Video Visual Analytics

Von der Fakultät Informatik, Elektrotechnik und
Informationstechnik der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

Vorgelegt von

Markus Johannes Höferlin

aus Herrenberg

Hauptberichter: Prof. Dr. Daniel Weiskopf
Mitberichter: Prof. Dr. Gunther Heidemann
Prof. Min Chen, BSc, PhD, FBCS, FEG, FLSW

Tag der mündlichen Prüfung: 27. Mai 2013

Visualisierungsinstitut
der Universität Stuttgart

2013

Acknowledgments

First of all, I thank my advisor Daniel Weiskopf not only for giving me the opportunity to carry out doctoral studies at VISUS, but also for the many and helpful discussions, explanations, and hints. Moreover, I have to thank him for always being patient with me and good-humored. It was really a pleasure and fun to work with him.

I am truly indebted and thankful to my brother, closest collaborator, regular co-author, and friend Benjamin Höferlin. This dissertation would surely not have been possible without him and I cannot imagine having a better colleague than him.

I would like to show my gratitude to Gunther Heidemann for giving me extensive freedom to pursue the video project. I also want to thank him for giving Benjamin Höferlin the opportunity to stay in Stuttgart, which made the fruitful collaboration with him much easier.

It is a great pleasure to thank Min Chen for the excellent collaboration and for the opportunity to visit his former institute at Swansea University, where I spent an outstanding time.

I would like to thank my office mates for countless on- and also numerous off-topic discussions: Thomas Müller, Frank Grave, Andre Burkovski, Florian Heimerl, Kuno Kurzhals, and Michael Stoll. I am obliged to many students who supported me, namely Sara Ahmed Reda Farrag, Jiyang Liu, Rajesh Reddy, Rudolf Netzel, Kuno Kurzhals, Johannes Engelhardt, Jürgen Räuchle, Peter Hoffmann, Manisha Singh, Hendrik Siedelmann, and Dominik Herr. I thank Peter Eltermann, Benjamin Hipp, Michael Wörner, Steffen Koch, and Benjamin Höferlin for proofreading this dissertation. Moreover, I owe sincere and earnest thankfulness to all co-authors of the papers that are the basis for this dissertation.

I also want to mention Michael Wörner for recording many spoken explanations for video supplementary material, Martin Falk for endless LaTeX and Inkscape hints, and Markus Üffinger for various informal discussions. I am also grateful to all other colleagues at VIS and VISUS who made these past years a memorable time.

Last but not least I want to thank my parents, Brigitte Höferlin and Lothar Höferlin, who always encouraged and supported me.

Contents

List of Abbreviations and Acronyms

CCTV	closed-circuit television
CRT	cathode ray tube
DOM	degree of membership
DTW	dynamic time warping
EDA	explorative data analysis
FDEB	force-directed edge bundling
fps	frames per second
GIS	geographical information science
GPS	global positioning system
HMM	hidden Markov model
HOG	histogram of orientated gradients
HRTF	head-related transfer function
Hz	Hertz
ISO	International Organization for Standardization
IR	information retrieval
ISS	interactive schematic summaries
KDD	knowledge discovery in databases
KIS	known item search
KNN	k-nearest neighbors
LBP	local binary patterns
LCD	liquid crystal display
LCSS	longest common subsequence
MMN	mismatch negativity
MSER	maximally stable extremal regions
PCA	principal component analysis
px	pixels
RGB	red, green, blue
SfM	structure from motion
SIFT	scale-invariant feature transform
SLAM	simultaneous localization and mapping
sRGB	standard RGB
SURF	speeded up robust features
SVM	support vector machine
VPG	VideoPerpetuoGram

Abstract

The amount of video data recorded world-wide is tremendously growing and has already reached hardly manageable dimensions. It originates from a wide range of application areas, such as surveillance, sports analysis, scientific video analysis, surgery documentation, and entertainment, and its analysis represents one of the challenges in computer science. The vast amount of video data renders manual analysis by watching the video data impractical. However, automatic evaluation of video material is not reliable enough, especially when it comes to semantic abstraction from the video signal.

In this thesis, the visual analytics methodology is applied to the video domain to combine the complementary strengths of human cognition and machine processing. After depicting the challenges of scalable video analysis, a video visual analytics pipeline is proposed that relies on stream processing for scalability.

The proposed video visual analytics pipeline consists of six stages that are processed successively—*data stream* selection, *manipulation*, *feature extraction*, *filtering*, *relevance measure*, and *visualization*—before the results are presented to the *human analysts*. The *human analysts* can interact and modify each of these stages iteratively. To support sense-making, the *human analysts* can directly integrate and organize reasoning artifacts into a *reasoning sandbox*.

For the video visual analytics pipeline, various methods for the different stages are introduced that address data scalability, task scalability, and situational awareness. This work focuses mainly on the filtering and visualization stages, but provides reviews and discussions of techniques for the other stages as well.

In the *filtering* stage, four interaction guidelines—easy-to-use filter definition, confidence-incorporated filter definition, decision-guided filter definition, and filter feedback—are defined and applied to formulate filters *by properties*, *by sketch*, or *by example*. Due to the suitability of trajectories for filtering, a configurable similarity metric for trajectories is introduced that allows combining different facets (features) with different similarity measures.

Besides a survey on video visualization methods, the thesis contributes to the *visualization* stage by methods for fast-forward video visualization and hierarchical video exploration (the *interactive schematic summaries*). The *VideoPerpetuoGram* is extended and applied to different domains (video surveillance and snooker skill training), and an example of video visualization that solely depends on extracted features from video (the *layered TimeRadarTrees*) is discussed. Moreover, two sonification approaches with the purpose to improve situational awareness are introduced.

German Abstract
—Zusammenfassung—

Der Umfang der weltweit aufgenommenen Videodaten wächst stark und hat bereits Dimensionen erreicht, die schwer handhabbar sind. Die Videodaten stammen dabei aus verschiedensten Anwendungsgebieten, wie beispielsweise der Videoüberwachung, Sportanalyse, wissenschaftlichen Videoanalyse, Operationsdokumentation und der Unterhaltung, und deren Analyse ist eine der Herausforderungen der Informatik. Die gewaltige Menge der Videodaten macht eine manuelle Analyse durch Anschauen des Materials unmöglich. Die automatische Auswertung der Videos ist jedoch nicht ausreichend zuverlässig, vor allem wenn aus dem Video semantisch abstrahiert werden soll.

In dieser Arbeit wird die Visual Analytics Methodik auf das Gebiet der Videoanalyse angewandt, um die Stärken der menschlichen Wahrnehmung mit den Vorteilen der maschinellen Verarbeitung zu kombinieren. Nachdem die Herausforderungen der skalierbaren Videoanalyse betrachtet wurden, wird ein Verarbeitungsschema für die visuelle Analyse von Videodaten vorgestellt, das aufgrund der Skalierbarkeit auf Paradigmen der Datenstromverarbeitung setzt.

Das vorgeschlagene Modell der Videoanalyse durchläuft sechs aufeinanderfolgende Phasen, bevor die Ergebnisse den *Analysten* präsentiert werden: Die Auswahl der *Datenströme*, *Manipulation*, *Merkmalsextraktion*, *Filterung*, *Relevanzberechnung* und *Visualisierung*. Die *Analysten* können hierbei iterativ mit jeder einzelnen Phase interagieren und diese modifizieren. Um die *Analysten* bei Schlussfolgerungen zu unterstützten, ist es möglich, die gewonnenen Erkenntnisse in einen Argumentationsgraph zu integrieren und zu organisieren.

Es werden verschiedene Methoden für die einzelnen Phasen des Verarbeitungsschemas unter Berücksichtigung der Datenskalierbarkeit, der Aufgabenskalierbarkeit und des Situationsbewusstseins vorgestellt. Die Schwerpunkte der Arbeit liegen auf der Filterung und Visualisierung, für die weiteren Phasen werden jedoch vorhandene Verfahren begutachtet und eigene Ansätze zusammengefasst dargestellt.

Für die Phase der Filterung werden vier Interaktionsrichtlinien (Einfachheit, Integration von Konfidenz, Unterstützung von Entscheidungen und Rückmeldung der Filterauswirkung) definiert und auf die Formulierung von Filtern angewandt. Filter können hierbei anhand von Eigenschaften, eines Beispiels oder aber auch einer Skizze erstellt werden. Aufgrund des Potentiales von Trajektorien für die Filterformulierung wird zudem eine konfigurierbare Ähnlichkeitsmetrik für Trajektorien eingeführt, die eine beliebige Kombination unterschiedlicher Merkmale mit verschiedenen Ähnlichkeitsmaßen ermöglicht.

Die Arbeit steuert, neben einem Überblick von in der Literatur vorgeschlagenen Videovisualisierungsmethoden, einige neue Methoden für die Phase der Visualisierung bei. Dies umfasst Visualisierungen für den Videoschnellvorlauf und eine Methode zur hierarchischen Videoexploration (die *Interactive Schematic Summaries*). Das *VideoPerpetuoGram* wird erweitert und auf die Anwendungsgebiete Überwachung und Schulungen für Snookerspieler angepasst. Die *Layered TimeRadarTrees*, eine ausschließlich auf aus dem Video extrahierten Merkmalen basierende Visualisierungstechnik, werden diskutiert. Überdies werden zwei Sonifizierungsansätze vorgestellt, die entwickelt wurden, um das Situationsbewusstsein zu verbessern.

Chapter

1 Introduction[1]

A large portion of the rapidly increasing global data volume is video. Actually, video is the dominant data type in many data domains, for example, video accounts for half of the consumer internet traffic in 2011 [2], and more than 95 percent of clinical data is video [207]. The amount of video data organized in large collections grows at fast pace, for instance, YouTube reports of 72 h of video footage uploaded to their network every minute [280]. Furthermore, there are about 40 million surveillance cameras installed worldwide [223].

For reasonable interpretation of video data, comprehensive and reliable analyses are necessary. Considering the vast amount of video data, analysis methods have to be scalable and efficient. In general, there is a trade-off between quality and efficiency, but in particular, there are a few problems that suit well automated video analysis. Some examples of applications that are qualified for automated video analysis are the detection of shot transitions in movies (e.g., Koprinska and Carrato [169]) or the license plate recognition for traffic surveillance (e.g., Chang et al. [51]).

However, many problems cannot be addressed adequately by purely automated video processing. Imagine the challenge of searching for suspicious events within several hours of video data captured by a surveillance camera. While the task of detecting a specific person or a previously known event can be handled by an automated analysis process, the search for vaguely defined targets, such as the mentioned one, becomes unreliable (see Figure 1.1). Following Leyk et al., vagueness can be "defined as indeterminacy due to a lack of distinctness between ill-defined or fuzzy classes of objects" [189]. A search for such a vaguely defined target (which itself can consist of

[1] Based on Höferlin et al. [132, 137].

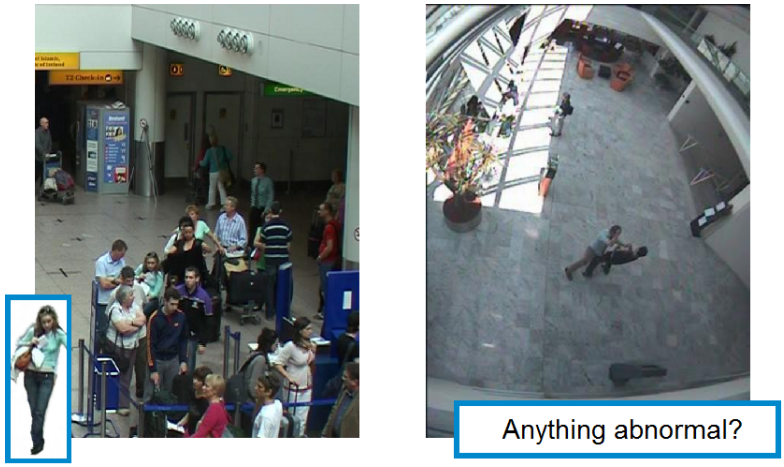


Figure 1.1 — Example of a well-defined search target (specific person in the left image) and a vaguely defined target (search for "abnormal" behavior in the right image). The images originate from the PETS 2004 dataset [99] and the PETS 2006 dataset [281].

several objects) is an ill-posed problem. However, the transition between well-defined and vaguely defined targets is smooth.

In general, one can observe that reliability of automated video analysis depends on the complexity of the problem, which in term can be expressed by i) the degrees of freedom inherent in the problem's definition (see for example Figure 1.1) and ii) the degrees of freedom present in the problem's context or environment (e.g., projection, illumination, noise), where dynamic environments further intensify the complexity and thus reduce reliability [224]. We can also identify that the former (i) also depends on the semantic level of the problem or in other words the amount of abstraction involved, and that this criterion is responsible for the classification if a problem is vaguely or well-defined. Automated video analytics systems that try to cope with complex problems often suffer from high false alarm rates [143]: there is a trade-off between recall and precision.

Due to these problems of automated video analysis, users often fall back to the traditional method: manual video analysis (i.e., watching the complete video), which obviously lacks scalability. This affects different application domains with different video characteristics, such as closed-circuit television (CCTV), sports analysis, surgery surveillance, scientific studies, or the analysis of large video or movie collections. Hence, the situation in video analysis can be summarized by the words of John Naisbitt [220]: "We are drowning in information but starved for knowledge."

