durch die geringe Glättung Aliasing-Artefakte auftreten, welche wiederum zur Detektion von Liniensegmenten führen. Diese zusätzlichen Detektionen sind dabei nicht zwingend an Objektgrenzen lokalisiert. So treten z. B. am rechten Rand des Lampenschirms oder am oberen Rand der Büste eine höhere Anzahl an Liniensegmenten auf, die zudem dichter beieinander liegen. Diese zusätzlichen Detektionen sind dabei auch in der Regularisierung zu erkennen und bewirken beispielsweise innerhalb der Büste Artefakte, über die nicht geglättet wird. Bei einer Erhöhung von $\sigma = 0.8$, wie es in Ausschnitt *(e)* zu sehen ist, werden weniger Liniensegmente erkannt, welche gleichzeitig eine höhere Aussagekraft bezüglich Objektgrenzen haben. Der Nachteil hierbei ist jedoch, dass die Liniensegmente wesentlich fragmentierter sind. Das heißt, dass es mehr Lücken zwischen einzelnen Linsensegmenten gibt und es folglich an den entsprechenden Stellen zu mehr Glättungen kommt.

Um diese Linienfragmentierung zu reduzieren kann die Winkeltoleranz τ angepasst werden, wodurch mehr benachbarte „Level-Lines“ zu einem gemeinsamen Segmentkandidaten gruppiert werden. In den Ausschnitten *(f)*, *(g)* und *(h)* sind Ergebnisse drei verschiedener Konfigurationen von τ abgebildet. Eine Reduktion der Toleranz auf $\tau = 10°$ (vgl. Ausschnitt *(f)*) bewirkt, dass der LSD wesentlich strikter ist. Dadurch werden deutlich weniger Liniensegmente erkannt, was diese Parametrisierung nicht praktikabel macht. Zwar wird an den entsprechenden Stellen die Regularisierung verbessert, aber es werden zu wenige Liniensegmente erkannt. Wird die Toleranz auf $\tau = 50$ mehr als verdoppelt (vgl. Ausschnitt *(h)*), so werden zu viele Liniensegmente detektiert. Auch hier führt die hohe Anzahl an Segmenten, die nicht passend lokalisiert sind, zu mehr Strukturen in Bereichen, in denen eigentlich regularisiert werden sollte. Eine bessere Wahl der Toleranz ist in *(g)* dargestellt. Hier ist sie mit $\tau = 30$ ein wenig höher als die Standardkonfiguration. Dies führt zu mehr zusammenhängenden und aussagekräftigen Liniensegmenten, während die Dichte der Detektionen nicht zu hoch ist. Diese Konfiguration bietet ein gutes Mittelmaß zwischen zu wenig und zu viel Segmenten.

Eine Anwendung der durch die detektierten Liniensegmente erweiterten Tensoren in der TGV^2-gestützten Rekonstruktion führt zu den quantitativen Ergebnissen in Tabelle 5.8. Hierbei wurden für den bildgestützten Aufbau der Gewichtstensoren G die gleichen Parametrisierungen verwendet wie in Tabelle 5.7. Für die Konstruktion der Tensoren G', basierend auf das Ergebnisbild des LSDs, wurde eine möglichst steile Gewichtsfunktion verwendet, damit die Regularisierung über die Liniensegmente hinweg möglichst stark reduziert wird. Beim Vergleich der Zahlen aus Tabellen 5.7 und 5.8 ist zu erkennen, dass der Fehler in den meisten Ergebnissen durch die Hinzunahme der Liniensegmenten leicht zunimmt. Gerade bei den

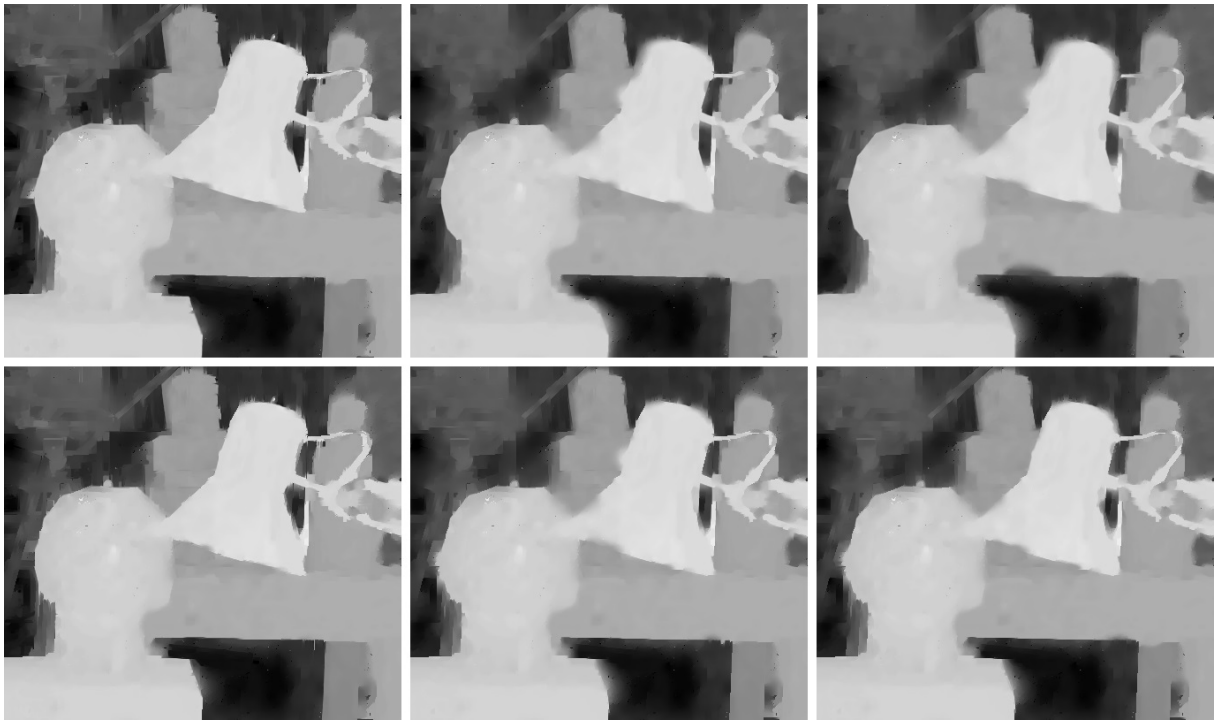

Abbildung 5.12: Gegenüberstellung verschiedener Gewichtstensoren mit und ohne Erweiterung durch den LSD. Von links nach rechts, von oben nach unten: **(a)** Gewichtstensor auf Basis des Perona-Malik-Terms mit $\lambda_{pm} = 3$, ohne verbesserte Anpassung durch detektierte Liniensegmente. **(b)** Perona-Malik-Term mit $\lambda_{pm} = 10$, ohne LSD. **(c)** e-Funktion mit $\alpha_{exp} = 5$ & $\beta_{exp} = 0.5$, ohne LSD. **(d)** Perona-Malik-Term mit $\lambda_{pm} = 3$, mit LSD. **(e)** Perona-Malik-Term mit $\lambda_{pm} = 10$, mit LSD. **(f)** e-Funktion mit $\alpha_{exp} = 5$ & $\beta_{exp} = 0.5$, mit LSD.

Tensoren mit einer strikten Anisotropie werden die quantitativen Ergebnisse schlechter. Nur die Ergebnisse der zwei nachgiebigeren Tensoren, basierend auf dem Perona-Malik-Term mit $\lambda_{pm} = 10$ bzw. auf der e-Funktion mit $\alpha_{exp} = 5$ und $\beta_{exp} = 0.5$, werden wie erwartet durch die Integration des LSDs verbessert.

In Abbildung 5.12 ist ein qualitativer Vergleich zwischen einzelner Tensoren abgebildet, die ohne und mit der Integration von Liniensegmenten erstellt wurden. Die erste Reihe zeigt drei Ausschnitte aus den Tiefenkarten, die ohne den Ergebnissen des LSDs berechnet wurden. In Reihe zwei sind entsprechend die Ausschnitte der Tiefenkarten dargestellt, die mittels erweiterter Tensoren berechneten wurden. In den Ausschnitten *(a)* und *(d)* wurde für die Konstruktion von G der Perona-Malik-Term mit $\lambda_{pm} = 3$ gewählt. Dies entspricht in beiden Durchführungen (mit und ohne LSD) jeweils dem quantitativ besten Ergebnis. Die anderen vier Ausschnitte zeigen die Ergebnisse einer weniger strikten Gewichtsfunktion. In *(b)* und *(e)* ist für G der Perona-Malik-Term mit $\lambda_{pm} = 10$ gewählt worden und in *(d)* und *(f)* wurde die Exponentialfunktion mit $\alpha_{exp} = 5$ und $\beta_{exp} = 0.5$ verwendet.

Gerade in den Ausschnitten *(e)* und *(f)* ist eine qualitative Verbesserung durch die Hinzunahme der Liniensegmente zu erkennen. So werden zum Beispiel die Tiefendiskontinuitäten im oberen Bereich der Lampe oder am unteren Rand des Tisches schärfer hervorgehoben als in den Ausschnitten *(b)* und *(c)*. Zudem wird durch die Erweiterung der Tensoren das Kabel der Lampe besser rekonstruiert. Für die adaptive Regularisierung mittels einem strikten Tensor, basierend auf dem Perona-Malik-Term mit $\lambda_{pm} = 3$, zeichnet sich durch die Erweiterung mittels des LSDs jedoch keine sonderlichen Unterschiede ab.

Zusammenfassend bedeutet dies, dass je nach Wahl der Gewichtsfunktion, die für den Aufbau der Tensoren G verwendet wird, die Integration von Liniensegmenten nicht unbedingt nötig ist. Ist die Anisotropie der ursprünglichen Tensoren schon strikt genug, werden die Objektgrenzen bereits gut rekonstruiert. Wird jedoch ein schwächerer anisotroper Gewichtstensor verwendet ist die Hinzunahme der Liniensegmente bedeutend für den Erhalt der Tiefendiskontinuitäten an Objektgrenzen. Eine endgültige Diskussion über die Verwendung des LSDs ist in Kapitel 5.4 zu finden.

Tensoren	Frame 75	Frame 300	Frame 380
Perona-Malik $\lambda_{pm} = 3$	**12,68**	**30,73**	21,23
Perona-Malik $\lambda_{pm} = 6$	12,80	30,81	21,49
Perona-Malik $\lambda_{pm} = 10$	12,99	30,97	21,73
Charbonnier $\lambda_{ch} = 3$	12,74	30,79	21,52
Exponential $\alpha_{exp} = 5, \beta_{exp} = 0,2$	12,82	30,96	21,69
Exponential $\alpha_{exp} = 5, \beta_{exp} = 0,3$	12,71	30,88	**21,26**
Exponential $\alpha_{exp} = 5, \beta_{exp} = 0,5$	13,01	31,00	21,62

Tabelle 5.8: Quantitative Ergebnisse des TGV^2-Verfahrens mit erweiterten Gewichtstensoren. Da die Veränderung der Tensoren keine Auswirkung auf die Laufzeit hat, wird diese hier nicht aufgelistet.

5.3.2.3 *Vorglättung zur Berechnung der Gewichtstensoren*

In den vorherigen Kapiteln wurden verschiedene anisotrope Gewichtstensoren für die lokale adaptive Gewichtung der Regularisierung vorgestellt und untersucht. Zunächst ging dabei hervor, dass bei einer strikten Anisotropie die Kanten an den Objektgrenzen zwar präzise erhalten wurden, jedoch innerhalb der Objekte nicht genug Regularisierung auftrat. Daraufhin wurde untersucht, ob unter der Verwendung eines nachgiebigeren Tensors, der durch das Ergebnis eines Liniensegment-Detektors erweitert wurde, eine bessere Homogenität innerhalb von Objekten, bei einer gleichzeitigen Diskontinuität an den Grenzen erzielt wird. Eine weitere Möglichkeit strenge Gewichtstensoren innerhalb von Objekten zu schwächen, ist die Berechnung der Tensoren auf Basis eines vorgeglätteten Eingangsbilds. Ähnlich wie bei

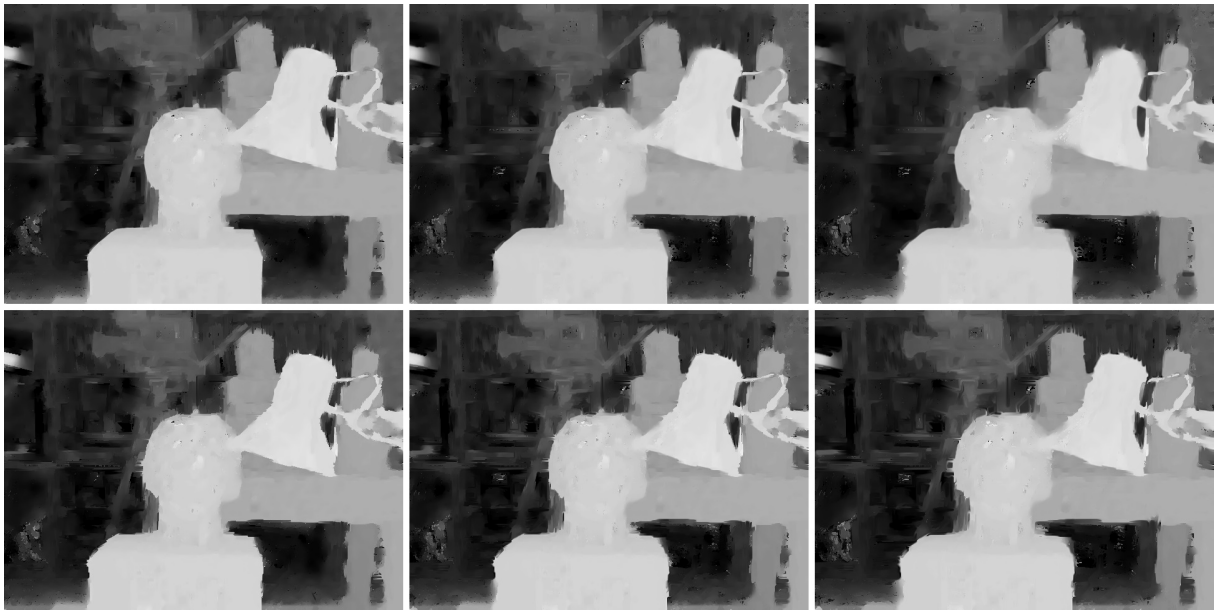

Abbildung 5.13: Gegenüberstellung der Auswirkung verschiedener Standardabweichungen σ zur Vorglättung mittels eines Gaußkerns. Gewichtstensoren basierend auf Perona-Malik-Term mit $\lambda_{pm} = 3$. Kostenfunktion: SAD mit fester Aggregationsnachbarschaft der Größe 11×11. Von links nach rechts: **(a)** $\sigma = 0.5$, **(b)** $\sigma = 1.0$, **(c)** $\sigma = 1.5$, **(d)** $\sigma = 0.5$ mit Erweiterung durch LSD, **(e)** $\sigma = 1.0$ mit Erweiterung durch LSD, **(f)** $\sigma = 1.5$ mit Erweiterung durch LSD.

der Verbesserung der isotropen Diffusionstensoren im Umgang mit Rauschen, wird dabei das Eingangsbild mit einem Gaußkern mit der Standardabweichung σ gefaltet.