To overcome these problems and to take the complexity and data scale of the video domain into consideration, the field of video visual analytics has been established in recent years. Although video visual analytics has several older roots, the cornerstone

to starting this relatively new area of its own can be traced back to IEEE VisWeek 2009. There, a workshop on Video Analytics was held to connect multimedia analysis with visual analytics [58] and the IEEE VAST Challenge 2009 [120] included a video mini-challenge for the first time. The main differences between video visual analytics and other already established application areas of visual analytics are characterized by the combination of the questions and tasks that are related to video analysis with the huge amount of complex and dynamic data.

1.1 Research Questions

In this context, the thesis adresses the following research questions:

R1 Which types of questions are asked in the context of video analysis, and how can the strengths of humans and machines be combined to answer these questions in the presence of the vast amount of video data and the complexity of the analysis problem?

R2 Which process structure and which process stages are appropriate to analyze video data with respect to scalability?

R3 How can the particular stages be designed and which techniques and methods are appropriate for those?

1.2 Outline and Contributions

This section provides the outline of the thesis and the contributions of the authors with respect to the particular reused publications. Please note that all of these publications are co-authored by my adviser Daniel Weiskopf, who contributed constantly with ideas and carefully revised all manuscripts. All publications that are co-authored by Benjamin Höferlin have in common that the parts written by me have been revised by him, and vice versa.

In detail, the material from the following co-authored publications has been partly reused for this thesis: Borgo et al. [31, 32], Bosch et al. [34], Burch et al. [46], and Höferlin et al. [126, 127, 128, 130, 131, 132, 134, 136, 137, 138, 139, 140]. The copyrights of the original publications belong to the corresponding publishers and/or the authors. The authors' contributions to these publications are provided in Table 1.1.

In Chapter 2 of this thesis, a description of the video visual analytics process is formulated based on previous work by Pirolli and Card [236], Keim et al. [159], and the research agenda of visual analytics [282]. Additionally, the typical questions asked

in the context of video analysis as well as the particular challenges to visual analytics that arise from the complex type of video data are reviewed. With regard to these questions and challenges, the resulting video visual analytics pipeline is described in Chapter 2.4. The pipeline consists of several stages that are processed successively, where the *human analysts* can interact with and modify each of them. The stages are *data streams*, *manipulation*, *feature extraction*, *filtering*, *relevance measure*, and *visualization*. Additionally, the *reasoning sandbox* stage is integrated in the video visual analytics pipeline to support the analysts in their analytic discourse. These particular stages are discussed subsequently throughout the thesis, starting with *data streams* and *manipulation*, which are outlined in Chapter 2.5 and 2.6, respectively. Chapter 2 as well as the introductions of the particular stages in the subsequent chapters were published in [132].

Chapter 3 covers the *feature extraction* stage. After surveying common methods for automated video analysis (published in [31, 32]), the extraction of the particular feature trajectory is explained in more detail due to its importance for most of the proposed analysis methods in this thesis (published in [137]). The chapter closes with an application example, where computer vision techniques are applied to extract features that can be used in visualizations for snooker skill training (published in [136]).

Chapter 4 details *filtering*, which is used for data reduction. Besides data aggregation, it is one of the main concepts for achieving data scalability in the video visual analytics pipeline. First, a flexible similarity metric for trajectories is introduced that allows combining arbitrary facets (features) of trajectories with three types of similarity measures (published in [140]). To support users in their filter formulation, four interaction guidelines are developed and their implementation in different concepts for filter definition are discussed: filtering by properties (published in [137]), by sketch, and by example. Finally, a method for ad-hoc learning of classifiers that can be used as filters is briefly introduced (published in [130]).

In Chapter 5, the *relevance measure* stage is introduced that evaluates the importance of particular data elements, such as video frames or trajectories. In contrast to filters, relevance measures do not exclude data from further analysis, but can be used for visualization or adaption of the playback speed. Due to various possibilities of defining relevance, this chapter further introduces three different measures: two measures that rate the frame importance in video (information-based and visual attention model based, published in [127] and [131], respectively), and a relevance measure for trajectory importance based on filter definitions (published in [137]).

The *visualization* stage is addressed in Chapter 6. In the beginning, a comprehensive survey in the field of video visualization is provided (published in [31, 32]). Then, several video visualization techniques are proposed for different objectives: video visualization for fast-forward (published in [127, 139]), the *interactive schematic summaries* (ISS) for

hierarchical video exploration (published in [138, 140]), video visualization for tracked moving objects (published in [137]), and video visualization in context of snooker skill training (published in [136]). Afterward, the *layered TimeRadarTrees*, a technique that can be used to visualize extracted features from video, is introduced (published in [46]). The chapter closes with a brief discussion of two video sonification (the auditory pendant to video visualization) approaches (published in [126, 128]).

The *reasoning sandbox*, which tightly couples the sense-making and the foraging loops, and thus is fundamental to support the human analysts in their analytic discourse and sense-making, is briefly discussed in Chapter 7. Chapter 8 concludes the thesis and provides directions for future research.

In addition, further co-authored work that goes beyond the focus of this thesis can be found in Burch et al. [45, 47], Höferlin et al. [129], Käppeler et al. [154], Reddy et al. [245], and Zweigle et al. [319]; co-authored patents can be found in Class et al. [62, 63] and Höferlin et al. [133, 135].

Table 1.1 — Contributions of the authors with respect to the particular publications.

Publication(s)	Contributions
[132]	Joint work with Benjamin Höferlin, Gunther Heidemann, and Daniel Weiskopf [132]. This work is mainly based on the implementation of reused own publications. Benjamin Höferlin contributed with the idea of the integrative view of visual analytics and the sense-making loop, made improvements to the implementation of the framework architecture, and wrote the first part of the paper (motivation, integrative view of visual analytics, challenges, and related work). I contributed by the implementation and integration of the reasoning sandbox, prepared the supplementary material (case study slides), and wrote the second part of the paper (video visual analytics pipeline, description of the particular stages, and evaluation). The ideas about the video visual analytics pipeline, the framework, and the development of the challenges for video analysis emerged during various discussion between Benjamin Höferlin and me.
[31, 32]	Joint work with Rita Borgo, Min Chen, Edward Grundy, Ben Daubney, Gunther Heidemann, Benjamin Höferlin, Heike Jänicke, Daniel Weiskopf, and Xianghua Xie [31], which was revised and extended for journal publication by the same team [32]. Each author contributed to particular parts, where Rita Borgo took also the organization of the team. I contributed to the video visualization survey by the sections another video or an animation and additional information and actions.
[137]	Joint work with Benjamin Höferlin, Daniel Weiskopf, and Gunther Heidemann. Originates from an IEEE VAST Challenge participation [34] and a workshop paper [134]. The idea was developed together with Benjamin Höferlin during various discussions. Benjamin Höferlin contributed by the implementation of the trajectory extraction and uncertainty propagation and wrote the sections introduction, structure of the proposed framework, and video vision. I implemented the video visualization and the user interaction and wrote the related work, video visualization, and user interaction sections.

Publication(s)	Contributions
[136]	Joint work with Edward Grundy, Rita Borgo, Daniel Weiskopf, Min Chen, Iwan W. Griffiths, and Wayne Griffiths. The idea of this work stems from Min Chen. The team from UK contributed by snooker expertise, conducted the user study, and wrote the paper. I implemented the video visualization, developed, implemented, and wrote the computer vision part, and prepared the images.
[138, 140]	Joint work with Benjamin Höferlin, Gunther Heidemann, and Daniel Weiskopf, where the journal article [140] is an extended version of the conference paper [138]. The ideas were developed together with Benjamin Höferlin during various discussions. Benjamin Höferlin wrote the introduction and the browsing example, and parts of the trajectory processing section. I made the implementation, conducted the user study, prepared the video for supplementary material, wrote the visualization section, parts of the trajectory processing section, and the initial user feedback section.
[130]	Joint work with Benjamin Höferlin, Rudolf Netzel, Daniel Weiskopf, and Gunther Heidemann. The idea for this work stems from Benjamin Höferlin. The implementation was done in context of a diploma thesis by Rudolf Netzel. I contributed mainly by supervising him and by preparation of a video for supplementary material (also together with Rudolf Netzel).
[127]	Joint work with Benjamin Höferlin, Daniel Weiskopf, and Gunther Heidemann. The idea stems from Benjamin Höferlin, who also implemented the relevance extraction, and wrote the introduction and the descriptions of the measurement approach. I implemented the visualization, wrote the related work and visualization sections, and prepared the supplementary material videos. The user study was prepared and conducted together with him.
[131]	Joint work with Benjamin Höferlin, Hermann Pflüger, Gunther Heidemann, and Daniel Weiskopf. The idea stems from Benjamin Höferlin, he also supervised Hermann Pflüger, who implemented the approach in context of his diploma thesis. Benjamin Höferlin also wrote the approach description. I wrote the introduction, related work, and the section with the application to video fast-forward.

Publication(s)	Contributions
[139]	Joint work with Kuno Kurzhals, Benjamin Höferlin, Gunther Heidemann, and Daniel Weiskopf. While the idea stems from me, the implementation and conduction of the user study was done by Kuno Kurzhals in context of his diploma thesis, which was supervised by Benjamin Höferlin and me. Kuno Kurzhals also wrote the evaluation section, and Benjamin Höferlin wrote the introduction. I additionally wrote the description of the visualization approaches, the discussion, and conclusion.
[46]	Joint work with Michael Burch and Daniel Weiskopf. The idea, implementation, and large text portions were contributed by Michael Burch. I prepared the dataset, contributed to the presentation of the paper and refined the manuscript.
[126]	Joint work with Benjamin Höferlin, Michael Raschke, Gunther Heidemann, and Daniel Weiskopf. The idea stems from Benjamin Höferlin, who also wrote the majority of the paper. The implementation and user study design and preparation was made in collaboration between Benjamin Höferlin and me. Michael Raschke executed the user study and wrote the evaluation section. The videos for supplementary material were prepared by me.
[128]	Joint work with Benjamin Höferlin, Boris Goloubets, Gunther Heidemann, and Daniel Weiskopf. The idea stems from Benjamin Höferlin. The implementation and conduction of the user studies were made by Boris Goloubets in context of his student and diploma theses, which were supervised by Benjamin Höferlin. Benjamin Höferlin also wrote the introduction, related work, and the description of the approach. I wrote the evaluation section and contributed to the evaluation of the user study.

Chapter

2 Principles of Video Visual Analytics[1]

In this chapter, the visual analytics process is discussed with respect to prior work in literature, first. Then, the challenges and questions that arise in the analysis of video data (Chapter 2.1) are reviewed. Chapter 2.2 discusses scalability aspects in this context, succeeded by an overview of related research fields. Based on the requirements of video visual analytics, the video visual analytics pipeline is described in Chapter 2.4. Finally, a brief discussion of the first two stages of the video visual analytics pipeline, which are not in the focus of this thesis, is provided.

The general goal of visual analytics as "the science of analytical reasoning facilitated by interactive visual interfaces" [282] is to generate insight from data. According to the mantra of Keim et al. [158], the different stages involved in an iterative visual analytics process can be sketched by "Analyse First – Show the Important – Zoom, Filter and Analyse Further – Details on Demand." The process of making sense of data with respect to investigational tasks was studied by Pirolli and Card [236] and inspired the research agenda of visual analytics [282]. According to Pirolli and Card, the sense-making process can be split into two conceptual loops: a foraging loop and a sense-making loop. The foraging loop includes processes aimed at gathering, searching, and filtering data and extracting relevant information as foundation for further reasoning. These elementary reasoning artifacts[2] are then developed into a mental theory or hypothesis that is best supported by the evidence extracted from the

[1] Based on Höferlin et al. [132].

[2] Following the terminology of the research agenda of visual analytics [282], elementary reasoning artifacts include relevant information, assumptions, and evidence. Besides elementary reasoning artifacs, there are pattern artifacts (e.g., temporal and spatial patterns), higher-order knowledge constructs (e.g., arguments and causality), and complex reasoning artifacts, such as hypotheses.

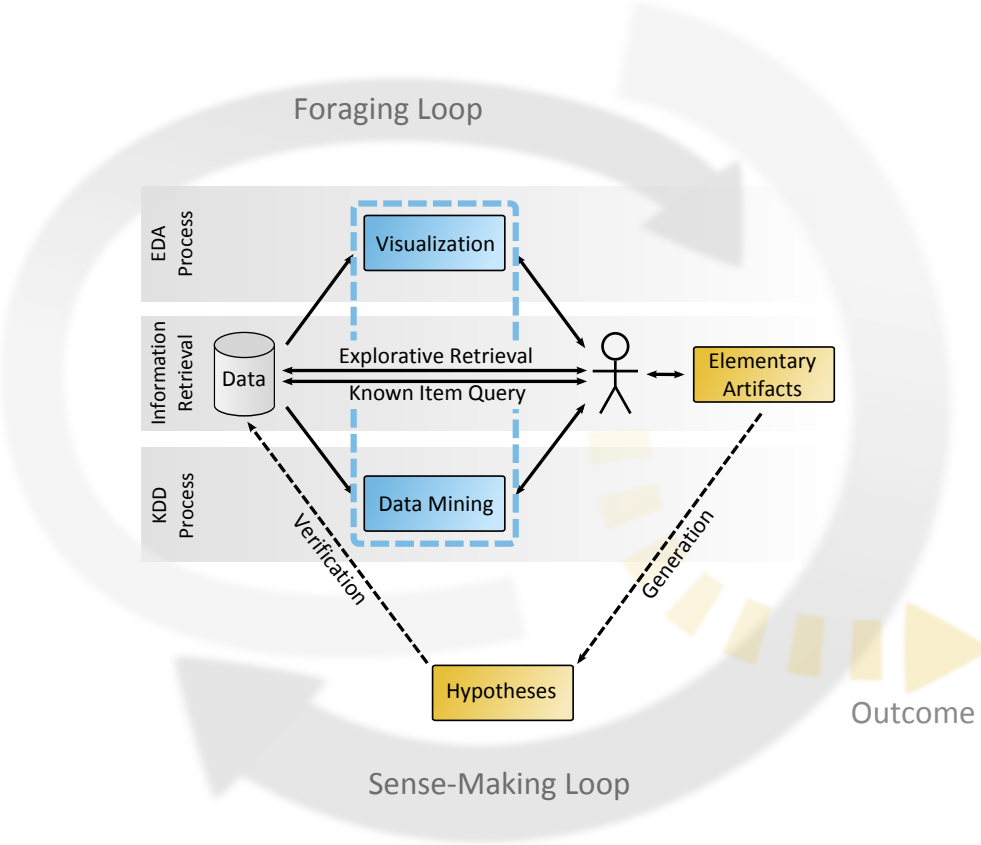


Figure 2.1 — Integrative view of visual analytics and the sense-making process. The tight integration of visualization, data mining, and user feedback (marked by the blue dashed rectangle) leverages different data analysis methodologies for pattern and structure discovery in the foraging loop (components are depicted in blue). Higher-level reasoning products (yellow boxes) are involved in the sense-making loop. Based on elementary reasoning artifacts, hypotheses and more complex scenarios are generated during the discourse of an analysis. After verification or rejection of these hypotheses against the data basis, users may go back to the foraging loop several times in this iterative process, until the final outcome or insight is produced.

data or inferred from an argumentative basis.

A simplified model of the sense-making process integrated into the visual analytics process is depicted in Figure 2.1. This visual analytics process model slightly differs from that of Keim et al. [159] by putting stronger emphasis on the sense-making process. The core of the proposed model represents the sense-making loop in which hypotheses are developed from elementary reasoning products such as evidence. After hypothesis generation, each mental theory has to be checked against the data by

prediction and hypothesis testing and may finally lead to some outcome in form of task-depending insights or knowledge from data. In contrast to the sense-making loop that mainly involves information products of higher levels of abstraction, the foraging loop essentially consists of extracting relevant information from, and discovering knowledge in, raw data. The tasks belonging to the sense-making loop can therefore be summarized with reasoning and deduction, whereas the foraging loop involves separating signal from noise, relevant from irrelevant information as well as extracting patterns and building models.

Tight coupling of the human recognition capabilities with the processing power of computers typically characterizes the sense-making process (foraging loop and sense-making loop) in visual analytics. Bertini and Lalanne term this "integration of automatic and interactive data analysis" and refer to it as the "fingerprint of Visual Analytics" [28]. Considering the foraging loop of visual analytics, the combination of automatic and interactive data analysis is reflected in the close integration of three major knowledge extraction methodologies: *explorative data analysis* (EDA), *knowledge discovery in databases* (KDD), and *information retrieval* (IR).

EDA, a term coined by Tukey [287], is the human-centered data-driven process (bottom-up) of generating models of phenomena of the data (patterns and structure). In contrast to classical mathematical data analysis, EDA is facilitated by visual representations of the data and, thus, allows deriving statistical models by data exploration instead of relying on models pre-imposed by the analysts. However, after data exploration, any discovered model has to be evaluated within a confirmatory step.

As counterpart to the visual representation in EDA, the data-driven KDD process (bottom-up) utilizes automatic data mining by "applying data analysis and discovery algorithms that produce a particular enumeration of patterns (or models) over the data" [93]. These automatically mined patterns are subsequently confirmed or rejected by the analysts. Data mining algorithms allow evaluating a large amount of different models while the conformational step of interpretation by the analyst within the KDD process assures that statistical significance is not undermined.

The models in EDA and KDD are mainly data-driven. In contrast to this bottom-up data analysis by exploration, IR enables the analysts to pose queries to the database to satisfy their information need. The step of query formulation depends on previous knowledge of the analysts or previous iterations of the foraging process. The top-down IR process can either be used in exploratory or in confirmatory ways and help the analysts develop and check higher knowledge constructs (as defined in the research agenda of visual analytics [282]) for subsequent sense-making.

Although Figure 2.1 suggests a clear separation and order of processes, in practical realizations of the visual analytics process transitions are diluted. The mentioned processes are often carried out iteratively and even recursive application of them at

different levels of abstraction is possible. That means besides repeating particular tasks, sub-tasks may also be represented by a visual analytics process or parts of it. This especially applies to the support of (sub-)tasks by visualization and automatic methods, contributing to the high integration of visualization and automatic approaches in visual analytics.

In the context of visual analytics of video data, the authors of the research agenda of visual analytics argue that "new techniques must be developed to integrate the [these] capabilities for analyzing streaming video data into the analyst's toolkit" [282]. To this end, this thesis proposes a processing pipeline for analyzing video streams (Chapter 2.4). Additionally, the thesis introduces multiple novel techniques that address different stages of the pipeline. For the purpose of requirements analysis of such a video visual analytics pipeline, the challenges in video analysis are reviewed in the next section.

2.1 Challenges in Video Analysis

In general, one can identify three main goals targeted by semantic video analysis[3]: *status determination*, *event detection*, and *model generation* (see Figure 2.2). In the case of status determination, the amount of data considered in video analysis remains quite manageable, since the (overt) status of a scene or a recorded entity at a previously specified point in time can be determined using a small fraction of a video sequence (e.g., the current weather or the number of persons in a queue at a particular point in time). However, the detection of events in, or the generation of models from, video data typically involves analysis of a large portion of video footage. While the processing interval to generate a model of the video data coincides with the target interval of which the model has to be built, processing interval and target interval in event detection typically differ (see Figure 2.2). Often, a large portion of video footage has to be analyzed to detect a typically small interval of the video sequence that covers the requested event. The main objective in event detection is therefore to determine the point in time at which a distinct event occurs. In contrast, models are built to describe the common patterns of a single sequence or a set of multiple video sequences (e.g., common movement patterns of a soccer player in sports analysis). Based on such description, predictions on future data can be made or the model can be used to define an event detection task: for example, the search for abnormal behavior of subjects in video surveillance as an outlier according to a common pattern (model).

[3] These goals are derived from a literature survey of applications in the video domain, such as video retrieval and video surveillance, e.g., Goodwin and Goodwin [114], Lomell [196], Francois et al. [102], Gill et al. [110], Svensson et al. [273], and Keval [162].

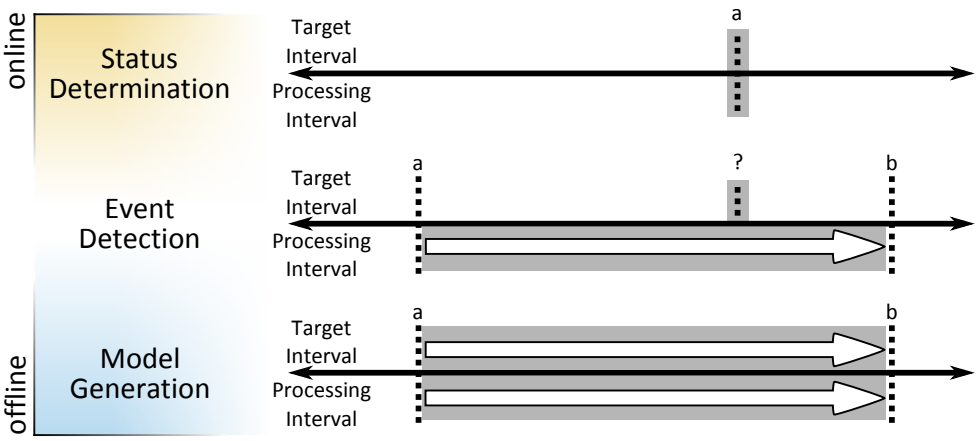


Figure 2.2 — Types of questions asked in video analysis with their according processing and target intervals. The diverging color indicates the diffuse transition between online and offline processing for the particular tasks. Time is represented on the right side by the x-axis, and known points in time are marked by 'a' and 'b'. For status determination, only a small portion of data has to be taken into account. When searching for an event, the point in time is usually unknown (denoted by '?') and, hence, the processing interval often ranges over a large time-span. For model generation, the time-span of analysis typically matches the processed interval.

Furthermore, the problem of video analysis can be categorized into offline analysis that considers only historic video data and online analysis of real-time video streams. In practical applications, both types of analyses exhibit different foci on the kinds of questions that they can answer. The determination of status is more common in real-time applications, whereas model generation is rather performed on historic video data. However, a sharp separation does not exist, as depicted in Figure 2.2.

As mentioned briefly in Chapter 1, the dominant challenges in video analysis originate from the large amount of video data to be analyzed, the quality of search target definition, and the complexity of the video domain. Well-defined search targets (e.g., the definition of a critical event in monitoring situations) require a proper model and information about the acceptable or necessary deviation from this model. The model can either be built from data or be compiled from previous knowledge of the analyst. Although precise search target definition is important for all three video analysis tasks, the impact of vaguely defined search targets on event detection is most severe. In event detection, vaguely defined search targets hinder the application of *known item search* (KIS) by automatic event detection, which would generate huge reduction of analysis costs due to the large ratio between target and processing interval. However, vaguely defined search targets demand for more exploratory data analysis that requires human analysts and their background knowledge. Additionally, the complexity of the

video domain prevents successful application of automatic video analytics approaches. The complexity of the video domain is characterized by the mainly unstructured and dynamic type of data, the numerous degrees of freedom (e.g., uncontrolled environment with regard to imaging parameters or illumination conditions [224]), many ambiguities, and often low signal-to-noise ratio.

The unreliability of automatic video analytics approaches is a severe problem, especially in the context of security applications (e.g., video surveillance). In such context, high recall is mandatory. However, high precision is similarly important, because high false alarm rates annoy and desensitize the users. Unfortunately, the low precision achieved by recent video analytics approaches is the reason that this area is questioned in general [143]. Hence, human analysts are required to analyze the video data manually or in a semi-automatic fashion by utilizing low-level (close to the signal) computer vision approaches that are rather reliable. As consequence to the main challenges of video analysis—vast amount of data, complex data, and vaguely defined search targets—purely automatic or purely manual analysis is not applicable. Visual analytics, however, provides a way to combine the strengths of both worlds (see Keim et al. [160]) to form a scalable way of both targeted and exploratory video analysis.

2.2 Scalability

As data scales up, the information gap (i.e., the gap between the amount of available data and the demanded information[4]) increases. This requires techniques to facilitate knowledge extraction scalable with data increase. The authors of the research agenda of visual analytics identify scalability as the major challenge in the context of visual analytics [282]. They distinguish five types of scalability that in turn all arise from the problem of exploding data volume: information scalability, visual scalability, display scalability, human scalability, and software scalability. In general, three methods can be applied to achieve scalability in the presence of huge amount of data: *serialization*, *parallelization*, and *data reduction*. In the context of visual analytics, these three methods have to be considered from the perspective of both human and machine:

- Serialization is often used to process datasets that are too large to be kept in working memory of human or machine. Hence, data can be processed in data streams. Since video data is time-oriented, a natural streaming dimension already exists that can be used to process the data sequences.
- To increase the processing capacity compared to a single working instance (e.g., a computer or a human analyst), work distribution often is the method of choice.

[4] following the terminology of Wurman [312]: information that does not *inform* anymore is just considered as data.

This leads to parallelization of work on the one hand by collaboration of analysts or by crowdsourcing anonymous labor, and on the other hand by distributed computing, such as cluster computing or cloud computing.

- To enable data scalability of a single working instance, the problem size (most commonly the amount of data) has to be reduced. Data reduction is a natural approach often applied in the form of task-dependent filters (e.g., see filtering of the sensory input to the consciousness in the human brain [296]). Another possibility is the use of aggregation methods, such as hierarchical problem solving or divide-and-conquer approaches that start with a coarsely aggregated view of the data and end up with the details of the data if required. Note that reduction methods that aggregate data—in contrast to filters or queries—can conflict with sequential processing of the data, if aggregation and serialization dimensions overlap (i.e., only the already sequentially processed data can be aggregated).

These three methods and their application to visual analytics finally lead to the different notions of scalability enumerated by the authors of the research agenda of visual analytics [282].