Abbildung 5.13 zeigt eine Reihe von Tiefenkarten bei denen die Gewichtstensoren auf Basis verschiedener Vorglättungen in Abhängigkeit der Standardabweichung σ berechnet wurden. Die Tensoren basieren dabei auf dem Perona-Malik-Term mit $\lambda_{pm} = 3$. Als Kostenfunktion wurde die SAD mit einer Nachbarschaftsgröße von 11×11 gewählt. Während die Tensoren in der ersten Reihe alleinig auf dem vorgeglätteten Eingangsbild basieren, sind die Tensoren in der zweiten Reihe durch die Hinzunahme des LSDs erweitert. Qualitativ ist zu erkennen, dass die Objektkanten mit zunehmendem σ insgesamt glatter werden. Gleichzeitig bewirkt eine stärkere Vorglättung jedoch auch, dass deutlich mehr Kanten verschwimmen. Des Weiteren scheint gerade im Hintergrund die Homogenität innerhalb der Objekte zuzunehmen. Dabei werden weniger Details im Bücherregal erhalten, was den Hintergrund insgesamt glatter erscheinen lässt. Durch die Erweiterung mittels des LSDs werden selbst bei $\sigma = 1,5$ die Kanten besser erhalten. Jedoch scheinen die Objekte insgesamt mehr auszufransen. Dies lässt sich durch die stärkere Regularisierung entlang der Liniensegmente erklären. Gerade durch die Strukturen im Hintergrund werden viele Liniensegmente orthogonal zu den Objektgrenzen im Vordergrund detektiert (vgl. Abb. 4.6), wodurch die Kanten der Vordergrundobjekte ausfransen.

In Tabelle 5.9 ist ein quantitativer Vergleich der verschiedenen Stufen der Vorglättung aufgeführt. Darin zeigt sich, dass gerade bei einem größeren σ die Erweiterung durch den LSD zu besseren Ergebnissen führt. Im Vergleich zu den Zahlen aus Tabellen 5.7 (einfache Tensoren) und 5.8 (erweiterte Tensoren), erzielt die Verwendung einer Vorglättung ähnlich gute Ergebnisse.

	Standardabweichung	Frame 75	Frame 300	Frame 380
ohne LSD	$\sigma = 0,5$	**12,68**	**30,46**	**21,70**
	$\sigma = 1,0$	13,14	30,70	22,61
	$\sigma = 1,5$	13,62	31,03	23,29
mit LSD	$\sigma = 0,5$	**13,00**	**30,53**	**21,17**
	$\sigma = 1,0$	13,33	30,92	21,96
	$\sigma = 1,5$	13,59	31,10	22,37

Tabelle 5.9: Quantitativer Vergleich der Standardabweichungen des Gaußkerns für die Vorglättung der Eingangsbilder.

5.3.2.4 *Einfluss des Datenterms*

Der zweite Teil des Energiefunktionals der TGV^2-gestützten Rekonstruktion ist der Daten- oder Kostenterm. Durch ihn werden die Eingangsdaten miteinander verglichen und den verschiedenen Tiefenschätzungen Kosten zugeordnet. Durch die Minimierung der Summe aus Regularisierungs- und Datenterm wird ein Kompromiss zwischen der Glättung und der Tiefenschätzung anhand von Punktkorrespondenzen gefunden. Ähnlich wie im Kapitel 5.3.1.4, welches sich mit den Kostenfunktionen und Aggregationsnachbarschaften als Teil des gewöhnlichen Plane-Sweep-Verfahrens beschäftigt, wird im Folgenden untersucht, welchen Einfluss der Datenterm auf die zu berechnende Tiefenkarte hat und wie die Kosten besser in die Rekonstruktion einfließen können.

Während im vorangegangenen Kapitel die lokale adaptive Gewichtung des Regularisierungsterms an die Bildstrukturen untersucht wurde, kann der Einfluss der Regularisierung und der Vergleichskosten zusätzlich global festgelegt werden. Dabei wird der Einfluss der Terme im Energiefunktional durch die entsprechenden Multiplikatoren λ_i gesteuert (vgl. Gl. 3.13). Während λ_a gemäß der Empfehlung aus [19] auf den Wert $8\lambda_s$ fixiert ist, wird im Folgenden für die einzelnen Kostenfunktionen untersucht, welche globale Gewichtung die besten Ergebnisse erreichen. Hierbei werden zunächst die Summe der Absoluten Differenzen (SAD) mit einer konstanten und adaptiven Nachbarschaftsgröße betrachtet. In den folgenden Experimenten ist für die anisotrope Gewichtung der Perona-Malik-Term mit $\lambda_{pm} = 3$ gewählt, sowie die Tensoren durch die Hinzunahme von Liniensegmenten erweitert.

Die Ausschnitte *(a)* und *(b)* in Abbildung 5.14 zeigen zwei verschiedene Parametrisierungen von λ_s und λ_d bei einer konstanten Nachbarschaftsgröße von 11×11. In beiden Parame-

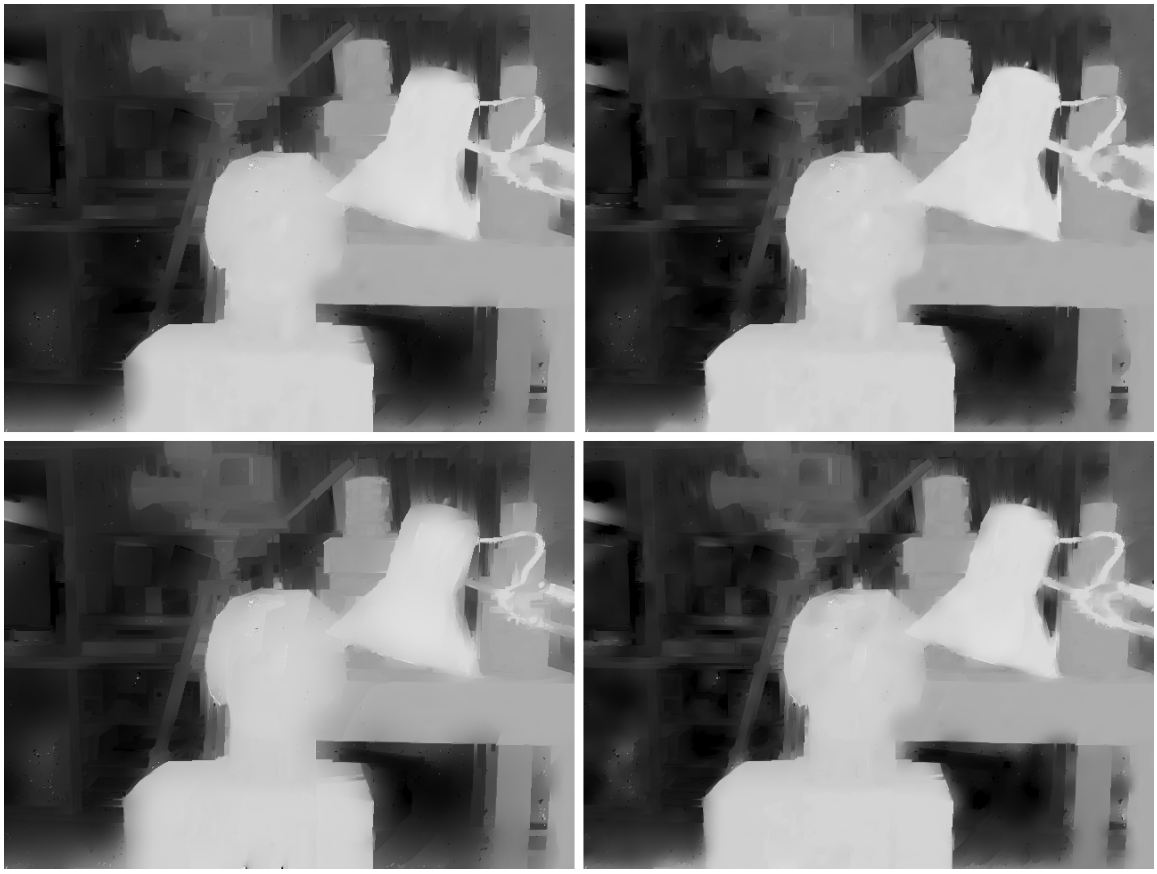

Abbildung 5.14: Gegenüberstellung verschiedener Konfigurationen des Datenterms unter der Verwendung der SAD als Kostenfunktion. Größe der Aggregationsnachbarschaft: 11×11. G basierend auf Perona-Malik-Term mit $\lambda_{pm} = 3$, erweitert durch LSD. Von links nach rechts, von oben nach unten: **(a)** Feste Nachbarschaftsgröße. $\lambda_s = 1$ & $\lambda_d = 0,2$. **(b)** Feste Nachbarschaftsgröße. $\lambda_s = 1$ & $\lambda_d = 0,5$. **(c)** Adaptive Aggregationsnachbarschaft. $\lambda_s = 1$ & $\lambda_d = 1$. **(d)** Adaptive Aggregationsnachbarschaft. $\lambda_s = 0,2$ & $\lambda_d = 1$.

trisierungen hat der Regularisierungsterm einen größeren Einfluss. Während die Parameter zu *(b)* ein quantitativ besseres Ergebnis erzielen (vgl. Tab.5.10), ist es schwierig qualitativ zu beurteilen, welche Parametrisierung besser ist. Deutlich zu erkennen ist, dass in *(b)* die Tiefendiskontinuitäten an den Objektkanten sauberer hervorgehoben werden. Jedoch führt die stärkere Regularisierung in *(a)* zu einer schöneren und weicheren Struktur im Hintergrund und an der Kamera. Im Vergleich zu den Ergebnissen, die in Abbildung 5.12 zu sehen sind, bei denen $\lambda_s = \lambda_d = 1$ gewählt sind, zeigen die zwei Tiefenkarten aus der ersten Reihe von Abbildung 5.14 eine deutlich glattere Rekonstruktion.

Für die Tiefenkarten, welche in Reihe zwei der Abbildung 5.14 zu sehen sind ist eine Adaptive Aggregationsnachbarschaft gewählt. Wie in Kapitel 4.4 erläutert, entsteht bei einer konstanten Größe der Nachbarschaft das Phänomen des „foreground-fattening“. Durch die

implizite Annahme, dass alle Pixel innerhalb einer Nachbarschaft die gleiche Tiefe haben, werden hierbei Objekte die im Vordergrund liegen dicker. Um dies zu vermeiden wird bei der Kostenaggregation als Teil der TGV^2-gestützten Rekonstruktion eine Adaptive Nachbarschaft verwendet, in der die einzelnen Pixel je nach Lokalität und Erscheinung anders gewichtet werden. In der Berechnung der Tiefenkarte *(c)* ist zunächst wieder eine gleichwertige globale Gewichtung des Regularisierungs- und des Datenterms gewählt. Da die Objektgrenzen teilweise mehr verschwimmen als in *(a)* und *(b)* (vgl. oberer Rand der Lampe oder das Umfeld der Kamera), ist in *(d)* ein stärkerer Einfluss durch den Datenterm gewählt. Die Verwendung der Adaptiven Nachbarschaft bewirkt, dass auch kleinere Strukturen besser rekonstruiert werden. So beispielsweise das Lampenkabel und das Stativ der Kamera. Zudem werden auch feinere Details im Bücherregal oder auf der Kamera sichtbar. Die quantitative Auswertung in Tabelle 5.10 ergibt zudem, dass die Verwendung einer Adaptiven Nachbarschaft, mit einer höheren Gewichtung des Datenterms, in *(d)* die bisher beste Rekonstruktion bewirkt.