Involving the human in the sense-making process introduces an additional risk that has to be considered in the context of data scalability. Due to humans' perceptual limits, the concept of *situational awareness* becomes important. According to Endsley, situational awareness "can be thought of as an internalized mental model of the current state of the operator's environment" [88] that covers three levels of understanding in the context of dynamic systems, such as video analysis: perception, comprehension, and prediction of entities and their status [87]. An erroneous or incomplete model, that is a deficit in situational awareness, can cause severe consequences. Further, the pressure of time that is involved in many real-time analysis tasks, such as online, pro-active video surveillance, aggravates the problem even more. Major common perceptual deficits relevant in video analysis are:

- difficulties to identify unexpected changes during blinks, flickers, or disruptions called *change blindness* [249],
- *inattentional blindness* reflected by poor recognition of changes that are outside the focus of attention [203], and
- the short period of attention (about 20 minutes [118]) when monitoring video screens.

Analysts face distractions and additional responsibilities within their work environment that have to be reflected in a holistic view on their situational awareness. Besides taking the limits of humans' perceptual capabilities into account, missing data and uncertainty of the data have to be conveyed to the users in order to maintain proper

situational awareness for decision making. Hence, a scalable visual analytics system has to support the users in keeping track of the sequentially processed data and its uncertainties originating from measurement process (i.e., data acquisition) and data transformations. In collaboration environments, situational awareness further involves an overview of the analysis state of the team members including their data interpretations, derived reasoning artifacts, and the implied confidences. In situations where data reduction techniques have been applied, the confidence of filters applied has to be communicated to the human analysts as well as the patterns or structure lost by aggregation.

The three methods to enable scalable analysis (serialization, parallelization, and data reduction) and the notion of situational awareness inspired the video visual analytics pipeline (see Chapter 2.4) and its corresponding prototypical implementation discussed throughout this thesis.

2.3 Related Research Fields

Video visual analytics is a quite new research area that is part of the field of multimedia analytics [58]. However, due to its integrative character it is connected with many research areas. Among them are automatic video analysis and computer vision, data mining and information retrieval, data stream management systems and moving object databases, as well as visualization and human-computer interaction. This section focuses the discussion on the fields most related to video visual analytics. Related work concerning particular stages of the video visual analytics pipeline, such as feature extraction or video visualization, is discussed in the corresponding chapters.

Video Visual Analytics. There have been several projects on retrieval, exploration, and analysis of large archives of video data. Due to their integration of human and machine analysis, some of these approaches can be considered as video visual analytics.

The *Informedia* project [60, 125] provides exploratory query mechanisms for searching in video libraries, such as broadcast news databases. It aims to retrieve relevant videos out of huge video databases by defining filters. The filters make use of text annotations and image classification algorithms to refine the set of relevant videos. This allows search for shots conveying different characteristics, such as “outdoor shot”, “includes buildings”, or “includes faces”. The search is also supported by the annotations of time stamps, locations on a map view, and keywords in speech.

Ferguson et al. [98] integrate content-based analysis techniques into *interactive TV* devices to explore videos and video archives. For this purpose, they use keyframes that represent automatically segmented shots or scenes in the video, and a video review (the playback of these keyframes) for each video in the archive. Their system is capable

of searching for video parts based on text (e.g., originating from subtitles), low-level visual features (e.g., color, texture, shapes), and faces. They propose processing specific video content, such as sports and news, differently to take advantage of their particular characteristics. Further projects in this category are *Mediamill* [310, 266] and the works of Luo et al. [201, 202].

Other video analysis systems guide users' attention by alarms to certain events in order to increase their efficiency and effectiveness. These events have to be modeled beforehand. Examples are approaches for left-luggage detection [17], fire detection [193], loitering pedestrians detection [147], and the detection of forbidden area violations [188].

Forensic video processing systems provide another type of video analysis. Such an environment is proposed by Jerian et al. [150]. They apply several processing functions to video footage to enhance the video quality, and hence enable the investigation of important regions, such as the license plate of a car. The processing functions include shifted lines correction, compression artifacts reduction, projective registration, and motion deblurring.

In contrast to these approaches, the video visual analytics pipeline proposed in this thesis puts more emphasis on the analytical discourse than on pure retrieval of particular parts of the video stream.

Besides the approaches to video visual analytics, there are two areas of visual analytics that cope with data domains related to video data: geospatial visual analytics and visual analytics of dynamic data. Moreover, moving object analysis comes into focus due to the suitability of trajectories in many video analysis applications.

Geospatial Visual Analytics. Although video visual analytics and geospatial visual analytics both are concerned with spatio-temporal data dimensions, the questions and challenges in these areas differ in many respects. A few approaches from geospatial visual analytics may carry over to video analysis (e.g., the VideoPerpetuoGram (VPG) described in Chapter 6 can be seen as an adoption of the space-time cube, the most prominent element in Hägerstrand's time geography [171] that is applied in geospatial visual analytics [10]), but others have to be adapted or newly designed. The questions in video visual analytics are even more focused on time than on space, although the general components of the questions are the same: "what", "where", and "when" [11]. Often, the "where" question plays a secondary role in video analysis, since location is largely predetermined by the recorded field of view. In addition, the data characteristics differ. Video data offers typically much higher resolutions than geo-referenced data, both temporally and spatially[5]. Additionally, raw video data provides much redundant

[5] Videos are typically captured at frame rates between 24 and 60 fps to allow for continuous motion experience by the human visual system. The perception of such apparent motion in film is based

information by dense optical sampling. Hence, it requires more storage and processing capacities and additional calculations for feature extraction.

Geospatial visual analytics, in contrast, has typically to cope with more objects and longer observation periods, which entails their own open challenges [9]. Generally, video analysis adopts the Eulerian view on the data: data is sampled on a static grid — the pixels. However, the Lagrangian particle-based view dominates geospatial visual analytics. The Lagrangian view reflects the flow transport of a single particle in a vector field. In video analysis, an additional tracking step is required to move from the Eulerian to a Lagrangian perspective (see Kuhn et al. [174]). However, missing information in video data challenges the feature extraction step. Due to the projection from a 3D scene to a 2D image and the discretization and quantization of the continuous signal, much information is lost. This problem has large impact on tracking due to the presence of occlusion and a limited field of view. As a result, many dedicated research areas have emerged that try to solve the correspondence problem for tracking under these conditions (e.g., multi-target multi-camera tracking, person re-identification, etc.). In the context of geospatial visual analytics, these issues do usually not exist, since the data already contain correspondences and metadata necessary for analysis (e.g., GPS data). Nevertheless, both fields have the analysis of spatio-temporal trajectories of moving objects in common.

Moving Object Analysis. Trajectory features of tracked moving objects are suitable for many video analysis tasks. Thus, the presented work overlaps with other research domains that utilize trajectories. Besides differences in terminology and emphasis on different issues, the challenges are quite similar: acquisition and description of trajectories as well as their query, classification, or summarization. The latter often utilize an iterative process that involves context-depending trajectory visualization.

In the field of geographical information science, Mark and Egenhofer [208] termed "the continuous set of positions occupied by an object in geographic space over some time

on two phenomena [294]: stroboscopic motion and visual persistence. Stroboscopic motion appears at frame rates between 5 to 10 fps. To reduce the flicker effect that is to increase visual persistence, higher frame rates are required. Hence, "at a rate of at least 16 frames per second, the 'motion' in a film seems smooth and natural" [243]. Exceptions to these frame rates can be found in time-lapse recording of video footage (frame rates for recording the video sequence is smaller than typical video frame rates) and slow motion captures (frame rate for recording the video is much higher than typical video frame rates). In the *geographical information science* (GIS) domain various primary data sources are used. Remote sensing imagery for example offers typically frame rates between one image a day and one image a year [73] and popular *global positioning system* (GPS) datasets, such as the Microsoft's *GeoLife GPS Trajectories* dataset provide no sampling intervals faster than a second. More information on temporal and spatial resolution of data used in the GIS domain can be found in the textbook of Longley et al. [197]. For video analysis, the resolution largely depends on the field of application. For video surveillance, for example, typical and useful resolutions are discussed in the handbook of Kruegle [173] and studies of Keval and Sasse [164, 163].

period" a *geospatial lifeline*, whose "data consist of discrete space-time observations", "describing an individual's location in geographic space at regular or irregular temporal intervals". Geospatial lifelines often originate from GPS-tracking data and are analyzed in an interactive exploratory fashion involving their visualization (e.g., Dykes and Mountain [83], Andrienko et al. [8]), which bridges the gap to visual analytics. Important progress was also achieved for the description of geospatial lifelines and their analysis with respect to the temporal context (e.g., Hornsby and Egenhofer [145], Laube et al. [182]).

Research in spatial databases was originally motivated by geographical information systems, which themselves served as foundation (together with temporal databases) for spatio-temporal databases. *Moving object databases* are one particular line of research that "deal with moving objects whose position is recorded at, not always regular, moments in time" [175]. The database community has especially contributed to the representation of spatio-temporal objects (e.g., trajectories), their comparison, indexing, and query. Additionally, uncertainty inherent to the measurement of trajectory data as well as query-imprecision support was in focus of research on moving object databases. Further introduction is given by the textbook of Güting and Schneider [121]. An overview of operators to query for trajectories is presented by Wolfson [309]

Visual Analytics of Dynamic Data. Because of its volume, video data is generally processed as stream and can therefore be considered dynamic data, regardless of whether it is recorded historic data (offline) or real-time data from a live capture device (online). Hence, many aspects of dynamic visual analytics [206] also apply to video visual analytics. Despite this overlap, there are fundamental differences between video visual analytics and dynamic visual analytics as it is defined by Mansmann et al. [206]. They especially consider dynamic visual analytics in the context of real-time tasks, such as monitoring of network traffic. However, video visual analytics inherently applies to dynamic data, regardless of their origin. In the same way, their notion of situational awareness falls short for video visual analytics. In contrast to their definition of situational awareness as the target of dynamic visual analysis, situational awareness in the context of video visual analytics is a mandatory element for a successful analysis in each stage of the sense-making process (foraging and sense-making loops), which leads to a consideration from a broader perspective.

To summarize, the presented video visual analytics approach differs from previously published work by its tight integration of the analytical reason process enabling the users to extract knowledge from video data by a unique combination of KIS and video browsing. Besides detailed analysis of the tasks and scalability challenges of the video domain, the contribution lies in the novel architecture of scalable video visual analytics and its prototypical implementation that addresses the technical, perceptual, and cognitive challenges of visual analytics in the video domain.

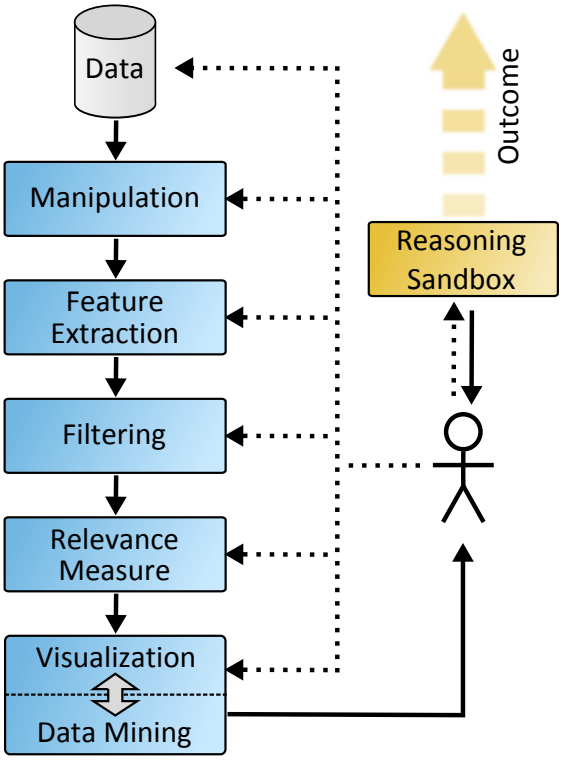


Figure 2.3 — The video visual analytics pipeline. The video data streams are processed successively by several stages and finally presented to the human analysts (solid arrows). Feedback of the analysts (dashed arrows) closes the processing loop and allows them to adjust data transformation at any of the processing stages.

2.4 Video Visual Analytics Pipeline

As discussed above, visual analytics of video data has to cope with large amounts of complex video data and often vaguely defined task descriptions. The video visual analytics pipeline presented in this section considers this and the implementation of the prototype integrates most aspects of the three methods to achieve data scalability for humans and computers (described in Chapter 2.2).

Together with the issue of data scalability, the system helps analysts in maintaining situational awareness, both in the foraging loop and during sense-making. Beyond data scalability, the visual analytics system can be used for all the video analysis tasks introduced in Chapter 2. This type of scalability, termed *task scalability*, likewise covers online and offline tasks.

The video visual analytics pipeline in Figure 2.3 results from the mentioned requirements. A first prototype of this system was developed for a VAST Challenge 2009 participation. It received the award "Outstanding Video Analysis Tool" [34]. During the last years, it has been largely extended by several visualization and interaction techniques as well as methods of automatic video processing. It now provides a rich visual analytics system with wide support of reasoning in video data. Before each component of the video visual analytics pipeline is discussed in detail in subsequent

chapters, this section provides a general overview.