Als zweite Kostenfunktion ist in Kapitel 2.3 die Hammingdistanz der Census-Transformation vorgestellt. Abbildung 5.15 zeigt vier Tiefenkarten die mittels dieser Kostenfunktion berechnet wurden. Aufgrund der Erkenntnis aus Abbildung 5.7 werden die Kosten hierbei in einer Adaptiven Nachbarschaft der Größe 21×21 aufsummiert. Für den globalen Einfluss der beiden Terme wurde hierbei in *(a)* zunächst wieder eine Gleichgewichtung gewählt. Nicht nur qualitativ, sondern auch quantitativ (vgl. Tab. 5.10) ist ersichtlich, dass eine solche Gewichtung keine zufriedenstellende Ergebnisse liefert. Zwar werden in *(a)* die Objektkanten sauber hervorgehoben, jedoch entstehen Artefakte in Bereichen, die eigentlich homogen sein sollten. Ähnlich wie bei der Verwendung der CT im gewöhnlichen Plane-Sweep, lassen sich diese Artefakte durch das Auftreten von Mehrdeutigkeiten aufgrund der geringeren Aussagekraft der Bitfolgen begründen. Wie bereits in Kapitel 5.3.1.4 erwähnt, benötigt eine Verwendung der CT daher eine stärkere Regularisierung. Dies führt zu Tiefenkarte *(b)*, bei der mit $\lambda_s = 1$ und $\lambda_d = 0,2$ dem Datenterm eine deutlich geringere Gewichtung als dem Regularisierungsterm gegeben wird. Diese Änderung der Gewichte bewirkt zwar bessere Ergebnisse, aber dennoch treten weiterhin Artefakte auf. Eine eingängige Schlussfolgerung wäre eine weitere Erhöhung der Regularisierung, jedoch führt dies zu einer größeren Glättung an den Kanten, die nicht vom LSD erkannt wurden.

Aus diesem Grund wurde im Rahmen dieser Arbeit eine zusätzliche adaptive Gewichtung des Datenterms eingeführt. Ähnlich wie bei der anisotropen Gewichtung der Regularisierung, ist hierbei die Idee, den Datenterm in manchen Bereichen stärker zu gewichten als in anderen. So sollte zum Beispiel der Einfluss des Datenterms innerhalb von Szenenobjekten und homogenen Bildbereichen reduziert werden, und gleichzeitig an Objektgrenzen und in strukturierten Bereichen erhalten bleiben. Dies führt zu einer lokalen Gewichtung des Datenterms, die invers zu der des Regularisierungsterms ist. Hierfür wurde in das Energiefunktio-

Abbildung 5.15: Gegenüberstellung verschiedener Konfigurationen des Datenterms unter der Verwendung der CT-basierten Kostenfunktion. Adaptive Aggregationsnachbarschaft der Größe 21×21. G basierend auf Perona-Malik-Term mit $\lambda_{pm} = 3$, erweitert durch LSD. Von links nach rechts, von oben nach unten: **(a)** $\lambda_s = 1$ & $\lambda_d = 1$. **(b)** $\lambda_s = 1$ & $\lambda_d = 0,2$. **(c)** Lokale adaptive Gewichtung des Datenterms. $\lambda_s = 1$ & $\lambda_d = 0,2$. **(d)** Lokale adaptive Gewichtung des Datenterms. $\lambda_s = 0,2$ & $\lambda_d = 0,2$

nal der TGV^2-gestützten Rekonstruktion (vgl. Gl. 3.13) eine zusätzliche vom Bildgradienten abhängige Gewichtsfunktion eingefügt. Dies führt zu einem neuen Energiefunktional gemäß

$$\boldsymbol{u} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \left\{ \underbrace{\lambda_s \ |G(\nabla \boldsymbol{u} - \boldsymbol{v})| \ + \ \lambda_a \ |\nabla \boldsymbol{v}|}_{R(\boldsymbol{u})} + \lambda_d \ (1 - g(\nabla I_{ref})) \ C(\boldsymbol{u}) \right\}. \tag{5.2}$$

Für die Gewichtsfunktion $g(\nabla I_{ref})$ wird dabei die gleiche Funktion gewählt wie zum Aufbau des anisotropen Gewichtstensors G. Zu beachten ist dabei, dass der lokale Gewichtsfaktor für den Datenterm invertiert sein muss. Abbildung 5.16 zeig die Gewichtsfunktion für den anisotropen Tensor, basierend auf dem Perona-Malik-Term mit $\lambda_{pm} = 3$, sowie die entsprechend invertierte Funktion für die lokale adaptive Gewichtung des Datenterms. Im Lösungsalgo-

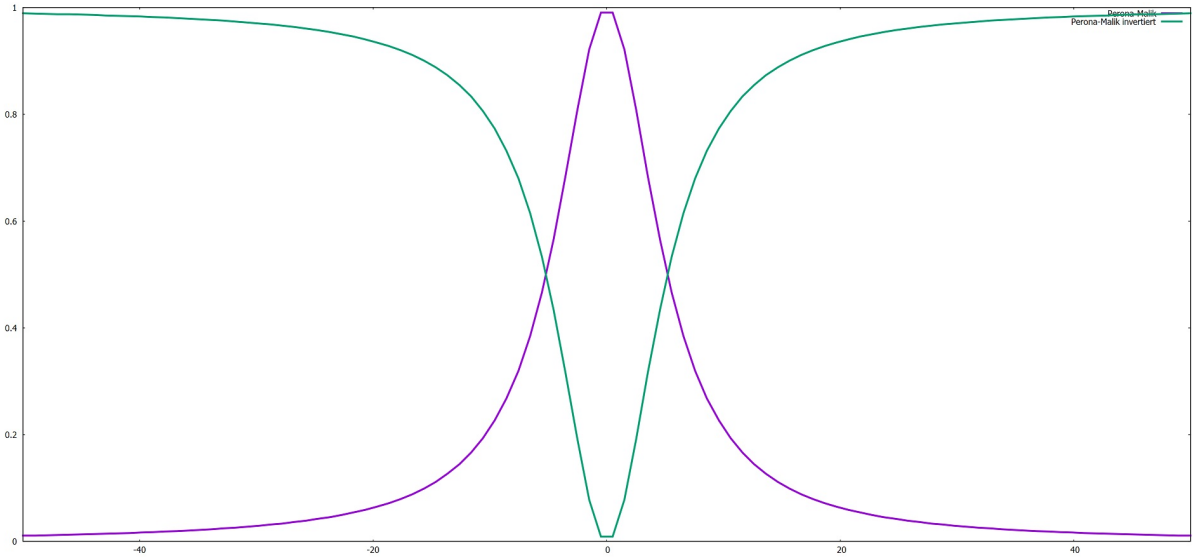

Abbildung 5.16: **Violett:** Funktionskurve des Perona-Malik-Terms mit $\lambda_{pm} = 3$. **Grün:** Invertierte Funktionskurve des Perona-Malik-Terms mit $\lambda_{pm} = 3$.

rithmus der Gleichung 5.2 fließt die $g(\nabla I_{ref})$ in die Lösung des Datenterms wie folgt ein: $\boldsymbol{a}^{n+1} = \text{argmin}_{\boldsymbol{a}\in\Gamma} \left\{ \lambda_3 \, (1 - g(\nabla I_{ref})) \, C(\boldsymbol{a}) + \boldsymbol{L}^n(\boldsymbol{u}^n - \boldsymbol{a}) + \frac{(\boldsymbol{u}^n - \boldsymbol{a})^2}{2\theta^n} \right\}$.

Die Ergebnisse der Erweiterung durch eine lokale adaptive Gewichtung des Datenterms sind in den Tiefenkarten *(c)* und *(d)* der Abbildung 5.15, sowie in den letzten beiden Reihen der Tabelle 5.10 aufgeführt. Tiefenkarte *(c)* zeigt, dass bei gleicher Parametrisierung wie in *(b)* durch die adaptive Gewichtung des Datenterms wesentlich bessere homogene Strukturen innerhalb von Objekten erreicht werden. Die Integration der adaptiven Gewichtung des Datenterms erlaubt es nun, den globalen Einfluss der Regularisierung weiter zu reduzieren, ohne dabei die homogene Struktur innerhalb der Objekte zu verlieren (vgl. Tiefenkarte *(d)*). Die quantitativen Ergebnisse aus Tabelle 5.10 zeigen, dass mittels der Erweiterung der adaptiven Gewichtung des Datenterms unter Verwendung der CT-basierten Kostenfunktion ähnlich gute Ergebnisse erreicht werden wie mit der SAD-basierten Kostenfunktion. Auch die qualitativen Vergleiche zwischen Abbildungen 5.14 *(d)* und 5.15 *(d)* bestätigen dies.

5.3.2.5 *Auswirkungen der Iterationen*

Als letzten Schritt der experimentellen Auswertung des Systems soll im Folgenden untersucht werden, inwieweit die Iterationen des Algorithmus zur Lösung des TGV^2-basierten Energiefunktional reduziert werden können, ohne zu viel Qualität der berechneten Tiefenkarte zu verlieren. Eine Verringerung der Iterationen führt gleichzeitig zu einer Reduzierung der Laufzeit, welches für die Konfiguration eines echtzeitfähigen Systems ausschlaggebend sein kann.

In Kapitel 3.3.4 wird die Strategie vorgestellt, mit der das TGV^2-gestützte Energiefunktional gelöst wird. Der entsprechende Algorithmus ist dabei aus zwei verschachtelten Iterationen

		Frame 75	Frame 300	Frame 380
SAD	konstante Nachbarschaft (KN), $\lambda_s = 1$, $\lambda_d = 0.2$	12,98	29,91	19,98
	konstante Nachbarschaft (KN), $\lambda_s = 1$, $\lambda_d = 0.5$	12,43	30,22	20,64
	adaptive Nachbarschaft (AN), $\lambda_s = 1$, $\lambda_d = 1$	13,26	30,82	20,18
	adaptive Nachbarschaft (AN), $\lambda_s = 0.2$, $\lambda_d = 1$	**12,23**	**29,87**	**19,92**
CT	adaptive Nachbarschaft (AN), $\lambda_s = 1$, $\lambda_d = 1$	18,31	37,09	25,85
	adaptive Nachbarschaft(AN), $\lambda_s = 1$, $\lambda_d = 0.2$	15,07	33,16	21,56
	AN, lokale Datengew., $\lambda_s = 1$, $\lambda_d = 0.2$	14,06	31,83	20,91
	AN, lokale Datengew., $\lambda_s = 0.2$, $\lambda_d = 0.2$	**12,69**	**30,74**	**20,41**

Tabelle 5.10: Quantitative Gegenüberstellung der Kostenfunktionen als Teil der TGV^2-gestützten Rekonstruktion.

aufgebaut. Die innere Schleife führt eine Reihe an Primal-Dual-Iterationen ($smoothItr$) aus, in denen der Regularisierungsterm gelöst wird. In der äußeren Schleife ($globalItr$) wird die Lösung des Datenterms gesucht und in jeder Iteration die Lösungen des Regularisierungs- und Datenterms immer weiter einander angeglichen. Gemäß der Empfehlung in [19] wurden für die bisherigen Berechnungen die Anzahl der Iterationen wie folgt gewählt: $globalItr = 80$ und $smoothItr = 150$. Das quantitativ beste Ergebnis der vorangegangenen Berechnungen, mit entsprechender Anzahl an Iterationen, ist noch einmal in Reihe eins der Tabelle 5.11 aufgelistet. Die entsprechende Laufzeit ist im zweiten Teil der Tabelle zu finden.

Weiterhin sind in der Tabelle 5.11 die quantitativen Ergebnisse verschiedener Anzahlen von Iterationen aufgelistet. Die Ergebnisse zeigen, dass eine Verringerung der Iterationen keine quantitativ hohen Verschlechterung (außer bei Frame 300), jedoch aber eine deutliche Beschleunigung des Verfahrens bewirkt. Des Weiteren geht aus den Zahlen hervor, dass eine Verkleinerung der Anzahl an Glättungsiterationen größere Auswirkungen auf die Qualität der Ergebnisse hat als eine Verringerung der globalen Iterationen. In den jeweils letzten Reihen der beiden Teile ist ein weiterer, bisher unbekannter Parameter η aufgelistet. Dieser ist im Rahmen dieser Arbeit mit dem Zweck neu eingeführt worden, Anzahl der durchgeführten Glättungsiterationen mit jeder globalen Iteration zu verringern. Dadurch wird erzielt, dass zu Beginn der Berechnung, bei der die Lösungen des Regularisierungs- und des Datenterms noch wenig aneinander angeglichen sind, eine hohe Anzahl an Glättungsiterationen durchgeführt wird. Diese soll aber gleichzeitig mit zunehmendem angleichen der Lösungen abnehmen. Hierbei werden die Anzahl der Glättungsiterationen in jeder globalen Iteration gemäß $smoothItr^n = smoothItr^0 - \eta \cdot n$ mit $0 \leq n < globalItr$ berechnet.

Aus den quantitativen Ergebnissen geht hervor, dass eine Reduktion der Iterationen zwar einen geringen Verlust an Qualität bringt, jedoch eine teilweise starke Verbesserung der Laufzeit bewirkt. Gerade die Verwendung der schrittweisen Reduzierung bietet das beste Qualität-zu-Laufzeit Verhältnis. Ein qualitativer Vergleich zwischen dem besten Ergebnis der vorhe-

Abbildung 5.17: Qualitative Darstellung der Auswirkung durch eine verringerte Anzahl Iterationen. TGV^2 mit SAD und einer Adaptiven Nachbarschaft der Größe 11×11. Von links nach rechts: **(a)** $smoothItr = 150$, $globalItr = 80$, $\eta = 0$. **(b)** $globalItr = 40$, $smoothItr^0 = 150$, $\eta = 2$.

rigen Berechnungen und dem der schrittweisen Reduktion der Iterationen, angewendet auf Frame 75 des Tsukuba-2012 Datensatzes, ist in Abbildung 5.17 gegeben. Erneut ist der Unterschied zwischen den beiden Ergebnissen qualitativ schwer zu erkennen. Zusammenfassend gilt, dass die Anzahl der Iterationen je nach Anwendungsfall entsprechend gewählt werden kann. Ist z. B. eine geringere Laufzeit wichtiger als die Genauigkeit des Modells, so kann eine geringere Anzahl an Iterationen gewählt werden, was sich stark auf die Laufzeit aber gering auf die Qualität der Tiefenkarten auswirkt.