Video *data* is streamed through a couple of stages before it is presented to the *human analyst*. First of all, a video *manipulator* with the objective to enhance video quality (e.g., by color correction or noise reduction) is applied. Then, additional features, such as foreground masks, trajectories, and diverse properties of the features, are extracted (*feature extraction*). Here, subsequent stages determine the kind of features calculated (e.g., trajectory-based filters, particular properties that have to be visualized and are used for data mining). In the *filtering* stage, which is important for the scalability of the system, filters reject the data that has been defined irrelevant to the analyst. The importance of the filtered data is then judged by *relevance measures*. Subsequent methods may access the relevance rating to transform the representation of the data. Finally, the data is presented to the user by the *visualization* stage that is highly integrated with *data mining* techniques.

To this end, the unidirectional flow of video data reflects the fundamental idea of stream processing realized in the system (solid arrows). Nevertheless, the *human analysts* can interact with each of the particular stages; this is outlined by the dotted arrows in Figure 2.3. After user interaction, the stream processing structure necessitates the data to pass in addition to the modified stage each subsequent stage again.

In the early stages, analysts select the video data that has to be analyzed and apply video manipulators. Then, they decide which parts of the data are relevant for their analysis, by designing filters, relevance measures, and larger filter pipelines. Combining various coordinated views allows analysts to observe the aspects of the data important to them. Moreover, they interact with the visualizations and apply different kinds of aggregation and mining methods to discover interesting patterns.

These six stages of the video visual analytics pipeline mainly correspond to the foraging loop in Figure 2.1. The IR process is represented by the sub-loop between *filtering* and the *human analyst*. The KDD process can be found within the sub-loop *data mining/visualization—human analyst* with focus on *data mining*, where the EDA process is integrated into the same sub-loop with focus on *visualization*. The support of the sense-making loop in Figure 2.1 can be found in the video visual analytics pipeline of Figure 2.3 in the form of the connection between the *reasoning sandbox* and the *human analyst*.

In the *reasoning sandbox*, elementary reasoning artifacts, such as relevant information, assumptions, patterns, higher-order knowledge constructs, and evidence can be organized to support the analysts in formulating sound hypotheses. After hypotheses generation, the *human analyst* can check them against the data utilizing again the foraging pipeline in Figure 2.3.

The following two sections outline the *data* and the *manipulator* stages, where the succeeding chapters discuss the remaining stages of the video visual analytics pipeline

in detail.

2.5 Data Streams

In the first stage of the video visual analytics pipeline (see Figure 2.3), the analysts can select one or multiple data streams to be considered in the analysis. This step corresponds to the first step of *search and filter* of the foraging loop of Pirolli and Card [236]. The data sources are not restricted to video data alone. In particular, additional non-video data sources can be any other time-dependent data streams that provide useful enrichment in the analysis. Examples of such data streams are ATM transactions or badge data (i.e., entrance and exit times of persons for particular buildings as provided by the IEEE VAST Challenge 2009). Please note that data streams, although dynamic, can be pre-recorded offline data or real-time online data streams, such as from live cameras. Furthermore, the sampling rates of the added data streams can largely vary (e.g., VAST Challenge 2009 video data: ~15 frames per second (fps); CCTV time-lapse video ~0.25–8 fps [164]) and can be either regularly (e.g., video streams) or irregularly sampled (e.g., badge data).

To achieve data scalability, the visual analytics system generally streams all data sources. However, as soon as temporal aggregation techniques or aggregation of multiple data streams come into play, online algorithms or time windows are required. These windows can either be sliding windows or user-defined static time windows. Further, parts of the streamed data may be sometimes cached by the particular stages (e.g., the *feature extraction* or the *visualization* stage) to resolve temporal dependencies in processing.

2.6 Manipulation

The algorithms in this stage are characterized by not changing the data type between input and output data. Therefore, this stage can be deemed a processing stage rather than a stage of extraction or understanding. The objective of the *manipulation* stage is to enhance the raw data signals to improve the quality and effectiveness of successive stages (especially *feature extraction*, *relevance measure*, and *visualization*). With respect to video, this stage may contain approaches to improve contrast (e.g., in foggy situations), correct colors (e.g., to counterbalance illumination changes during the day), deshake (e.g., important for pillar mounted cameras), deblur, or split scenes in video sequences (e.g., for pan-tilt-zoom cameras, such as in the IEEE VAST Challenge 2009 dataset). The choice of the manipulators applied to the data streams is left to the *human analysts*.

Chapter

3 Feature Extraction[1]

In the *feature extraction* stage, a variety of features are calculated for usage in later stages. In general, the *feature extraction* stage consists of many specific feature extractors that depend either on the original data stream (i.e., the manipulated video stream), or on a composition of previously calculated features. Some feature extractors further build and adapt their own (online) models, or cache the data stream and required features using the sliding window technique. For example, calculated features, such as optical flow or foreground segmentation, only require the video stream and their internal models. The internal model for foreground segmentation is represented by a background model trained on the video data seen so far. Blob[2] extraction in contrast, depends on the optical flow (motion blobs), and the foreground segmentation (foreground blobs) as well as on the video stream itself. Finally, trajectories of moving objects are extracted by applying a tracking algorithm to the previously extracted blobs. This also involves an internal model of the movement characteristics as well as the sliding window technique. The method applied for trajectory extraction in the video visual analytics system is described in more detail in Chapter 3.2. Another example of feature extraction in the context of snooker skill training is provided in Chapter 3.3.

The structure of the *feature extraction* stage is mostly created automatically. A graph of feature extractors is constructed that is based on the features required for *filters*, *relevance measures*, and *visualizations* selected by the users as well as on these features' own dependencies.

---

[1] Based on Höferlin et al. [132], Borgo et al. [31, 32] (Chapter 3.1), Höferlin et al. [137] (Chapter 3.2), and Höferlin et al. [136] (Chapter 3.3).

[2] continuous image region

The *feature extraction* stage can be seen from different perspectives. The important role of trajectories in the presented video visual analytics system originates from the assumption that changing parts in a video sequence are more relevant than static parts. Hence, feature extraction represents, on the one hand, a first data reduction step triggered by a model of background knowledge about the relevant parts in video data. To this end, the raw video stream is aggregated into a more abstract type of information. In fact, some visualizations only present features without the original video data, such as the *interactive schematic summaries* (ISS) (see Chapter 6.3), which uses only trajectory and background image features. On the other hand, feature extraction can serve as an enrichment of the raw data by additionally created features. This is the case for the extraction of trajectories of moving objects from the data. Inspired by the flow visualization community, the transition from video data to trajectories also marks the transition between the Eulerian perspective on the data to the corresponding Lagrangian view. However, both types of descriptions have their raison d'être and complement each other.

In the next section, the state-of-the-art methods used in the field of computer vision to extract information from image sequences are presented.

3.1 Related Work on Video Analysis

Before the different techniques are surveyed in detail, the structure of video and the used terminology of the video segments are discussed. A video of a certain length is depicted in Figure 3.1. Each video consists of a sequence of images, or *frames*. When one or more frames, depicting a continuous action in time and space, are combined in a contiguous recording, this is called a *shot* [226]. The assembly of subsequent shots of a semantic unit is called a *scene*. Both, shots and scenes, can be of arbitrary length and the single units usually differ in length, i.e., there are scenes in a video that only take a split second while others might take several minutes.

In the following, the computer vision techniques that are interesting in context of the video visual analytics pipeline are reviewed. These methods are broadly split into two subgroups: *low-level* and *high-level vision*. Low-level vision techniques often operate at the pixel level of an image and are generally employed to reduce the dimensionality or complexity of an image so that it can be processed by higher-level, often more complex, algorithms. Low-level vision can be interpreted as a filtering step used to remove redundant information that is often of little or no interest. The typical output of these algorithms may be a set of interest features, optical flow vectors, or an image segmentation. However, this information alone often provides little useful insight with regards to the contents of an image sequence.

Alternatively, high-level algorithms that almost exclusively operate on the output

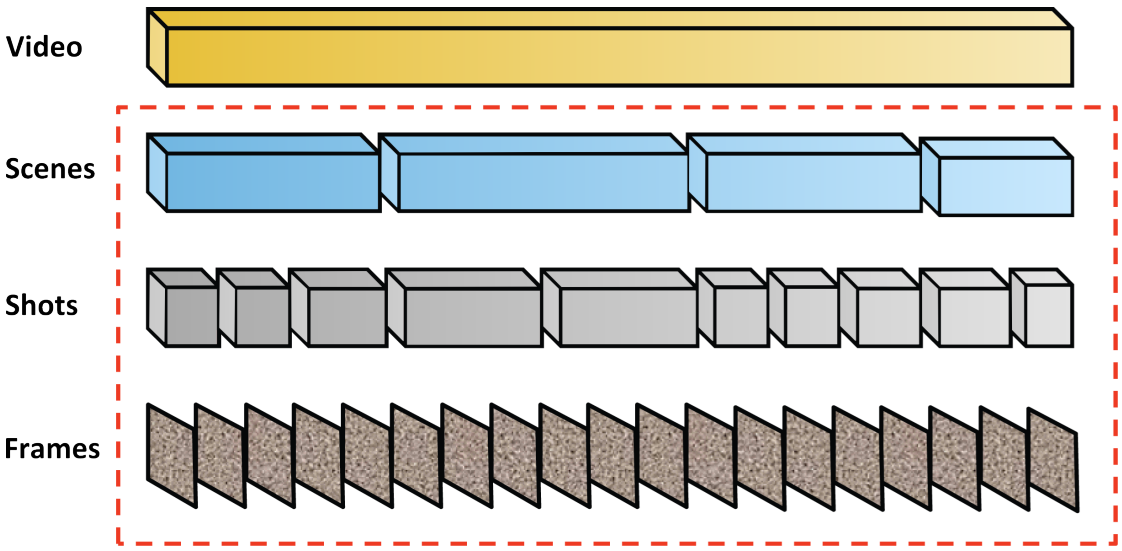


Figure 3.1 — Segments of a video.

of low-level vision approaches can be used to automatically extract some high-level information from a video sequence, such as a list of events that have taken place, a set of locations where objects have been detected, or a 3D reconstruction of the scene depicted in the sequence. The primary goal of most computer vision practitioners is this high-level extraction of data. However, one of the principal difficulties that are encountered is to overcome errors produced by low-level algorithms. As a result, approximately equal effort is spent currently by the vision community on improving low-level methods as are invested in developing high-level approaches.

3.1.1 Low-Level

This section describes the *low-level* vision techniques that are particularly relevant to the domain of video analysis. These methods can be grouped into three principal areas: *optical flow estimation*, *image segmentation*, and *feature extraction*. Whilst optical flow estimation and image segmentation provide a well-defined output that can also be directly used in the *visualization* stage, feature extraction will often produce a more abstract output that is only of benefit to higher-level algorithms designed to exploit it.

Optical Flow Estimation

Motion estimation is one of the most fundamental techniques relevant to video analysis since it exploits the key element that distinguishes video from single images: the temporal dimension. Whilst the focus of this section will be on commonly used

differential methods, block matching can also be used to extract motion information and should briefly be mentioned. In its simplest formulation, block matching takes each image patch and exhaustively compares it against its neighboring frames to find the best matching location. This approach is typically used for video compression and is therefore not concerned with the correctness of the estimated motion, only that matched blocks closely resemble one another. Various methods have been proposed to perform block matching efficiently such as the diamond search adopted for the reference implementation of MPEG4 [318]. A comprehensive survey of block matching techniques is provided by Huang et al. [148].

The most popular methods for motion estimation between two consecutive frames are differential methods. These approximate optical flow using a first-order Taylor expansion of image motion and as such assume only small displacements between consecutive frames. However, they are capable of achieving sub-pixel accuracy. Differential methods to estimate optical flow can be split into *local* and *global* methods. Whilst local methods attempt to solve the motion for small regions of the image independently, global methods attempt to solve motion for the entire image in one instance.

A popular local method is proposed by Lucas and Kanade [199]: they apply an iterative approach that uses Newton-Raphson gradient descent to minimize the dissimilarity between patches in consecutive images. The shortcoming of this approach is that it fails to address the aperture problem. This is where locally the motion between two frames is ambiguous and cannot be uniquely identified. This results in some regions for which the motion is unknown or poorly estimated, for example large motions can often be incorrectly observed along the edges of objects.

Global methods solve the same first-order Taylor expansion of image motion, but introduce a regularization term or smoothness penalty. The addition of the smoothness penalty allows the estimation of optical flow in regions where local methods would fail because of the aperture problem. This enables one to estimate dense flow. However, this method is particularly sensitive to image noise [23, 40]. The most notable global method is that of Horn and Schunck [144].

Whilst the local method of Lucas and Kanade fails to solve the aperture problem, their formulation provides a method to test how well a particular image patch could be tracked. This is achieved by examining the eigenvalues of the covariance of the image gradients [261]. Two large eigenvalues imply large gradients (i.e., edges) in adjacent directions of the patch, which represent a good feature to track. Using this method, each motion vector can have a level of certainty attached to it about how reliable the feature used can be tracked. This is often invaluable for higher-level algorithms since noisy data can automatically be discarded. Some methods have been suggested to 'densify' the sparse output of the Lucas-Kanade method using interpolation [122], which provides better dense motion estimation compared with global methods in

sequences where there is little texture. Another approach is that of Bruhn et al. [40]: they combine local and global methods to extract optical flow by using local confidence measures that effectively grow to a dense representation.