	Iterationen	Frame 75	Frame 300	Frame 380
DAA [*cm*]	$globalItr = 80$, $smoothItr = 150$	12,23	29,86	**19,92**
	$globalItr = 80$, $smoothItr = 70$	12,30	**29,90**	20,02
	$globalItr = 40$, $smoothItr = 150$	**12,19**	29,97	20,14
	$globalItr = 40$, $smoothItr = 70$	12,25	30,03	20,25
	$globalItr = 40$, $smoothItr^0 = 150$, $\eta = 2$	12,21	29,97	20,17
Laufzeit [*s*]	$globalItr = 80$, $smoothItr = 150$	36,31	36,08	36,03
	$globalItr = 80$, $smoothItr = 70$	17,95	17,38	17,43
	$globalItr = 40$, $smoothItr = 150$	19,18	18,89	17,30
	$globalItr = 40$, $smoothItr = 70$	9,04	8,43	8,74
	$globalItr = 40$, $smoothItr^0 = 150$, $\eta = 2$	13,51	13,49	13,42

Tabelle 5.11: Quantitativer Vergleich verschiedener Anzahl an Iterationen in der Berechnung des Energiefunktionals der TGV^2-gestützten Rekonstruktion.

5.4 DISKUSSION

In den vorherigen Kapiteln wurden einige experimentelle Ergebnisse zu verschiedenen Parametrisierungen des Plane-Sweep- und des TGV^2-gestützten-Verfahrens vorgestellt. Im Folgenden soll nun diskutiert werden, welche der Einstellungen die quantitative und qualitativ besten Ergebnisse erzielen. Abschließend werden die finalen Ergebnisse der ausgewählten Parametrisierungen vorgestellt.

5.4.1 *Wahl der Ebenenparametrisierung*

Zunächst wurden in Kapitel 5.3.1 die Ergebnisse des isolierten Plane-Sweep-Verfahrens erläutert. Dabei widmet sich der erste Teil des Kapitels der Auswirkungen verschiedener Ebenenorientierungen und Abtastfrequenzen auf das Ergebnis. Wie in Abbildung 5.4 gezeigt, kann eine nicht-frontoparallele Orientierung der Ebenen genutzt werden, um planare Szenenobjekte besser zu rekonstruieren. Dazu muss jedoch zumindest teilweise bekannt sein, wie die Szene aufgebaut ist. Nicht nur qualitativ, auch quantitativ lässt sich für die Teilsequenzen um das Frame 300 des Tsukuba-2012 Datensatzes durch die Hinzunahme einer weiteren Orientierung eine Verbesserung erkennen (vlg. Tab. 5.3). Typischerweise besteht die Szene einer Luftaufnahme aus vielen planaren Objekten, wie beispielsweise dem Boden oder den Häuserdächern bzw. Häuserfassaden. Zudem sind diese Ebenen, je nach Blickrichtung der Kamera, nicht unbedingt frontoparallel orientiert. Aus diesem Grund kann eine Anpassung der Ebenenorientierung im Plane-Sweep-Verfahren für die Verwendung bei Luftaufnahmen durchaus nützlich sein und bessere Ergebnisse liefern. Jedoch sollte hierbei beachtet werden, dass eine weitere Ebenenorientierung auch eine Laufzeitsteigerung bewirkt. Somit muss abgewogen werden,

ob sich die eventuell nur geringe Verbesserung in der Qualität der Tiefenkarte, die zusätzliche Laufzeit wettmacht. Denn wie die Ergebnisse in Tabelle 5.3 zeigen sollte die zusätzliche Orientierung gut an die Szenenstruktur angepasst sein, um eine Verbesserung zu bewirken. Die alleinige Hinzunahme von willkürlich gewählten Orientierungen bewirkt lediglich einen Anstieg der Laufzeit.

Ein weiterer Parameter, von dem die Ebenen des Plane-Sweep-Verfahrens abhängen, ist die Distanz d der Ebene zum optischen Zentrum der Referenzkamera. Die Wahl der Abstände und damit die Frequenz mit der die Szene abgetastet wird, hat gemäß den Ergebnissen in Tabelle 5.4 eine, zumindest quantitativ, größere Auswirkung auf die Qualität der Tiefenkarten als die Wahl der Ebenenorientierungen. Dies lässt sich dadurch erklären, dass eine hohe Abtastfrequenz die fehlenden Ebenenorientierungen ausgleichen kann und auch nicht-frontoparallele Strukturen in der Szene gut rekonstruiert (vgl. Abb. 5.5). Qualitativ gesehen sind selbst bei einer sehr hohen Abtastfrequenz in Bereichen von nicht-frontoparallele Strukturen Abstufungen zu erkennen, welche bei einer entsprechend passenden Orientierung nicht auftreten würden. Aus den Zahlen in Tabelle 5.4 geht ebenfalls hervor, dass bei den Daten von Tsukuba-2012 eine adaptive Wahl der Ebenen in Abhängigkeit der Disparitätsänderungen keine quantitativen Verbesserungen bringen. Dennoch ist die adaptive Ebenenwahl je nach Datensatz notwendig, da hochfrequente Strukturen gemäß der Pixel abgetastet werden sollten um ein gutes Matching zu erhalten. Die besten Ergebnisse für die drei Teilsequenzen des Tsukuba-2012 Datensatzes, gerade auch unter Berücksichtigung der Laufzeit, sind mit einer Schrittweite von $4cm$ zwischen dein einzelnen Ebenen erreicht worden.

Schlussfolgernd gilt, dass die Wahl der Ebenenparametrisierungen nicht nur für die Qualität, aber auch für die Laufzeit des Verfahrens ausschlaggebend ist. Hierbei kann bei einer Kenntnis über den Aufbau der Szene die Parametrisierung entsprechend optimiert werden. So werden beispielsweise von Pollefeys *et al.* in [9] die Ergebnisse einer spärlichen Analyse der Szene verwendet um die Ebenen des Plane-Sweeps besser an die Strukturen anzupassen. Im Hinblick auf die Verwendung des umgesetzten Systems als Teil eines Structure-from-Motion Ansatzes kann die Szenenanalyse, die als Teil des Trackings durchgeführt wird, ebenfalls dazu genutzt werden, um eine optimierte Parametrisierung der Ebenen zu erreichen. Denn bei einer Kenntnis über den Aufenthaltsort der Szenenobjekte kann in den entsprechenden Bereichen häufiger und gezielter abgetastet werden. Bei einer fehlenden Information über den Aufbau der Szene ist eine frontoparallele Orientierung mit mäßiger Abtastung zu empfehlen. Dies hält die Laufzeit gering und erzielt gleichzeitig eine gute Rekonstruktion. Für die Rekonstruktion der Testsequenzen aus Tsukuba-2012 wird im Rahmen dieser Arbeit lediglich eine frontoparallele Ebenenorientierung sowie, eine Schrittweite von $4cm$ zwischen den einzelnen Ebenen gewählt.

5.4.2 *Anzahl der Eingangsbilder*

Zur Rekonstruktion verwendet das umgesetzte System mehr als zwei Einzelbilder um besser mit der Verdeckungsproblematik umzugehen /vgl. Kap. 4.3). Die Notwendigkeit der höheren

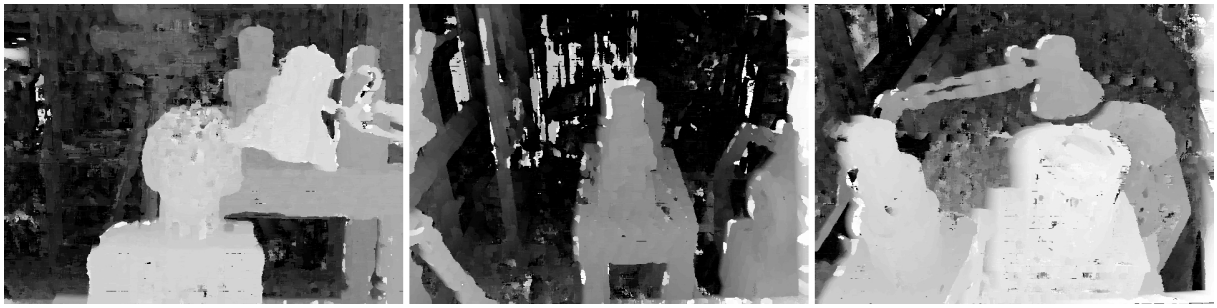

Abbildung 5.18: Berechnete Tiefenkarten der Testsequenzen mittels Plane-Sweep-Verfahren basierend auf 2 Eingangsbildern. Kostenfunktion: SAD mit einer Aggregationsnachbarschaft der Größe 11×11.

Anzahl an Eingangsdaten zeigt die Abbildung 5.18. Diese zeigt Tiefenkarten der jeweiligen Ausschnitte aus dem Tsukuba-2012 Datensatzes, welche mit nur zwei aufeinanderfolgenden Einzelbilder berechnet wurden. Hierbei ist die deutlich schlechtere Qualität zu erkennen. Gerade im Bereich von Tiefendiskontinuitäten, an denen Verdeckungen auftreten, sind weiße Stellen zu erkennen. Diese deuten auf Fehlzuordnungen von Bildkorrespondenzen hin, welche aufgrund der verdeckten Pixel zustande kommen. Bei der Verwendung von mehr als zwei Aufnahmen, können die entsprechenden Pixel entweder in der linken oder rechten Teilgruppe wiedergefunden werden, wodurch die Fehlzuordnungen vermieden werden.

In Kapitel 5.3.1.3 wird untersucht welchen Einfluss die Anzahl der verwendeten Eingangsbilder auf die Qualität der Rekonstruktion hat. Dabei geht hervor, dass sowohl das quantitativ, als auch qualitativ beste Ergebnis bei einer größeren Anzahl an Eingangsbildern (hier 11) erzielt wird. Zwar bedeutet dies zugleich auch eine längere Berechnungszeit, jedoch hat dies weniger Einfluss auf die Echtzeitfähigkeit des Systems, da je mehr Einzelbilder der Eingangssequenz verwendet werden, desto weniger Tiefenkarten müssen berechnet werden. Wird beispielsweise ein Eingangsvideo mit einer Bildrate von $30Hz$ verwendet, und zur Berechnung der Tiefenkarten jeweils 11 Einzelbilder herangezogen, so darf die Berechnungszeit ungefähr $1/3s$ benötigen um dennoch in Echtzeit eine Tiefenkarte zu erstellen. Zudem wirkt sich die Erhöhung der Eingangsbilder nicht so stark auf die Laufzeit aus wie die Steigerung der Abtastfrequenz oder die Hinzunahme mehrere Orientierungen.

Bei einer großen Anzahl an verwendeten Eingangsbilder besteht jedoch gleichzeitig die Gefahr, dass nicht alle Eingangsbilder die zu rekonstruierende Szene zu genüge zeigen. Gerade wenn nicht alle Einzelbilder der Sequenz verwendet werden kann es vorkommen, dass nicht genügend Überlappungen zwischen den Aufnahmen vorliegen. Gerade bei Luftaufnahmen, die eine hohe Auflösung aufweisen, kann dies vorkommen.

5.4.3 *Kostenfunktionen und Aggregationsnachbarschaften*

In den experimentellen Ergebnissen werden die beiden Kostenfunktionen, sowie verschiedene Aggregationsnachbarschaften einander gegenübergestellt. Sowohl als Teil des Plane-Sweep-Verfahrens als auch im Zusammenhang der TGV^2-gestützten Rekonstruktion. Die Ergebnisse des Plane-Sweeps sind in Abbildung 5.7 und in Tabelle 5.6 aufgelistet. Sie zeigen wie die Größe der Nachbarschaft je nach verwendeter Kostenfunktion andere Auswirkungen auf die Qualität der Tiefenkarte hat. Zunächst wird deutlich, dass bei beiden Kostenfunktionen mit zunehmender Größe der Nachbarschaft die Struktur der Szenenobjekte immer mehr verändert wird. Dies lässt sich auf das bereits mehrfach erwähnte Phänomen des „foreground-fattening" zurückführen. Um dies zu verhindern wurde als Teil des TGV^2-gestützten Verfahren eine adaptive Aggregationsnachbarschaft umgesetzt. Wie besonders in Abbildung 5.14 zu erkennen, bewirkt diese, dass selbst die kleinsten Strukturen, wie das Lampenkabel und die Schrauben im Kamerastativ rekonstruiert werden. Diese Verbesserung wird auch in Abbildung 5.19 deutlich. Darin sind zwei Ausschnitte des Fehlers zwischen der berechneten Tiefenkarten und der Groundtruth enthalten. Je dunkler die darin enthaltenen Pixel sind, desto größer ist der Fehler. Bei der Rekonstruktion ohne eine adaptive Aggregationsnachbarschaft (vgl. Ausschnitt *(a)*) sind die Objekte mit einem teilweise prägnanteren Rand umzogen als in *(b)*. Da das Ergebnis des Plane-Sweeps als Initialisierung für das TGV^2-Verfahren dient, wirkt sich diese Verdickung auch auf dessen Ergebnisse aus. So ist in *(b)* zu erkennen, dass dieser Rand zwar deutlich schwächer wurde, jedoch immer noch vorhanden ist. Die Ursache hierfür ist der Einfluss der initialen Tiefenkarte des Plane-Sweeps. Diese enthält die Verdickung der Objekte und überträgt diese mit in das TGV-Verfahren. Zwar führt die darin verwendete adaptive Nachbarschaft dazu, dass die Ränder nicht verstärkt werden, jedoch ist es durch die Regularisierung nicht möglich, diese Ränder gänzlich verschwinden zu lassen. Dies zeigt, dass die Qualität der initialen Tiefenkarte einen teilweise hohen Einfluss auf das Endresultat des TGV-Verfahrens hat.