Other local methods use local spectral phase differences to estimate motion displacements between images [100]. Stein proposes a real-time approach using the census transform to represent a pixel's neighborhood [268]. An evaluation of optical flow methods is provided by Barron et al. [23] and Galvin et al. [105]. For a comprehensive survey of global optical flow methods, the reader is referred to Weickert et al. [299].

Image Segmentation

Image segmentation is a generic term for grouping pixels in an image or video into a number of predefined classes, such as those that belong to a particular object or those that are part of the foreground. Pixels are classified using image cues such as color or texture [259], and often, the spatial locations of pixels are exploited to prefer neighboring pixels to be members of the same class. These include methods such as *split and merge*, *region growing*, and edge-based techniques (comprehensive surveys are provided by Cheng et al. [56] and Lucchese and Mitra [200]). These approaches often result in a segmented image that is represented as a set of blobs, where each blob corresponds to a different homogeneous region. However, a blob may not necessarily have a semantic meaning.

In general, image segmentation is not a well-defined problem since a good segmentation is itself somewhat subjective and dependent on what the user requires. For this reason, methods must often be trained for the task for which they are required (e.g., skin detection [152]). Perhaps one of the most popular segmentation methods in video is background subtraction [233, 210], or more generally change detection [240]. Here, the segmentation algorithm is trained on a particular scene to detect (segment) any pixels or regions that change temporally. An evaluation of current background subtraction techniques applied to the most challenging conditions, such as during gradual illumination changes or whilst observing a scene containing a dynamic background, is provided by Brutzer et al. [41].

Further methods for image segmentation include *dynamic programming* [95, 68], *graph cuts* [37], and *level sets* [69]. These approaches allow segmentation to be formulated as an energy minimization problem and have the advantage that they allow the inclusion of complex shape priors specific to the task for which they are required, for example, segmenting cows [177], leaves [95] or hands [68]. These methods are particularly robust to noise and background clutter and it is the inclusion of the aforementioned shape priors that leads to this robustness.

The drawback of these energy minimization approaches is that they cannot be used 'out of the box' and must be trained to the specific task for which they are required.

Contrary, methods that segment homogeneous regions can be treated as a black box. For images that contain little clutter, these methods can achieve acceptable results.

Feature Extraction

This section describes low-level features commonly used in computer vision algorithms. These can be subdivided into two principal categories: global and local features. Global features describe a property of the entire image, such as statistics about the luminance or color, whilst local features describe the properties of only a small region.

The key advantage of local features is that extracted information can be attributed to a particular location in the image; this is crucial if, for example, an object is being tracked or detected within an image. Although surprising, if applied to a tightly constrained problem, global features can yield encouraging results. For example, wildlife frames containing quadrupeds can be detected using the image's power spectrum [265], which effectively describes the dominance of each frequency in constructing the image.

Some global features may be learned adaptively for a specific video clip, for example, statistical techniques such as *principal component analysis* (PCA) can be used to project entire frames into a two- or three-dimensional space allowing a complete video to be visualized easily. Furthermore, clustering this low dimensional representation permits automatic keyframe extraction [108].

However, the limitation of global features to provide information about specific regions of an image constraints their use in video analysis; their strength lies in applications where the interest is in looking at large scale properties of an image sequence, for example to detect shot boundaries, or for classification problems where the domain is very constrained.

Low-level features can either be generated exhaustively at every point in the image, in which case a higher-level learning algorithm can be used to select the set of features that are most relevant to a particular problem, or interest point detectors can be used to detect image regions of interest automatically. Different interest point detectors regard interesting features in different ways. For example, the Kanade-Lucas-Tomasi feature tracker [261] defines an interest feature as an image patch with a gradient matrix with two large eigenvalues. Other standard interest feature detectors include the Harris corner detector [123], Förstner-Gülch [101], and the Reisfeld symmetry operator [246].

Within the last decade, invariant local features became popular. These approaches include *scale-invariant feature transform* (SIFT) [198] or *speeded up robust features* (SURF) [25] that rely for scale adaption on the scale-space theory introduced by Lindeberg [192]. Other techniques, such as *maximally stable extremal regions* (MSER) [209], intrinsically adapt the detected region size. A variety of affine interest point detectors as well as suitable region descriptors are evaluated by Mikolajczyk et al. [213, 214]. A

recent evaluation of the matching performance of several detector-descriptor combinations for 3D object features is provided by Moreels and Perona [218].

Low-level features used by machine learning techniques to train classifiers or detectors include simple rectangular features that are fast to compute and capture large scale structure as well as some information on image texture [292], *histogram of orientated gradients* (HOG) features [71], which primarily capture information about image gradients, and *local binary patterns* (LBP) [6], which capture texture. These features are designed to be fast to compute and offer some robustness to noise or small changes in, for example, the illumination or orientation of the object. These features are often much simpler than their interest point detector counterparts and therefore less discriminative.

Thus far, all features presented are only spatial in nature. However, often these features can be extended to the temporal domain, for example, in the form of a temporal extension of the SIFT feature [257], temporal Gabor filters [79], temporal Harris corner features [180], and temporal simple rectangular features [157]. Typical uses for these types of features are for video retrieval or action recognition. A discussion on spatio-temporal interest-points and an evaluation of volume descriptors is presented by Laptev and Lindeberg [180, 181]. Whilst all of the above features are hand-designed, a promising technique is to use machine learning techniques to engineer low-level features automatically [184].

3.1.2 High-Level

In this section, high-level methods used to extract information from video sequences are reviewed. These are split into three categories: *recognition and detection*, *tracking*, and *3D reconstruction*.

Recognition and Detection

Recognition and detection can both be seen as classification problems. However, the difference between them is that detection can be seen as a two-choice classification problem, and recognition as a 'one of N' classification problem. Counterintuitively, this does not imply that detection is an easier problem. Take for instance a pedestrian detector: whilst the positive class is well-defined, the negative (no pedestrian) class must represent every possible image that does not contain a pedestrian; of course, this image class is infinite and cannot be accomplished. A recognition task, however, is often more constrained. For example, given a text character, which letter is it most likely to be?

A recognition or detection system is composed of two parts: a set of low-level features, such as those discussed above, and a classifier, which will be trained using examples of each class. Popular classifiers include *decision trees*, *neural networks*, *AdaBoost*,

support vector machines (SVMs), and *k-nearest neighbors* (KNN). There are several well-documented implementations of all of these classifiers and a good introductory text to machine learning is provided by Bishop [29]. All of the above methods are trained using a set of positive and negative labeled examples, and cross-validation may be used to prevent overfitting to the training data.

The typical approach to object detection is using a sliding window to exhaustively test whether an object is located at each pixel location in the image at varying scales. For example, this method has been used for face detection using AdaBoost combined with rectangular features [292] and pedestrian detection using an SVM combined with HOG features [71]. For the detection of objects that exhibit a lot of variation in appearance due to changes in orientation or articulation, a part-based method may achieve improved results (e.g., Felzenszwalb et al. [96]). Modeling context can also be used to improve detection accuracy (see Divvala et al. [78] for a recent review).

For classifying sequential data, *hidden Markov models* (HMMs), which are commonly used in speech recognition, remain a popular choice, for example, to classify the trajectories of the hands performing different gestures [306] or martial art actions [272]. Recently, combining temporal features and using classifiers such as those discussed above became popular [157, 180, 79]. For example, temporal corners are used to detect sudden changes in motion present in actions such as walking or bouncing a ball [180]. Subtle actions such as grooming, eating, and sleeping performed by rodents have been recognized using *Gabor filters* applied to the temporal dimension of an image sequence [79].

One of the difficulties with action recognition is the lack of clarity where exactly (in a temporal sense) an action starts and where an action finishes. This can lead to difficulties in creating a consistent training set of positive and negative examples for a given action. However, methods such as multiple instance learning can be used to address this problem. This requires that for each positive example a positive event is known to have occurred without specifying the exact temporal location or duration. This has been applied to, for example, the detection of shoppers picking items off a shelf [146] and automatically learning sign language from TV subtitles [42].

Tracking

Tracking and detection are closely related. If detection was 100 % accurate, tracking would be redundant; an object could simply be located in an image in each frame independently. However, this is currently not the case and tracking exploits knowledge of an object's location in a previous time instance to make a prediction, and thus narrow the search space of the object's location at the present time. Most tracking algorithms assume detection or initialization in the first frame to be a separate problem. The integration of tracking and detection into a common framework still remains an

open problem in computer vision, although some recent attempts have been made (e.g., Andriluka et al. [12]).

There are a small number of established tracking algorithms, most notably the *Kalman filter* and the *particle filter*. The Kalman filter (tutorial provided by Welch and Bishop [301]) assumes Gaussian noise and a linear dynamics model, whereas the *particle filter* (a tutorial is provided by Arulampalam et al. [14]) is a stochastic approach and as such makes no assumption about the underlying probability distributions or dynamics. Each has a number of variations, the most popular is the *extended Kalman filter* [301], which modifies the Kalman filter to incorporate nonlinear dynamics, and the *annealed particle filter* [77], which uses the method of simulated annealing to allow the stochastic search of the particle filter to be performed efficiently.

Most recent developments made in the field of tracking have been domain specific, in particular modeling the solution space or system dynamics of a particular problem. For example, in case of 3D human pose estimation, methods such as Gaussian process models [289] or PCA [317] have been used to learn action specific models (e.g., walking) to reduce the dimensional space for tracking. For tracking individuals in crowded environments, models of social interaction have been learned to predict how people will behave to improve the performance of tracking algorithms [229].

Tracking can also be made more robust by learning the appearance of the object online. For example, learning the appearance of individual limbs while tracking articulated objects [241], or adapting an offline trained classifier to a specific instance of an object observed during run time [104].

3D Reconstruction

For visual analytics of video data, 3D reconstruction is especially important to determine the 3D locations of objects and for visualization, but also the 3D object appearance may be of interest. There are many methods used to extract a 3D representation of a scene or object observed in video. Well-established methods include approaches such as *structure from motion* (SfM) [75], *space carving*, and *stereo reconstruction*. A brief discussion of these well-established techniques together with methods that typically attempt to reconstruct 3D structure from single images using cues such as shading, shape, and texture is provided in this section. A good overview of vision-based 3D reconstruction is provided by Szeliski [274], and a recent survey, focusing on reconstruction from video, is provided by Stoykova et al. [269].

SfM takes a set of images and attempts to extract both a 3D reconstruction and the camera's motion. This is typically achieved by finding point correspondences across multiple images. The main benefit of SfM is its relatively inexpensive nature both in terms of computation and memory; to date entire cities have been reconstructed [4]. Furthermore, a dense reconstruction can be estimated by *a priori* knowledge, such as

assuming all surfaces are planar [103] or by applying stereo matching using the initial structure as a set of constraints. The principal assumption in most SfM algorithms is that the scene is rigid and any motion observed is due to either camera motion or motion of the entire scene moving as a rigid entity. Non-rigid motion of the face or simple deformable objects (e.g., a shoe) have been accommodated in the SfM framework by extracting a set of rigid 3D basis shapes. This allows constructing the objects in each frame from a linear combination of the basis shapes [284, 283]. An approach that can cope with much larger deformations is to segment the object into a piecewise model and reconstruct each piece independently. The problem is then to robustly segment features, which can be achieved using energy minimization [254]. The algorithms used in SfM are relatively mature and well understood, and a number of commercially available software packages exist. As the process of 3D reconstruction becomes automated, it is desirable to exclude objects not demanded in the final 3D reconstruction. For example, in a reconstruction of a city, cars and pedestrians could be automatically detected and removed [65]. There are many good tutorials and textbooks on SfM, such as that of Hartley and Zisserman [124] or more recently Moons et al. [216].

Another method closely related to SfM, is vision-based *simultaneous localization and mapping* (SLAM), where feature tracking is performed online to create a sparse map and track the camera within this map. Typically, the focus in this work is on achieving real-time performance using a typical consumer web cam [74]. This has made the approach particularly applicable to create augmented realities [53, 167] with SLAM systems that can operate on cell phone devices [168]. For the purpose of augmented reality, a dense reconstruction is typically not needed, since higher-level structures such as planar surfaces can be inferred from the extracted sparse representation. While the typical SLAM framework used for tracking a monocular camera tightly entwines the tracking and mapping into a single estimation problem, a recent approach is *parallel tracking and mapping* [167]. This method performs tracking and mapping as two separate tasks to process the mapping as batch operation using a much larger set of features, while still performing tracking on a frame-by-frame basis. This results in an approach that is more robust to tracking failure and produces maps that are more accurate.

Stereo matching across two images can be used to create a dense depth map. Typically, an important step in this process is image rectification, where the epipolar lines of the images are rectified to be parallel to the horizontal axis of the image. This allows matching between pixels to be performed along scan lines of the image rows. The positional difference between two pixels that observe the same point is called disparity. The matching of pixels to estimate the disparity between two images can be performed via a number of standard optimization techniques, such as *dynamic programming* [290], *level sets* [92], *graph cuts* [293] and *loopy belief propagation* [97]. A survey and evaluation of existing stereo reconstruction methods is provided by Seitz

et al. [258].