Ein weiterer Aspekt, der beim Vergleich der Kostenfunktionen deutlich wird, ist dass die alleinige Verwendung der Hammingdistanz der Census-Transformation (CT) als Kostenfunktion nicht gut geeignet ist. Zwar ist die CT in manchen Fällen robuster als die Absoluten Differenzen jedoch haben die Bitfolgen eine deutlich geringere Aussagekraft als die Pixelintensitäten. Des Weiteren ist die CT anfällig gegenüber starken Veränderungen des Referenzpixels. Ändert sich dieser so sehr, dass sich die Verhältnisse zu den Nachbarpixeln ändern, so hat dies große Auswirkungen auf die Bitfolgen. Die führt unweigerlich zu Fehlzuordnungen und dadurch zu Mehrdeutigkeiten in der Tiefenschätzung. Gerade auch in Bereichen von homogenen Bildstrukturen weißt die CT Schwachstellen auf, da alle Pixel innerhalb des Bereiches dieselbe Intensität haben können. Hierdurch kommt es besonders auch innerhalb von Szenenobjekten zu Fehlzuordnungen und damit zu Sprüngen in den Tiefenkarten. In [20] und [46] werden robustere Varianten der CT als Kostenfunktion verwendet. Dabei verwendet Kuschk in seiner Abhandlung für den Wert des Referenzpixels einen Mittelwert aus den umliegenden Pixeln. Dies soll die CT robuster gegenüber Schwankungen in der Intensität des Referenzpixels ma-

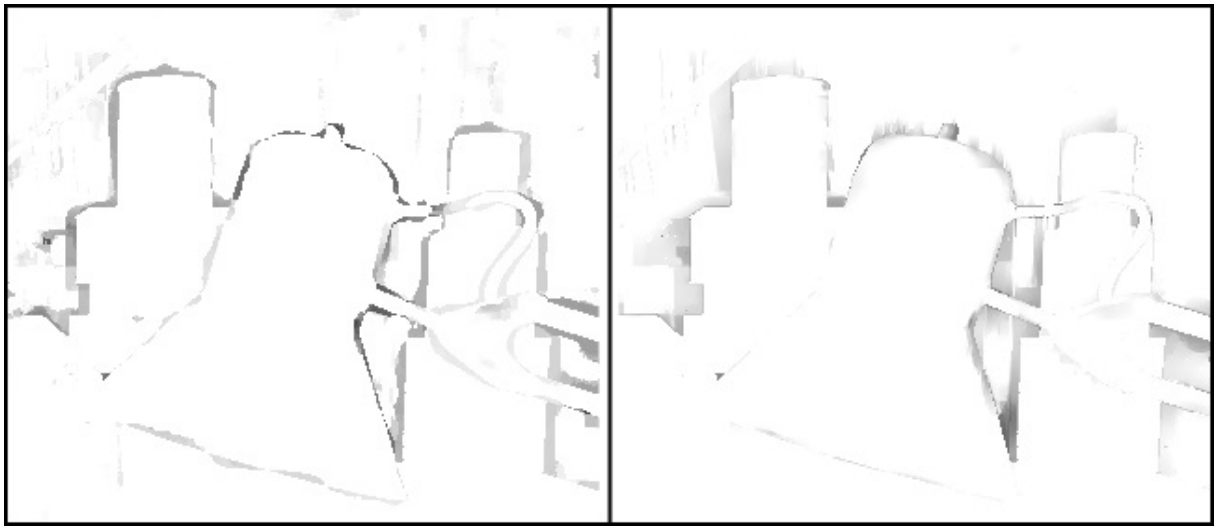

Abbildung 5.19: Gegenüberstellung des Fehlers der beiden Verfahren. Fehler als absolute Differenz zwischen Groundtruth und berechnetem Ergebnis. Dunkel entspricht großem Fehler, Hell entspricht kleinem Fehler. Links: das Ergebnis des Plane-Sweep-Verfahrens ohne Adaptiver Aggregationsnachbarschaft. Recht: Ergebnis des TGV^2-Verfahrens mit Adaptiver Aggregationsnachbarschaft. Tiefenkarte des Plane-Sweeps dient als Initialisierung des TGV^2-Verfahrens.

chen. Peris *et al.* hingegen kombinieren die CT mit den Absoluten Differenzen. Der Einfluss beider Kosten kann dabei unterschiedlich gewichtet werden. Hierdurch soll die Aussagekraft der CT durch die Intensitäten der Pixel verstärkt werden.

Wie in Abbildung 5.15 deutlich wird, ist es bei einer alleinigen Verwendung der CT selbst durch die Regularisierung nicht möglich die Mehrdeutigkeiten zu reduzieren. Lediglich durch die in dieser Arbeit eingeführten adaptiven Gewichtung des Datenterms kann die Hammingdistanz der CT als Kostenfunktion im Rahmen dieses Systems verwendet werden. Die Anpassung der Gewichtung des Datenterms erlaubt eine deutlich stärkere Regularisierung innerhalb eines Szenenobjektes bei einer gleichzeitigen Hervorhebung der Objektkanten. Diese Adaptivität kann dabei auch bei der Verwendung anderer Kostenfunktionen, wie z. B. die Summe der absoluten Differenzen, nützlich sein.

5.4.4 *Regularisierung*

Als Teil der TGV^2-gestützten Rekonstruktion wird eine Regularisierung verwendet um die Sprünge in den Tiefenkarten, welche beim Abgleich von Bildkorrespondenzen auftreten, auszugleichen. Durch die Regularisierung wird bewirkt, dass das Modell glatt wird und große Tiefenunterschiede zwischen benachbarten Pixel vermieden werden. Jedoch sollte die Regularisierung nur innerhalb von einem Objekt erfolgen, damit Tiefendiskontinuitäten zwischen einzelnen Szenenobjekten erhalten werden. Aus diesem Grund sind im Rahmen dieser Arbeit

verschiedene anisotrope Gewichtstensoren untersucht worden, die diese adaptive Regularisierung bewirken sollen. Dabei basieren die Tensoren auf Gewichtsfunktionen, welche die Gewichtungen aufgrund der auftretenden Bildgradienten bestimmen. Bei einem großen Bildgradienten wird davon ausgegangen, dass eine Objektkante vorliegt und somit die Regularisierung reduziert werden soll.

Die drei Gewichtsfunktionen, die hierbei verwendet wurden sind der Perona-Malik-Term, der Charbonnier-Term und die allgemeine Exponentialfunktion. Die ersten beiden Funktionen werden ursprünglich im Rahmen der nicht-linearen Diffusion verwendet und steuern dabei die Strömung der Diffusion. Dies bedeutet, dass sie nicht die Gewichtung auf Basis des Eingangsbildes bestimmen, sondern auf Basis der, durch die Diffusion entstehende Strömung. Im Rahmen dieser Arbeit werden die Gewichtsfunktionen jedoch auf die Eingangsbilder angewendet, was sie zu bild-gesteuerten Gewichtsfunktionen macht. Im Zusammenhang der nicht-linearen Diffusion wurde in festgestellt (vgl. [35]), dass der Perona-Malik-Term zu numerischen Fehlern führt. Abhilfe hierfür sollte der Charbonnier-Term schaffen. Da die Terme als Teil der Rekonstruktion aber bild-gesteuert eingesetzt werden, treten diese Fehler nicht auf. Im Gegenteil: aufgrund der steiler abfallenden Funktion des Perona-Malik-Terms eignet sich dieser besser für die Steuerung der Regularisierung. Dies zeigt sich auch in den experimentellen Ergebnisse aus Tabellen 5.8 und 5.7. Die Ergebnisse von $\lambda_{ch} = 3$ ähneln denen von $\lambda_{pm} = 2\lambda_{ch} = 6$.

Abbildung 4.2 zeigt, dass der Kurvenverlauf des Perona-Malik- und Charbonnier-Terms stark dem Verlauf der Exponentialfunktion ähnelt. Je nach Parametrisierung kann die Kurve der e-Funktion aber noch steiler abfallen. Wie die Ausschnitte in Abbildung 5.9 zeigen, führt eine steilerer Funktionsverlauf zu schärferen Kanten an Tiefendiskontinuitäten. Jedoch führt dies auch zu einer strengen Anisotropie der Tensoren, wodurch auch innerhalb von Szenenobjekten die Regularisierung ausgesetzt wird. Je nach Anwendungsfall muss hierbei abgewogen werden, ob lieber eine strikte Anisotropie genutzt werden soll um die Kanten scharf zu erhalten oder ob durch einen schwächeren Gewichtstensor über mehr Kanten hinweg regularisiert werden soll.

Um die Kanten an Szenenobjekten noch besser hervorzuheben wurde ein Liniensegment-Detektor (LSD) verwendet um die Regularisierung an den gefundenen Kanten noch stärker zu reduzieren. Dies soll zu präziseren und glatteren Tiefendiskontinuitäten an Szenenobjekten führen (vgl. Abb. 5.12). Wichtig ist dabei, dass für die Erstellung der Gewichtstensoren auf Basis der Liniensegmente möglichst steile Gewichtsfunktionen verwendet werden. Am besten eignet sich hierfür eine entsprechend parametrisierte Exponentialfunktion. Während hierdurch zwar an manchen Stellen die Hervorhebung der Kanten verbessert wird, kann es an anderen Stellen auch das Ergebnis verschlechtern. So werden zum Beispiel auch Liniensegmente innerhalb von Szenenobjekten gefunden, wodurch an den entsprechenden Stellen keine vollständige Regularisierung stattfinden kann. An anderen Stellen werden dabei vorhandene Liniensegmente nicht detektiert. Gerade am oberen Rand der Lampe führt dies zum Ausfransen der Kante, da lediglich entlang der vertikalen Linien, die durch die im Hintergrund platzierten Bücher entstehen, regularisiert wird. Die Tiefenschätzung der Lampe wird

dabei in den Bereich des Hintergrundes übertragen. Somit sollte in zukünftigen Arbeiten untersucht werden, ob die Integration des LSDs auf Basis der Eingangsdaten nötig ist. Bei einer Rekonstruktion eines urbanen Gebietes kann der LSD bewirken, dass die Häuserkanten und -fassaden besser Rekonstruiert werden. Bei einer Szene wie die des Tsukuba-2012 Datensatzes führt eine Verwendung des LSD teilweise eher zu Verschlechterungen.

5.5 ABSCHLIESSENDE ERGEBNISSE

Nachdem in den letzten Kapiteln die Ergebnisse verschiedener Konfigurationen erläutert und diskutiert wurden, sollen nun die finalen Ergebnisse der Arbeit vorgestellt werden. Dazu wurde anhand der vorangegangenen Untersuchungen eine Konfiguration bestimmt, mit der die qualitativ und quantitativ besten Ergebnisse erzielt werden.

Die Vorstellung der endgültigen Ergebnisse erfolgt zunächst für das isolierte Plane-Sweep-Verfahren, das eine Tiefenkarte echtzeitnah erstellt. Hierbei wurden zur Rekonstruktion drei Sequenzen aus dem Tsukuba-2012 Datensatz mit jeweils 11 Einzelbilder verwendet. Die Kostenfunktion ergibt sich aus den Summen der Absoluten Differenzen (SAD) mit einer Aggregationsnachbarschaft der Größe 11×11. Da als Teil des isolierten Plane-Sweeps keine Regularisierung verwendet wird, erzielt die SAD als Kostenfunktion bessere Ergebnisse als die Hammingdistanz der Census-Transformation (CT). Zudem wurde die Szene nur mit einer frontoparallelen Ebenenorientierung mit $\vec{n} = (0, 0, 1)^T$ abgetastet. Die Ebene wurde dabei von $d_{max} = 300cm$ bis $d_{min} = 30cm$ mit einer Schrittweite von $4cm$ durch die Szene verschoben. Abbildung 5.20 enthält die erzielten Ergebnisse mit entsprechender Groundtruth. Die Laufzeiten der Berechnung sowie die DAA für die drei Tiefenkarten sind in Tabelle 5.12 aufgelistet.

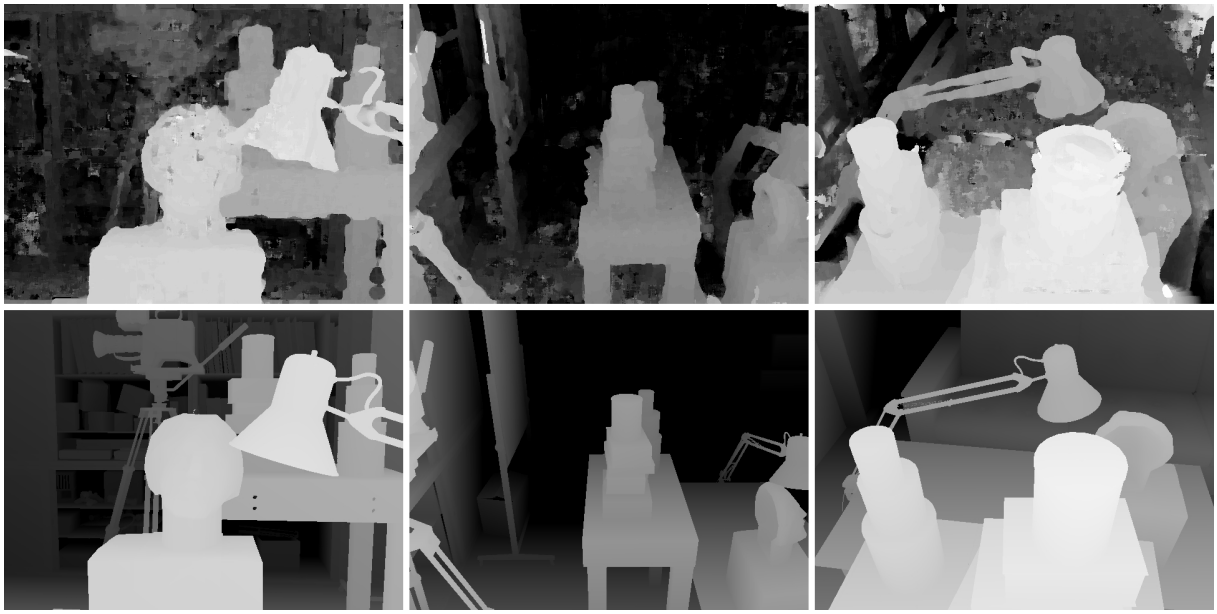

Abbildung 5.20: Abschließende Ergebnisse des isolierten Plane-Sweep-Verfahrens mit entsprechender Groundtruth. Abtastung durch frontoparallele Ebene mit $\vec{n} = (0, 0, 1)^T$, $d_{max} = 300cm$, $d_{min} = 30cm$ und einer Schrittweite von $4cm$. Kostenfunktion: SAD mit einer Nachbarschaftsgröße von 11×11. Von links nach rechts: **(a)** Tiefenkarte zu Frame 75. **(b)** Frame 300. **(c)** Frame 380.