An alternative technique to dense reconstruction is *space carving*, which requires a predefined 'search space' to be constructed in which the object or scene of interest is assumed to be contained. This space is split into voxels, each voxel is projected into every frame, and a measure of consistency is extracted. If a voxel is consistent across all views it is assumed to be on the surface of the object of interest, otherwise it is discarded [178]. In this approach, it is typically assumed that the cameras are fully calibrated and the surface of the object is Lambertian. Another approach is to use foreground silhouettes to extract the visual hull of an object [183]. Results from this method are typically poorer than those using color consistency or texture, but the method can be used to initialize approaches that are more complex. A review of methods used to extract a 3D reconstruction of complex, often moving, and deformable objects from multiple views in a studio setting is provided by Starck et al. [267]. A discussion of practical issues such as illumination and camera placement is also presented in that work.

Some approaches extract 3D structure from single images, which obviously can be applied also to video sequences. Although cues such as shading [82] or texture [195] can be used to extract some information about 3D structure independently, most approaches tend to achieve accurate results by making assumptions about the scene or the object being viewed. For example, in estimating the 3D shape of a human face, a 3D geometric prior model may first be learned to constrain the solution space [252]. Machine learning approaches are also popular to learn a regression from 2D binary silhouettes to 3D human poses [3]. To allow reconstruction of more unconstrained images, a classifier may be learned to identify different image elements (e.g., sky, ground, or buildings) [141]. For reconstruction of structured objects, such as buildings, a grammar can be learned that describes how different architectural features should relate to one another [170].

In the majority of cases, current monocular approaches tend to achieve quantitatively poor results compared to those using multiple views. However, for many tasks the results are qualitatively acceptable. Furthermore, for sequences with marginal texture, making assumptions about the environment may be the only method to resolve many of the ambiguities that exist. It is likely that the area of 3D reconstruction coupled with machine learning techniques will continue to receive much attention in future.

3.2 Trajectory Extraction

Due to various capabilities of the feature trajectory for filtering, visualization, and data mining techniques, the methods applied for automated trajectory extraction in the prototypical system are discussed in more detail in this section. The approach

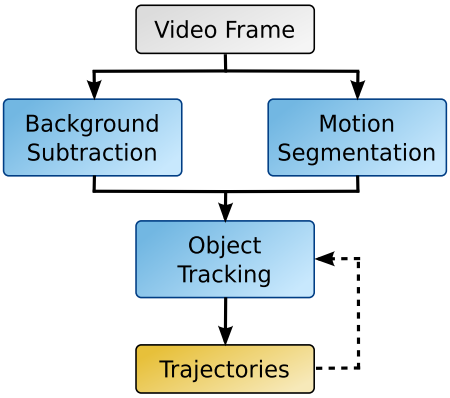


Figure 3.2 — Workflow of automated trajectory extraction.

combines well-known and basic computer vision techniques to achieve simple but robust feature extraction (see Figure 3.2).

Background subtraction classifies the pixels of a video frame as foreground and background utilizing a background model. Assuming that the background is to some extent static, the approach applies the *running Gaussian average* method [311] that utilizes a single Gaussian luminance distribution per pixel to describe the background. The background model is updated by the currently processed video frame to cope with noise and gradually changing illumination. Foreground objects are retrieved by thresholding the subtraction of the current video frame and the background model.

By *motion segmentation*, areas of homogeneous movement, which are considered foreground objects, can be identified. To calculate the required optical flow field of successive frames, the pyramidal *Lucas-Kanade method* [36] is applied. These two segmentation approaches (background subtraction and motion segmentation) are combined to avoid wrong trajectories and false alarms due to encoding artifacts or badly initialized background models. Finally, the proposed method refines the object regions by morphological operations utilizing prior knowledge of the video material (e.g., codec block size) and fuses the results of both segmentation methods. The fused object blob as well as the results of both segmentation methods are illustrated in Figure 3.3.

Object tracking integrates the observation of foreground regions of several frames. For this purpose, the proposed method utilizes a linear *Kalman filter* [301] with a dynamics model of second-order derivative to trace the extracted blobs. To support multiple object tracking, the object's dynamics model represented by the Kalman filter is used to predict a target window (red circle in Figure 3.3). Based on this prediction, target windows and object observations are associated by the *Hungarian algorithm* (also termed Munkres), according to their distance. The new observations are used to

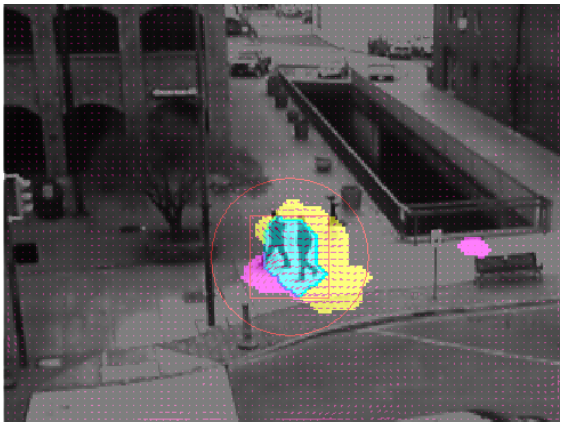


Figure 3.3 — A video frame (from the IEEE VAST Challenge 2009 dataset [1]) superimposed by the results of the different segmentation approaches. Blobs stem from background subtraction (magenta), motion analysis (yellow), and the fusion of both results (cyan). Small pink lines illustrate the optical flow. The object's position in the previous frame is represented by red rectangle, while the red circle limits the prediction window.

update the Kalman tracker. For trajectories without associated observation, for instance because of object occlusion, the proposed method applies an update rule according to Cipra and Romera [61].

Different trajectory representations and matching strategies exist in literature (see survey of Broilo et al. [39]). As fundamental representation of the trajectories of moving objects extracted from video, a piecewise linear approximation is chosen. Each trajectory T is represented by an ordered set of temporally equidistant samples t_i, and each sample includes its corresponding information. The number of samples n is determined by the duration of the trajectory and the sampling rate of the video (fps): $T = t_1, t_2, ..., t_n$. This representation without reducing the sample points is chosen, since it features the most accurate representation of the extracted trajectories, and similarity measures for filtering and data mining (see in Chapter 4.1) are easily applicable.

Blob information (i.e., the contour of the tracked object) is stored for each sample of a trajectory and further properties are derived therefrom. Among these features are the object's axis-aligned bounding box, its barycenter, and the object's contact point on the ground plane, which was projected to a virtual top-view for uniform scaling in the presence of perspective in the video data. Based on the video data and on these fundamental properties of each sample, a variety of facets $\mathcal{F}$ of each object's trajectory is further calculated. These facets can be utilized by subsequent stages.

To date, the system provides five facets of an object's movement characteristics that turned out to be the most important ones. These properties are inspired by the movement descriptors by Laube et al. [182], which they use to specify and compare geospatial lifelines captured by GPS devices. The facets are position, velocity, movement azimuth, time, and object class (which are provided by manual labels, but can be generally also classified by appropriate automated methods exploiting the object's appearance). However, a multitude of other object properties could be imagined.

To draw correct conclusions, the analysts have to know about the quality of the information provided by the automated extraction process. Therefore, the algorithmic uncertainty is calculated and can be accessed in later stages, for example, for uncertainty-aware filtering or uncertainty visualization. Since every step of the trajectory extraction process introduces (or at least propagates) uncertainty, the quality of each result has to be evaluated and, finally, fused together to a combined uncertainty estimation.

The used uncertainty model is based on the following assumptions and simplifications. The model neglects the uncertainty originating from the video capturing process, which basically reflects sensor noise and coding noise. Its influence is considered as negligible since the effect of noise is alleviated by morphological post-processing of the segmentation result. Further, the model focuses only on data uncertainty and in general omits handling of missing values and contradictory data. Thus, it is possible that some objects (i.e., blobs, trajectories) will be missed due to inappropriate thresholds. Likewise, it does not consider any systematic error that leads to reduced *accuracy*[3]; it solely regards random error that affects the *precision*[4] of the result. Finally, it assumes temporal precision of video sampling to be high enough for its purpose. Hence, temporal precision is ignored and only spatial precision of the extracted features is considered.

The different computer vision methods applied introduce their own methods of how measurement uncertainty is estimated. To model uncertainty inherent in the two segmentation approaches, the approach calculates each pixel's probability of being correctly assigned to the foreground or background class. The uncertainty of both segmentation results is then fused by averaging. Regions where blobs of both methods overlap are considered as object areas and are further tracked (see Figure 3.3). Similar to Pfoser and Jensen [232], the approach only maintains trajectories of moving points, in contrast to traces of whole blob areas. Hence, the mean and covariance of the blob area weighted by the segmentation quality is calculated.

The classification uncertainty of a pixel is retrieved by widely accepted methods and according to the segmentation approaches. For background subtraction, the approach

[3] closeness of the result to the ground truth
[4] repeatability of the result

exploits the standard deviation of the background model to specify the likelihood of being well classified. The quality of the optical flow field obtained by the Lucas-Kanade method is largely determined by the cornerness of the tracked image region, which is due to the aperture problem and exploited by the popular Kanade-Lucas-Tomasi tracking technique [261]. According to Barron et al. [23], the approach uses the magnitude of the smallest eigenvalue of the system matrix to determine the uncertainty of the estimated optical flow.

In the Kalman filter, the approach treats the mean of the uncertainty weighted blob area as the observation of the object's location, while its covariance is considered as the normal distributed measurement noise. In consequence, the trajectory is modeled by the location (filter state) and the positional precision (a-posteriori error covariance matrix) of the moving point in each frame. The square root of the error covariance can be used, for example, for visualization of trajectories' positional precision as in Chapter 6.4 according to standards of measurement uncertainty [277]. The temporal sampling of video frames is assumed dense enough to ignore positional deviation of the trajectory between two consecutive samples. If observations were missed for some frames, for instance in cases where the object is occluded, the trajectory is extrapolated according to its dynamics model. Inaccuracy of the dynamics model in combination with the process noise (which covers the simplifications of the dynamics model) leads to increasing uncertainty of the predicted positions.

In case of missing observations, the proposed method of modeling the trajectory uncertainty is quite similar to lifeline beads [232]. The main difference between lifeline beads and the proposed approach is that the former uses a uniform distribution to model the object's likelihood of being at any location within a bead, which is constrained by the object's estimated maximal velocity. In contrast, the approach based on Kalman filtering uses a multivariate Gaussian distribution with respect to the recent dynamics model and its accuracy to define the probability density function of the moving object's location.

Based on the ground plane projected locations of the moving object, the further movement descriptions (the facets $\mathcal{F}$) are extracted. The proposed approach can calculate these facets on different levels of granularity depending on filter configurations and the visualization requirements (see Chapter 4.1). For simplification, the descriptions of the whole trajectory are described below. Nevertheless, the calculations are identical for each segment of a split trajectory (according to a particular granularity), in general.

Movement descriptions, such as the facet velocity

$$\mathcal{F}_{\text{velocity}}(T) = \frac{\text{fps}}{n-1} \sum_{i=1}^{n-1} \sqrt{\left(\mathcal{F}_{\text{pos}}(t_i) - \mathcal{F}_{\text{pos}}(t_{i+1})\right)^2} \tag{3.1}$$

or movement azimuth

$$\mathcal{F}_{\text{azimuth}}(T) = \frac{180}{\pi} \cos^{-1}\left(\left[\frac{\mathcal{F}_{\text{pos}}(t_1) - \mathcal{F}_{\text{pos}}(t_n)}{\left\|\mathcal{F}_{\text{pos}}(t_1) - \mathcal{F}_{\text{pos}}(t_n)\right\|}\right]^T \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right) \tag{3.2}$$

are solely based on the object's locations. Hence, the proposed method uses error propagation [212] (also called *Gaussian error propagation* or *linear error propagation*) to calculate their uncertainties. Since the uncertainties are assumed Gaussian distributed, they are expressed as the standard deviations σ_{velocity} and σ_{azimuth} (the derivation can be found in [137]):

$$\sigma_{\text{velocity}}^{(1,n)} = \frac{\text{fps}}{n-1}\sqrt{\sum_{i=1}^{n-1}\frac{(v^{(i,i+1)})^T C_v^{(i,i+1)} v^{(i,i+1)}}{(v^{(i,i+1)})^T v^{(i,i+1)}}} \tag{3.3}$$

$$\sigma_{azimuth}^{(1,n)} = \frac{180}{\pi}\sqrt{\frac{1}{1-\frac{(x^{(1,n)})^2}{(v^{(1,n)})^T v^{(1,n)}}}\left[\frac{var_x^{(1)} + var_x^{(n)}}{(v^{(1,n)})^T v^{(1,n)}} + \frac{(x^{(1,n)})^2((v^{(1,n)})^T C_v^{(1,n)} v^{(1,n)})}{((v^{(1,n)})^T v^{(1,n)})^3}\right]} \tag{3.4}$$

with vector

$$v^{(i,j)} = \begin{pmatrix} x^{(i,j)} \\ y^{(i,j)} \end{pmatrix} = \mathcal{F}_{\text{pos}}(t_j) - \mathcal{F}_{\text{pos}}(t_i)$$

and the according covariance matrix representing the uncertainty

$$C_v^{(i,j)} = \begin{pmatrix} var_x^{(i,j)} & cov_{x,y}^{(i,j)} \\ cov_{y,x}^{(i,j)} & var_y^{(i,j)} \end{pmatrix} = C_{\text{pos}}^{(i)} + C_{\text{pos}}^{(j)}$$

where $\mathcal{F}_{\text{pos}}(t_i)$ denotes the trajectory's position vector $v^{(i)} = \begin{pmatrix} x^{(i)} \\ y^{(i)} \end{pmatrix}$ at sample i with the according covariance matrix $C_{\text{pos}}^{(i)}$; var_x/var_y and cov_x/cov_y are the x/y variances and covariances. The superscript represent the time index of the sample or the vector between two samples, for example in the case of (i, j). Note that the errors in $\mathcal{F}_{\text{pos}}(t_i)$ and $\mathcal{F}_{\text{pos}}(t_j)$ (for $i \neq j$) are assumed to be independent and uncorrelated.

3.3 Example: Computer Vision for Snooker Visualization

As discussed in the beginning of this chapter, the subsequent stages of the video visual analytics pipeline (e.g., *filtering* and *visualization*) determine the features that are calculated in the *feature extraction* stage. This section gives a more practical view on *feature extraction* and provides an example of a video processing pipeline as required by the snooker skill training visualization, which will be introduced in Chapter 6.5.

It is common practice for automated image and video understanding to constrain the degrees of freedom of the problem domain by *a priori* knowledge. For industrial computer vision solutions, the problem complexity is often further reduced by constraining the environment, for example, by the use of constant artificial illumination or fixation of the object pose by fitting the object into a mounting. In practice, even preparatory stages are introduced to determine and fixate the 3D object location (e.g., Höferlin et al. [133, 135], Class et al. [62, 63]) and hence, reduce the problem complexity successively. Additionally, fusion of data from multiple sensors and *a priori* knowledge about sensor characteristics are often utilized to improve the computer vision results further. In RoboCup[5] (soccer playing robots), for example, some teams mount two different kinds of cameras on the robots: an omnidirectional and a perspective camera [319]. Here, the characteristics of these cameras are exploited for robust 3D ball localization. The omnidirectional camera with its panorama view has only low spatial resolution but is able to estimate the direction of the ball precisely. In contrast, the perspective camera estimates ball directions worse due to decalibrations in consequence of collisions with other robots, but has excellent spatial resolution due to its directed view. For robust 3D ball localization, the reliable information of both are fused (i.e., ball direction from the omnidirectional camera, ball distance via ball size estimation from the perspective camera) [154]. These examples give an impression of strategies pursued for designing computer vision pipelines.

In the following, a particular video processing pipeline is discussed. The background of the subject is a feasibility study in collaboration with a snooker club with the aim to find out whether video visualization can be deployed to aid snooker skill training. The application background, the feasibility study, and the resulting snooker visualizations are discussed in detail in Chapter 6.5.

One objective of this feasibility study is to assess if coaches can recognize visual signatures [55] from video visualization. This requires appropriate video material as well as extraction of different temporal features.

[5] www.robocup.org

3.3.1 Data Capturing

The snooker club considered to invest in a suite of ceiling-mounted and computer-controlled video capture and replay equipment, but was uncertain whether performance analysis based on watching videos is scalable from training a few professional players to many amateur players. Thus, the proposed ceiling-mounted video capturing equipment was not available to the feasibility study and portable and relatively low-cost equipment was used to capture videos from different snooker shots.

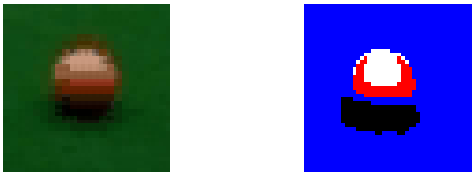
Two filming session were held. The first session allowed finalizing the equipment requirements. All videos used for the feasibility study were filmed in the second session. Therefore, two Casio Ex-FH20 cameras were used that support high-speed filming at up to 1000 fps. After the trial run, the decision was made in favor of 420 fps at 224×168 resolution. This setting allows capturing high-speed actions, such as ball spin and cue vibration, which the naked eye cannot easily observe. Although the resolution is less desirable for both video processing and visualization, it provides a worst-case scenario to test the technology. As a snooker hall is usually not well-lit and high-speed filming demands good lighting, four 500 W halogen floodlights mounted on two telescopic masts were used. Without computer-controlled camera synchronization, a 20 Hertz (Hz) strobe light was used to help synchronizing videos in the processing stage.

Four videos were used ("pink-pot-b1", "pink-pot-b2", "pink-pot-c1", and "pink-pot-c2") to assess the technical feasibility and to provide the evaluation with a practical case study. The videos show two different shots, b and c, each captured from the side (1) and the front (2). Some keyframes of the front view videos are shown in Figure 6.32 (e–f). The feasibility study focused on a particular cue action namely *spin avoidance* (see Chapter 6.5.1 for details). It is not feasible for human or machine vision to quantify spinning from normal snooker cue ball. Hence, a training cue ball, with its two halves colored in black and white respectively, was used.

3.3.2 Video Processing Pipeline

For the two above-mentioned shots and their video sequences (like those in Figures 6.32 (e–f)), the video visualization requires the following features:

1. the silhouette of a ball,
2. the different color segments of a ball,
3. the center of a ball, or of each segment,
4. the color separation line on the black-white cue ball.



(a) Original (b) Color segmentation

Figure 3.4 — Segmentation of the scene by classifying pixels according to colors will lead inevitably to false classifications at shadows and specular highlights.

There are many solutions in image processing literature for such problems. However, one common hindrance is that one can rarely apply an existing solution to a different problem domain. Like most previous work in this area (e.g., Daniel and Chen [72], Botchen et al. [35]), the problems to deal with included low-resolution images and MJPEG compression artifacts. Additional difficulties were specular highlighting on, and dark shadows cast by, snooker balls, because of the lack of diffuse lighting in the environment.

Objects in a snooker scene are colorful. Naturally, one would like to recognize objects by classifying pixels according to colors. However, using this straight forward approach to segment the scene will lead inevitably to false classification at shadows and specular highlights (see Figure 3.4). The problem with this approach is that pixel colors of shadows are similar to black pixels on the cue ball, while pixels of highlights are similar to white pixels on the cue ball. Therefore, it is impossible to distinguish those pixels without context information.

To overcome this problem, color information is not used until context information is available. Figure 3.5 shows an overview of the designed video processing pipeline.

Ball Detection and Tracking. The approach acquires context information by using the *generalized symmetry transform* proposed by Reisfeld et al. [247]. This measure of local symmetry utilizes three terms to calculate a symmetry map: a phase weight function, a gradient weight function, and a distance weight function.

For preparation, each frame is first converted to a gray value image and convolved with a Gaussian kernel to reduce noise. Next, the approach extracts edges in horizontal and vertical directions using a Sobel edge detector, and calculates the magnitudes and angles of the edges. These steps are applied using OpenCV [38]. Since it is easy to modify the distance weight function in order to pass only a specific radius, the approach applies the generalized symmetry transform to each possible radius separately, and searches for the global maxima which lead to the circle centers and radii. These centers and radii are tracked by OpenCV's linear Kalman filter.

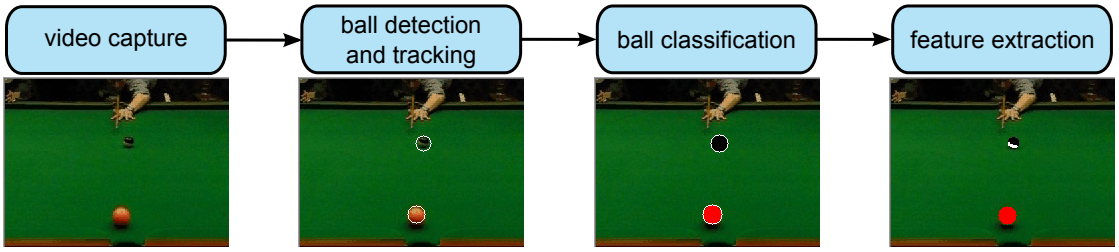


Figure 3.5 — The basic steps of the video processing pipeline for snooker skill training.

Ball Classification. Once the circular boundaries are detected, the approach classifies the balls according to pixel colors in their area. Pixels with little color distance to black or white vote for the black-white cue ball, while pink pixels vote for the pink ball. Of course, the shadows and highlights of the pink ball will vote for the black-white cue ball, but they also contain a substantial amount of unambiguous pink pixels leading to a robust segmentation result.

Feature Extraction. After these steps, the approach further segments the black-white cue ball in the same way into black and white segments, calculates the segment centers, counts the number of pixels in each segment, and finally estimates the separation line.

Results. The above-mentioned processing pipeline works robustly on the specific case with two balls. Nevertheless, problems may occur in scenes containing balls with similar colors, and varying illumination conditions. In these cases, algorithms that are more sophisticated should be used. This is the downside side of highly specialized computer vision pipelines: they are only applicable as long as the constraints are met that were fed into the algorithm as *a priori* knowledge. For extracting other features, such as grip and wrist motion, a new video processing pipeline is required.

Chapter

4 Filtering[1]

Data reduction by filtering is one of the main concepts in the video visual analytics pipeline for data scalability, besides data aggregation. Moreover, the filtering stage plays a versatile role in the proposed visual analytics system. Users can apply filters to define continuous queries to retrieve particular data instances of video data and its calculated features. Continuous queries are a concept from data stream management systems. In contrast to static queries, continuous queries "are evaluated continuously as data streams continue to arrive" [18]. By using filters as tools for IR, analysts can verify their reasoning hypotheses against the data. Due to these reasons, the filters are processed prior to the relevance measure and visualization stages.

The presented visual analytics system provides various ways of filter definition according to the different requirements of the users. However, all definition methods provide visual support and context information, such as charts of data distributions and video context information. The provided filters mostly apply to features extracted from video, such as blobs and trajectories, but can provide relevance feedback (see the subsequent chapter) to the raw video stream as well. Therefore, the term data instance is interchangeably used for video frames and their extracted features. However, the main focus here is the query formulation for trajectory-based video retrieval [39].

In cases where the information need of the analyst can be specified (e.g., KIS), a group of filters allows the users to constrain the continuously retrieved data instances *by properties*, such as the position, direction, velocity, and lifetime of trajectories (see Figures 4.6, 4.7, and 4.8). Each of these filters allows constraining the range of accepted

[1] Based on Höferlin et al. [132], Höferlin et al. [140] (Chapter 4.1), Höferlin et al. [137] (Chapter 4.2), and Höferlin et al. [130] (Chapter 4.3).

values of one particular property. To help the analysts determine proper ranges for the filter definition, the property distribution and video context are additionally displayed. Furthermore, fuzzy filter definitions account for potential vagueness and uncertainty of the analysts' query.

An alternative approach of specifying a filter model is its definition *by sketch* (see Figure 4.10). This definition interface allows the users to sketch a trajectory freely on top of a context video frame. After sketching the trajectory, the users may select the type of properties of the trajectory that will be included in the signature for retrieval. This way, a more detailed query (compared to the property filters) can be formulated very fast.

An exploratory approach to filter creation is their definition *by examples* (see Figure 4.5). This method can be used to search for similar trajectories according to various facets or to avoid sketching a trajectory from scratch by utilizing the query as starting point. Moreover, the users are allowed to transfer selected data instances directly into white or black list filters. Scatter/gather browsing by ISS, which will be introduced in Chapter 6.3, further allows users to define such a list filter by one click. However, black and white lists of data instances do not generalize to new and unseen data that was not added to the list. To enable generalization, users can decide to convert a list filter into a decision tree filter, which is trained on the examples of the black list and white list.

Another possibility for filter definition is the ad-hoc training of a classifier model by an inter-active learning scheme (Chapter 4.3). The advantage over pre-trained classifiers, which are widely used (e.g., person or car detectors that are included in many video management systems), is flexibility: new classifiers can be trained on-the-fly by multiple examples.

The previously mentioned filters only consider one data instance at a time. Their decision on pass or reject of the trajectory is also only based on the trajectory and the filter model. Another type of filters that consider the relationship between two trajectories is required, to enable aggregation queries (see Figure 4.9). However, continuous aggregation queries come at the cost of approximate answers calculated by sliding windows or any form of model prediction approaches [18]. Hence, filtering for trajectory relationships requires the definition of a time window besides other interaction characteristics.

The particular filter definitions can further be arranged in a filter graph for each data type (e.g., trajectories) and data source in the analysis project (see Figure 4.1). At the time of evaluation, the stream of data traverses this graph from the defined input to the node in the graph marked as output node. This way, analysts can construct complex queries that consist of arbitrary combinations of filter expressions. Switching the output node or rearranging the nodes' connectivity enables the analysts to use the filter graph both as query history and as toolbox of modules and alternative filter