Die quantitativen Ergebnisse zeigen, dass die Berechnungen auf einer leistungsstarken Desktop-Hardware durchaus in Echtzeit durchgeführt werden. Gemäß der genannten Berech-

nungszeit, werden durch das Plane-Sweep-Verfahren Tiefenkarten mit ~10Hz erstellt. Die Qualität der Tiefenkarten ist dabei zufriedenstellend. Es wird bereits eine gute Rekonstruktion der Szene erreicht, jedoch weisen die Tiefenkarten noch Fehler auf. So ist das Phänomen des „foreground-fattening“, sowie die Sprünge innerhalb der Szenenobjekte deutlich zu erkennen.

	Frame 75	Frame 300	Frame 380
$DAA\ [cm]$	14,35	31,25	23,03
$Laufzeit\ [s]$	0,11	0,10	0,10
Durchschnittliche Tiefe $[cm]$	186,52	256,89	161,90

Tabelle 5.12: Quantitative Ergebnisse des Plane-Sweep-Verfahrens in abschließender Konfiguration.

Als Nächstes werden die abschließenden Ergebnisse der TGV^2-gestützten Rekonstruktion dargestellt. Der anisotrope Gewichtstensor basiert dabei auf den Perona-Malik-Term mit $\lambda_{pm} = 3$. Wie in der Diskussion erläutert eignet sich im Zusammenhang des Tsukuba-2012 Datensatzes eine Erweiterung des Tensors durch den Liniensegment-Detektor nicht. Weswegen dieser auch in den abschließenden Ergebnissen nicht angewendet wurde. Die Gewichtstensoren sind auf Basis eines mit $\sigma = 0,5$ vorgeglätteten Referenzbildes aufgebaut. Abbildung 5.21 zeigt die Ergebnisse für die Berechnung mittels SAD als Kostenfunktion, sowie mittels der Hammingdistanz der CT. In beiden Fällen wurde eine adaptive Aggregationsnachbarschaft der Größe 21×21, sowie eine adaptive lokale Gewichtung des Datenterms verwendet. Bei der Verwendung der SAD ist $\lambda_s = 0,2$ und $\lambda_d = 1$ gewählt. Für die Hammingdistanz der CT gilt $\lambda_s = 0,2$ und $\lambda_d = 0,2$. Die Anzahl der Iterationen zur Lösung des Energiefunktionals sind gemäß $globalItr = 80$ und $smoothItr = 150$ gewählt. Die quantitativen Ergebnisse dieser Konfiguration sind in Tabelle 5.13 enthalten.

		Frame 75	Frame 300	Frame 380
SAD	$DAA\ [cm]$	12,07	29,44	20,41
	$Laufzeit\ [s]$	38,84	35,84	35,48
CT	$DAA\ [cm]$	12,79	31,02	20,90
	$Laufzeit\ [s]$	37,19	37,22	37,54
	Durchschnittliche Tiefe $[cm]$	186,52	256,89	161,90

Tabelle 5.13: Quantitative Ergebnisse des TGV^2-Verfahrens in abschließender Konfiguration angewendet auf den Tsukuba-2012 Datensatz.

Die qualitativen Ergebnisse zeigen im Vergleich zu denen des Plane-Sweeps eine deutlich glattere und präzisere Rekonstruktion. Es werden feine Details im Kamerastativ oder im Arm der Lampe sichtbar. Während auch die Objektkanten präziser rekonstruiert wurden, scheint eine Art Nimbus die Objekte der Tiefenkarten in Abbildung 5.21 zu umgeben. Diese

„Wolke“ tritt besonders bei feinen Objekten wie dem Lampenkabel auf, bei denen die initiale Rekonstruktion zum „foreground-fattening“ geführt hat. Diese Verdickung ist durch die Regularisierung nicht ganz zu entfernen, wird jedoch aber deutlich verschwommen.

Wie zu Anfang des Auswertungskapitels erwähnt, wird das TGV^2-Verfahren ebenfalls auf ausgewählte Datensätze des Middlebury Benchmarks angewendet, um die Ergebnisse dieser Arbeit mit denen aus [19] zu vergleichen. Hierbei ist die Konfiguration gleich gewählt wie bei der Anwendung auf den Tsukuba-2012 Datensatz. Die qualitativen und quantitativen Ergebnisse sind in Abbildung 5.22 bzw. Tabelle 5.14 dargestellt. Da der Middlebury Benchmark nicht repräsentativ zur Auswertung der Laufzeit ist, wird diese in den Ergebnissen nicht aufgelistet. Zudem ist zu beachten, dass die Angaben zu der DAA in Pixel sind.

	Tsukuba	Venus	Teddy	Cones
SAD	1,22	1,08	0,90	0,89
CT	1,28	1,11	0,96	0,91
Durchschnittliche Disparität $[px]$	10,76	8,88	13,41	16,12

Tabelle 5.14: DAA in Pixel des TGV^2-Verfahrens in abschließender Konfiguration angewendet auf den Middlebury Benchmark.

Zur Auswertung verwenden Kuschk und Cremers in [19] das Verhältnis der Pixel deren Abweichung zur Groundtruth > 1 ist. Zur Vergleichbarkeit wird das gleiche Qualitätsmaß für die Ergebnisse dieser Arbeit berechnet und mit denen aus [19] vergleichen (vgl. Tab. 5.15). Während die qualitativen Ergebnisse aus Abbildung 5.22 zufriedenstellend sind, suggerieren die Zahlen aus Tabelle 5.15, gerade bei dem Datensatz „Tsukuba“ und „Venus“, dass die erzielten Ergebnisse viel schlechter ausfallen. Dies lässt sich dadurch erklären, dass die auftretenden Disparitäten innerhalb und zwischen den Objekten durchaus konsistent sein können, was zu einem qualitativ guten Ergebnis führt. Eine globale Abweichung zur Groundtruth verfälscht jedoch das quantitative Ergebnis. Das Verhältnis des „Teddy“ und „Cones“ Datensatzes entspricht hingegen eher dem qualitativen Ergebnissen aus Abbildung 5.22 und ist dabei nur etwas schlechter als das aus [19]

	Tsukuba	Venus	Teddy	Cones
Ergebnisse dieser Arbeit	31,3%	48,05%	14,79%	12,33%
Ergebnisse aus [19]	4,33%	1,00%	9,66%	11,1%

Tabelle 5.15: Verhältnis der fehlerhaften Pixel, bei denen die Absolute Differenz zur Groundtruth > 1 ist. Vergleich zwischen Ergebnisse dieser Arbeit und der Ergebnisse aus [19].

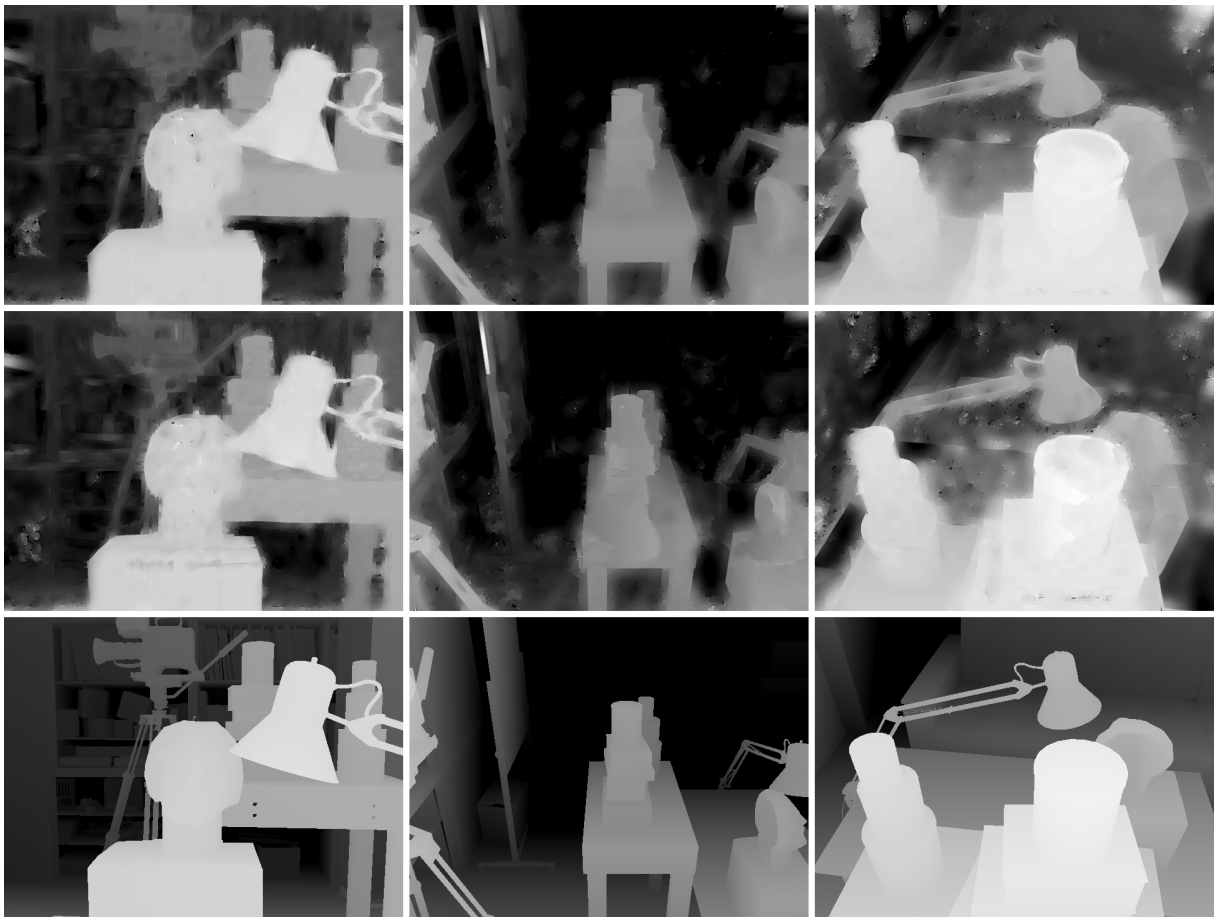

Abbildung 5.21: Abschließende Ergebnisse des TGV^2-Verfahren angewendet auf die Testsequenzen des Tsukuba-2012 Datensatzes. Gewichtstensoren basierend auf Perona-Malik-Term mit $\lambda_{pm} = 3$, ohne Erweiterung durch den LSD, berechnet auf vorgeglättetes Eingangsbild mit $\sigma = 0,5$. Verwendung einer adaptiven Aggregationsnachbarschaft der Größe 21×21. Anwendung der lokalen adaptiven Gewichtung des Datenterms. **Reihe** 1: SAD als Kostenfunktion mit $\lambda_s = 0,2$ und $\lambda_d = 1$. **Reihe** 2**:** Hammingdistanz der CT mit $\lambda_s = 0,2$ und $\lambda_d = 0,2$. **Reihe** 3**:** Groundtruth.

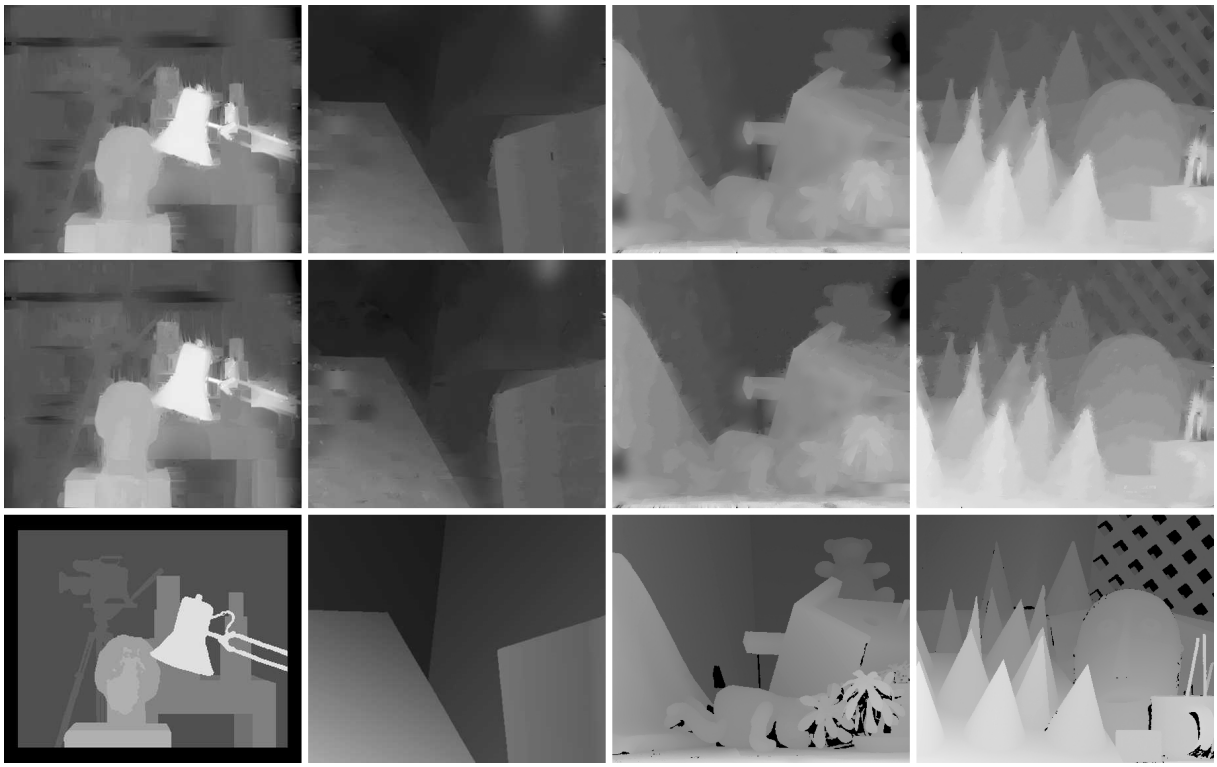

Abbildung 5.22: Abschließende Ergebnisse des TGV^2-Verfahren angewendet auf die ausgewählten Testdaten des Middlebury Benchmarks. Gewichtstensoren basierend auf Perona-Malik-Term mit $\lambda_{pm} = 3$, ohne Erweiterung durch den LSD, berechnet auf vorgeglättetes Eingangsbild mit $\sigma = 0,5$. Verwendung einer adaptiven Aggregationsnachbarschaft der Größe 21×21. Anwendung der lokalen adaptiven Gewichtung des Datenterms. **Reihe** 1: SAD als Kostenfunktion mit $\lambda_s = 0,2$ und $\lambda_d = 1$. **Reihe** 2**:** Hammingdistanz der CT mit $\lambda_s = 0,2$ und $\lambda_d = 0,2$. **Reihe** 3**:** Groundtruth.

6. FAZIT

Im Rahmen dieser Arbeit ist ein Framework zur echtzeitnahen 3D-Rekonstruktion mittels Structure-from-Motion umgesetzt worden. Das System ist dabei primär für die Rekonstruktion von urbanen Gebieten auf Basis von Luftaufnahmen vorgesehen. Da sich diese Arbeit lediglich der dichten Rekonstruktion widmet, wurde vorausgesetzt, dass die Eingangsdaten in Form von Einzelbildern mit entsprechenden Kameraposen gegeben sind. Auf Basis dieser Einzelbilder und Posen, führt das umgesetzte Framework eine dichte Tiefenschätzung für einzelne Keyframes der Eingangssequenz durch. Die resultierende Tiefenschätzung wird dabei in einzelnen Tiefenkarten gespeichert. Diese können anschließend fusioniert und in ein dreidimensionales Modell projiziert werden. Die globale Fusion und Modellerstellung ist dabei ebenfalls nicht als Teil der Arbeit vorgesehen.

Aus aktuellen Publikationen wurden für die Rekonstruktion zwei Ansätze ausgewählt, die sich ebenfalls mit der Rekonstruktion von urbanen Gebieten beschäftigen. Zum einem ist das Plane-Sweep-Verfahren von Pollefeys *et al.* aus [9] betrachtet worden. Dieses beschäftigt sich mit der Rekonstruktion von Häuserfassaden auf Basis eines Videos, welches von einem bodengestützten System aufgenommen wurde. Für eine akkurate Rekonstruktion wird darin ein erweiterter Plane-Sweep-Ansatz verwendet, der mehrere Ebenenorientierungen berücksichtigt um somit die verschiedenen Ausrichtungen der Häuser besser nachzubilden. Die Verwendung verschiedener Ebenenorientierung ist dabei auch für die Rekonstruktion aus Luftbilddaten sehr gut geeignet, da somit verschiedene Blickrichtungen der Kamera, sowie unterschiedliche Geländeformen berücksichtigt werden können. Als zweites Verfahren wurde der Ansatz von Kuschk und Cremers aus [19] zur Variations-basierten Rekonstruktion mittels der Verallgemeinerten-Total-Variation zweiter Ordnung (TGV^2) verwendet. Im Gegensatz zur Total-Variation (TV) erster Ordnung, die das Modell lediglich aus konstanten frontoparallelen Funktionen aufbaut, werden durch die TGV^2 affine Funktionen in der Rekonstruktion berücksichtigt. Dadurch lassen sich geneigte Oberflächen, wie z. B. Häuserdächer, besser rekonstruieren. Bei der TV erster Ordnung werden geneigte Oberflächen durch stückweise konstante Funktionen rekonstruiert, was zu „Staircasing"-Artefakten führt. Der Ansatz aus [19] wurde von Kuschk und Cremers zur Rekonstruktion eines urbanen Gebietes auf Basis von Luftaufnahmen angewendet und erzielt dabei sehr gute Ergebnisse.

Ein Nachteil des TGV^2-gestützten Verfahrens ist die fehlende Echtzeitfähigkeit. Weswegen das, in dieser Arbeit umgesetzte Framework in zwei Module unterteilt ist: Dem Plane-Sweep-Modul zur echtzeitfähigen On-the-fly Berechnung und dem TGV^2-Modul zur Offline-Verfeinerung. Ersteres basiert dabei auf ein Plane-Sweep-Verfahren ähnlich dem aus [9]. Je nach Konfiguration ist das Verfahren dabei echtzeitfähig und kann dichte Tiefenkarten noch während der Erfassung des Eingangsvideos (On-the-fly) berechnen. In dem daraus resultierenden Modell gehen jedoch viele Details verloren, da die Objekte auf die einzelnen Ebenen, mit

denen die Szene abgetastet wird, reduziert werden. Zudem enthält die Tiefenkarte aufgrund der diskreten Eigenschaft des Plane-Sweep-Verfahrens große Sprünge innerhalb der Szenenobjekte. Um die Qualität der Tiefenkarte zu verbessern wird diese an das zweite Modul weitergeleitet und zwischengespeichert. Nach Beendigung der Berechnung in Echtzeit durch das erste Modul, wird die TGV^2-gestützte Rekonstruktion im zweiten Modul des Frameworks dazu verwendet die ersten Tiefenkarten weiter zu verfeinern. Das daraus resultierende Modell soll auch kleine Details erhalten und eine starke Homogenität innerhalb der Objekte aufweisen.

Aus der abschließenden Evaluation des umgesetzten Frameworks gehen einige wichtige Aspekte hervor, die für eine Weiterentwicklung und Verbesserung berücksichtigt werden sollten. Zunächst wurden unter anderem verschiedene Ebenenorientierungen und Abtastfrequenzen im Zusammenhang des Plane-Sweep-Verfahrens untersucht. Zwar begünstigt die Anpassung der Ebenenorientierung an nicht-frontoparallelen planaren Objekte deren Rekonstruktion, jedoch sollte dabei bekannt sein wie diese Objekte im Raum orientiert sind. Eine Hinzunahme willkürlicher Orientierungen und Verschiebungsrichtungen der Ebenen führt zu keiner Verbesserung der Rekonstruktion. Auch bei der Erhöhung der Abtastfrequenz werden bessere Ergebnisse erzielt, wenn diese gezielt eingesetzt wird. Ist eine Kenntnis über den Aufbau der Szene vorhanden, so können die Orientierungen und die Häufigkeit der Abtastung an die Szene angepasst werden. Dabei ist in beiden Fällen zu beachten, dass die Hinzunahme von zusätzlichen Ebenen die Echtzeitfähigkeit des Verfahrens verletzen kann.

Des Weiteren ist die adaptive Regularisierung als Teil des TGV^2-Verfahrens evaluiert worden. Während die Tiefenschätzungen innerhalb von Szenenobjekten regularisiert, also an die umliegenden Schätzungen angepasst werden soll, ist die Glättung an Objektgrenzen auszusetzen um Tiefendiskontinuitäten zu erhalten. Für diesen Zweck enthält das Energiefunktional zur TGV^2-gestützten Rekonstruktion einen anisotropen Gewichtstensor, durch den die Regularisierung an die Bildstrukturen angepasst werden soll. Dieser wird auf Basis des lokalen Bildgradienten aufgebaut und soll bei einem betraglich hohen Gradienten die Regularisierung in Richtung des Gradienten aussetzen. In [19] ist zusätzlich ein Liniensegment-Detektor (LSD) integriert um die Tiefendiskontinuitäten an Objektgrenzen präziser zu rekonstruieren. Im Rahmen dieser Arbeit wurde die Auswirkung verschiedener Konfigurationen der Gewichtstensoren, sowie des LSDs untersucht. Dabei ging hervor, dass je nach Datensatz die Integration des LSDs keine oder nur geringe Verbesserung bringt. So können zum Beispiel Objekte ausfransen, wenn der LSD Liniensegmente detektiert, die nicht parallel zur Objektkante verlaufen. Während die Erweiterung durch den LSD im Zusammenhang von geradlinigen Strukturen wie Gebäuden Verbesserung bringt, ist sie für den zur Evaluation verwendeten Datensatz nicht praktikabel. Die quantitativ und qualitativ besten Ergebnisse wurden mit einem Gewichtstensor basierend auf einer steilen Gewichtsfunktion, der auf einem mittels Gaußkern geglätteten Eingangsbild berechnet und ohne den LSD erweitert wurde, erzielt.

Die dritte Erkenntnis die aus der Auswertung der Verfahren gezogen werden kann, betrifft die verwendeten Kostenfunktionen zum Abgleich von Bildstrukturen. In dieser Arbeit wurden die Summe der Absoluten Differenzen (SAD), sowie die Hammingdistanz der Census-

Transformation (CT) als Kostenfunktionen verwendet. Während die SAD alleinig auf den Intensitäten der Pixel beruht, basiert die Hammingdistanz der CT auf dem Verhältnis der Pixelintensitäten innerhalb einer Nachbarschaft. Dies soll die CT-basierte Kostenfunktion robuster gegenüber Beleuchtungsänderungen machen. Jedoch zeigen die Untersuchungen, dass die CT gerade innerhalb homogener Bereiche zu häufigeren Mehrdeutigkeiten in der Suche nach Punktkorrespondenzen führt. Diese Mehrdeutigkeiten in der Zuordnung werden selbst unter Verwendung der Regularisierung im Energiefunktional der TGV^2 nicht reduziert, was zu Artefakten unter der Verwendung der CT führt. Aus diesem Grund wurde in dieser Arbeit eine zusätzliche lokale adaptive Gewichtung des Datenterms eingeführt, welche dessen Einfluss innerhalb der Szenenobjekte weiter reduzieren soll. Diese Gewichtung basiert dabei auf dem gleichen Prinzip wie die lokale anisotrope Steuerung der Regularisierung. Eine Gewichtsfunktion, die zu der, für die anisotrope Gewichtung des Regularisierungsterm verwendete Funktion invertiert ist, soll dabei die Gewichtung des Datenterms in Abhängigkeit des Bildgradienten bestimmen. Hierbei wird der Einfluss des Datenterms innerhalb eines Objektes durch kleine Bildgradienten reduziert, bei einer gleichzeitigen Erhaltung des Einflusses in Bereichen mit großen Bildgradienten.

Die Verwendung der lokalen Gewichtung des Datenterms erlaubt eine weitere Reduzierung der Regularisierung, wodurch innerhalb der Objekte weiterhin genügend geglättet wird, aber die Objektkanten besser erhalten werden. Die besten Ergebnisse werden dabei unter der Verwendung der SAD mit einer lokalen adaptiven Gewichtung des Datenterms erzielt.

6.1 AUSBLICK

Die gewonnenen Erkenntnisse erlauben eine gezielte Verbesserung des Frameworks zur echtzeitnahen Rekonstruktion. Zunächst soll dabei das Framework in die Structure-from-Motion (SfM) Pipeline eingegliedert werden. Dabei erlaubt ein zusätzliches Verfahren zur Schätzung der Kamerabewegung, bereits vor der eigentlichen Rekonstruktion eine spärliche Analyse der zu rekonstruierenden Szene. Die daraus gewonnenen Informationen können dazu genutzt werden die Abtastrichtungen und -frequenzen des Plane-Sweep-Verfahrens besser an die Szene anzupassen. Dies erlaubt eine bessere Berücksichtigung der Szenenobjekte und die damit verbundene Verbesserung der Rekonstruktion. Außerdem zeigt die Auswertung, dass die Ergebnisse des Plane-Sweep-Verfahrens sich auch auf die Tiefenkarten der TGV^2-gestützten Rekonstruktion auswirken. Dies macht es umso wichtiger die Qualität der initialen Rekonstruktion zu verbessern. So kann zum Beispiel durch die Integration einer adaptiven Aggregationsnachbarschaft oder einer Subtiefenverfeinerung in das isolierte Plane-Sweep-Verfahren dessen Ergebnis verbessert werden.

Des Weiteren zeigt der Vergleich zwischen den Ergebnissen des TGV^2-Verfahrens dieser Arbeit und der von Kuschk und Cremers [19], dass die Qualität dieser Umsetzung noch deutlich gesteigert werden kann. So soll unter anderem die Berechnung der anisotropen Gewichtstensoren weiter verbessert werden. Vor allem die Integration des LSDs kann verfeinert werden, um weniger aber dafür aussagekräftigere Linsensegmente zu detektieren und die Ge-

wichtstensoren entsprechend anzupassen. Zudem soll untersucht werden, ob eine zuverlässigere Aussage über die lokalen Bildstrukturen, wie z. B. der Strukturtensor, genutzt werden kann um die anisotrope Steuerung der Regularisierung zu verbessern.

Gerade auch im Zusammenhang der Kostenfunktion basierend auf der Census-Transformation (CT) können verschiedene Erweiterungen die Qualität der Rekonstruktion steigern. So werden beispielsweise in [20] und [46] Erweiterungen verwendet, die die CT robuster machen sollen. Auch eine Verwendung anderer Kostenfunktionen wie beispielsweise Local-Binary-Patterns oder Mutual Information sollen geprüft werden.

Schlussendlich sollen vor allem durch die bereits erwähnte Eingliederung des Verfahrens in die SfM-Pipeline die Tiefenkarten der einzelnen Keyframes miteinander fusioniert und in ein globales Modell projiziert werden. Das Ziel ist die Entwicklung eines vollständigen Systems zur echtzeitnahen Erstellung eines genauen 3D-Modells aus Luftaufnahmen, das zu Unterstützung von Einsatzkräften in Krisengebieten verwendet werden kann.

LITERATUR

[1] R. Zabih und J. Woodfill, "A non-parametric approach to visual correspondence", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994.

[2] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera", in *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, 2003, S. 1403–1410. DOI: 10.1109/ICCV.2003.1238654. Adresse: http://doi.ieeecomputersociety.org/10.1109/ICCV.2003.1238654.

[3] G. Klein und D. Murray, "Parallel tracking and mapping for small AR workspaces", in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, 2007.

[4] H. Strasdat, J. M. M. Montiel und A. Davison, "Scale drift-aware large scale monocular slam", in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, 2010.

[5] J. Engel, T. Schöps und D. Cremers, "LSD-SLAM: large-scale direct monocular SLAM", in *European Conference on Computer Vision (ECCV)*, 2014.

[6] R. T. Collins, "A space-sweep approach to true multi-image matching", 1996.

[7] R. Yang und M. Pollefeys, "Multi-resolution real-time stereo on commodity graphics hardware.", in *CVPR (1)*, IEEE Computer Society, 17. Aug. 2004, S. 211–220, ISBN: 0-7695-1900-8. Adresse: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2003-1.html#YangP03.

[8] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang und M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions.", in *CVPR*, IEEE Computer Society, 6. Sep. 2007. Adresse: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2007.html#GallupFMYP07.

[9] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch und H. Towles, "Detailed real-time urban 3d reconstruction from video", *Int. J. Comput. Vision*, Bd. 78, Nr. 2-3, S. 143–167, Juli 2008, ISSN: 0920-5691. DOI: 10.1007/s11263-007-0086-4. Adresse: http://dx.doi.org/10.1007/s11263-007-0086-4.

[10] C. Häne, L. Heng, G. H. Lee, A. Sizov und M. Pollefeys, "Real-time direct dense matching on fisheye images using plane-sweeping stereo", in *2nd International Conference on 3D Vision, 3DV 2014, Tokyo, Japan, December 8-11, 2014*, 2014, S. 57–64. DOI: 10.1109/3DV.2014.77. Adresse: http://dx.doi.org/10.1109/3DV.2014.77.

[11] S. N. Sinha, D. Scharstein und R. Szeliski, "Efficient high-resolution stereo matching using local plane sweeps", in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Ser. CVPR '14, Washington, DC, USA: IEEE Computer Society, 2014, S. 1582–1589, ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.205. Adresse: http://dx.doi.org/10.1109/CVPR.2014.205.

[12] A. Chambolle, M. Novaga, D. Cremers und T. Pock, "An introduction to total variation for image analysis", in *in Theoretical Foundations and Numerical Methods for Sparse Recovery, De Gruyter*, 2010.

[13] B. K. Horn und B. G. Schunck, "Determining optical flow", Cambridge, MA, USA, Techn. Ber., 1980.

[14] T. Brox, A. Bruhn, N. Papenberg und J. Weickert, "High accuracy optical flow estimation based on a theory for warping", in *European Conference on Computer Vision (ECCV)*, Ser. Lecture Notes in Computer Science, Bd. 3024, Springer, 2004, S. 25–36. Adresse: http://lmb.informatik.uni-freiburg.de//Publications/2004/Bro04a.

[15] R. Ranftl, S. Gehrig, T. Pock und H. Bischof, "Pushing the limits of stereo using variational stereo estimation", in *IV*, to appear, 2012.

[16] R. A. Newcombe, S. J. Lovegrove und A. J. Davison, "DTAM: Dense Tracking and Mapping in Real-time", in *Proceedings of the 2011 International Conference on Computer Vision*, Ser. ICCV '11, Washington, DC, USA: IEEE Computer Society, 2011, S. 2320–2327, ISBN: 978-1-4577-1101-5. DOI: 10.1109/ICCV.2011.6126513. Adresse: http://dx.doi.org/10.1109/ICCV.2011.6126513.

[17] J. Stühmer, S. Gumhold und D. Cremers, "Parallel generalized thresholding scheme for live dense geometry from a handheld camera", in *ECCV Workshop on Computer Vision on GPUs (CVGPU)*, Heraklion, Greece, 2010.

[18] J. Stühmer, S. Gumhold und D. Cremers, "Real-time dense geometry from a handheld camera", in *Pattern Recognition (Proc. DAGM)*, Darmstadt, Germany, 2010, S. 11–20.

[19] G. Kuschk und D. Cremers, "Fast and accurate large-scale stereo reconstruction using variational methods", in *ICCV Workshop on Big Data in 3D Computer Vision*, Sydney, Australia, 2013.

[20] G. Kuschk, "Model-free dense stereo reconstruction for creating realistic 3D city models", in *Urban Remote Sensing Event (JURSE), 2013 Joint*, IEEE, Apr. 2013, S. 202–205, ISBN: 978-1-4799-0213-2. DOI: 10.1109/jurse\%23.2013.6550700. Adresse: http://dx.doi.org/10.1109/jurse\%23.2013.6550700.

[21] G. Kuschk, "Large scale urban reconstruction from remote sensing imagery", in *3D-ARCH 2013 - 3D Virtual Reconstruction and Visualization of Complex Architectures*, International Archives of the Photogrammetry, Remote Sensing und Spatial Information Sciences, Bd. XL-5/W1, Trento, Italy, Feb. 2013. DOI: 10.5194/isprsarchives-XL-5-W1-139-2013. Adresse: http://dx.doi.org/10.5194/isprsarchives-XL-5-W1-139-2013.

[22] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Fraundorfer, E. B. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. A. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, S. Weiss und L. Meier, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments", *IEEE Robot. Automat. Mag.*, Bd. 21, Nr. 3, S. 26–40, 2014. DOI: 10.1109/MRA.2014.2322295. Adresse: http://dx.doi.org/10.1109/MRA.2014.2322295.

[23] S. Weiss, M. Achtelik, L. Kneip, D. Scaramuzza und R. Siegwart, "Intuitive 3d maps for mav terrain exploration and obstacle avoidance.", *Journal of Intelligent and Robotic Systems*, Bd. 61, Nr. 1-4, S. 473–493, 2011. Adresse: http://dblp.uni-trier.de/db/journals/jirs/jirs61.html#WeissAKSS11.

[24] R. A. Newcombe und A. J. Davison, "Live dense reconstruction with a single moving camera", in *IEEE Conference on Computer Vision and pattern Recognition*, 2010.

[25] M. Pizzoli, C. Forster und D. Scaramuzza, "REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time", in *2014 IEEE International Conference on Robotics and Automation (ICRA 2014)*, Hong Kong, China, 2014.

[26] G. Vogiatzis und C. Hernández, "Video-based, real-time multi view stereo", 2011.

[27] P. Merrell, A. Akbarzadeh, L. Wang, J.-M. Frahm und R. Y. D. Nistér, "Real-time visibility-based fusion of depth maps", in *In Int. Conf. on Computer Vision and Pattern Recognition*, 2007.

[28] T. Pock, L. Zebedin und H. Bischof, "TGV-Fusion", in *Rainbow of Computer Science*, C. S. Calude, G. Rozenberg und A. Salomaa, Hrsg., Ser. Lecture Notes in Computer Science, Bd. 6570, Springer, 2011, S. 245–258, ISBN: 978-3-642-19390-3. Adresse: http://dblp.uni-trier.de/db/conf/birthday/maurer2011.html#PockZB11.

[29] R. I. Hartley und A. Zisserman, *Multiple View Geometry in Computer Vision*, Second. Cambridge University Press, ISBN: 0521540518, 2004.

[30] A. Rieder, *Keine Probleme mit Inversen Problemen*. Vieweg Verlag, ISBN: 3528031980, 2003.

[31] L. I. Rudin, S. Osher und E. Fatemi, "Nonlinear total variation based noise removal algorithms", in *Proceedings of the Eleventh Annual International Conference of the Center for Nonlinear Studies on Experimental Mathematics : Computational Issues in Nonlinear Science: Computational Issues in Nonlinear Science*, Los Alamos, New Mexico, USA: Elsevier North-Holland, Inc., 1992, S. 259–268. Adresse: http://dl.acm.org/citation.cfm?id=142269.142312.

[32] K. Bredies, K. Kunisch und T. Pock, "Total generalized variation", *SIAM J. Img. Sci.*, Bd. 3, Nr. 3, S. 492–526, Sep. 2010, ISSN: 1936-4954. DOI: 10.1137/090769521. Adresse: http://dx.doi.org/10.1137/090769521.

[33] A. Handa, R. A. Newcombe, A. Angeli und A. J. Davison, "Applications of legendre-fenchel transformation to computer vision problems", Imperial College - Department of Computing, Techn. Ber. DTR11-7, 2011.

[34] F. Steinbruecker, T. Pock und D. Cremers, "Large displacement optical flow computation without warping", in *IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009.

[35] J. Weickert, *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.

[36] H. H. Nagel und W Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences", *IEEE Trans. Pattern Anal. Mach. Intell.*, Bd. 8, Nr. 5, S. 565–593, Mai 1986, ISSN: 0162-8828. DOI: 10.1109/TPAMI.1986.4767833. Adresse: http://dx.doi.org/10.1109/TPAMI.1986.4767833.

[37] P. Perona und J. Malik, "Scale-space and edge detection using anisotropic diffusion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 12, S. 629–639, 1990.

[38] P. Charbonnier, L. Blanc-Feraud, G. Aubert und M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging", in *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, Bd. 2, Nov. 1994, 168–172 vol.2. DOI: 10.1109/ICIP.1994.413553. Adresse: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=413553&tag=1.

[39] L. Alvarez, R. Deriche, J. Sánchez und J. Weickert, "Dense disparity map estimation respecting image discontinuities: a {pde} and scale-space based approach", *Journal of Visual Communication and Image Representation*, Bd. 13, Nr. 1–2, S. 3 –21, 2002, ISSN: 1047-3203. DOI: http://dx.doi.org/10.1006/jvci.2001.0482. Adresse: http://www.sciencedirect.com/science/article/pii/S1047320301904821.

[40] R. G. von Gioi, J. Jakubowicz, J.-M. Morel und G. Randall, "LSD: A Fast Line Segment Detector with a False Detection Control", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 32, Nr. 4, S. 722–732, 2010, ISSN: 0162-8828. DOI: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.300.

[41] S. B. Kang, R. Szeliski und J. Chai, "Handling occlusions in dense multi-view stereo", in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Bd. 1, 2001, I–103–I–110 vol.1. DOI: 10.1109/CVPR.2001.990462.

[42] K.-J. Yoon und I.-S. Kweon, "Locally adaptive support-weight approach for visual correspondence search", in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Bd. 2, 2005, 924–931 vol. 2. DOI: 10.1109/CVPR.2005.218.

[43] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY, USA: Henry Holt und Co., Inc., 1982, ISBN: 0716715678.

[44] E. B. Goldstein, *Wahrnehmungspsychologie: Der Grundkurs*. Heidelberg: Spektrum Akademischer Verlag, 2007.

[45] R. Szeliski und D. Scharstein, "Sampling the disparity space image", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Bd. 26, Nr. 3, S. 419–425, 2004, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2004.1262341.

[46] M. Peris, S. Martull, A. Maki, Y. Ohkawa und K. Fukui, "Towards a simulation driven stereo vision system.", in *ICPR*, IEEE, 2012, S. 1038–1042, ISBN: 978-1-4673-2216-4. Adresse: http://dblp.uni-trier.de/db/conf/icpr/icpr2012.html#PerisMMOF12.

[47] S. Martull, M. Peris und K. Fukui, "Realistic cg stereo image dataset with ground truth disparity maps", *Technical report of IEICE. PRMU*, Bd. 111, Nr. 430, S. 117–118, 2012. Adresse: http://ci.nii.ac.jp/naid/110009482347/en/.

[48] D. Scharstein und R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", *Int. J. Comput. Vision*, Bd. 47, Nr. 1-3, S. 7–42, Apr. 2002, ISSN: 0920-5691. DOI: 10.1023/A:1014573219977. Adresse: http://dx.doi.org/10.1023/A:1014573219977.

[49] D. Scharstein und R. Szeliski, "High-accuracy stereo depth maps using structured light", in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Ser. CVPR'03, Madison, Wisconsin: IEEE Computer Society, 2003, S. 195–202, ISBN: 0-7695-1900-8, 978-0-7695-1900-5. Adresse: http://dl.acm.org/citation.cfm?id=1965841.1965865.

[50] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers und H. Bischof, "Anisotropic huber-l1 optical flow", in *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK, 2009. Adresse: http://gpu4vision.icg.tugraz.at/papers/2009/werlberger_bmvc2009.pdf.

STICHWORTVERZEICHNIS

DANKSAGUNG

In erster Linie möchte ich mich ganz herzlich bei meinen beiden Betreuern, Prof. Dr.-Ing Andrés Bruhn von der Universität Stuttgart und Dr.-Ing. Tobias Schuchert vom Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB) in Karlsruhe, für deren Unterstützung und Begleitung meiner Masterarbeit bedanken. Ich danke Herrn Bruhn für die Zustimmung zur Betreuung und Prüfung meiner Arbeit. Ich möchte mich hiermit aber auch beim Fraunhofer IOSB bedanken, das mir die Möglichkeit zur Durchführung dieser Masterarbeit gegeben hat.

Als Abschluss meines Studiums möchte ich die Gelegenheit nutzen mich auch bei meinen Eltern zu bedanken, die mich in den letzten Jahren in aller Art und Weise unterstützt und mir dieses Studium ermöglicht haben. Insbesondere bedanke ich mich auch bei meiner Frau, Eva Ruf. Besonders in den letzten Wochen meiner Masterarbeit habe ich von ihnen sehr viel Unterstützung erfahren.

Zu guter Letzt bedanke ich mich bei meinen Kollegen, Freunden und meiner Familie für Ihre Unterstützung und Hilfsbereitschaft.

ERKLÄRUNG

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift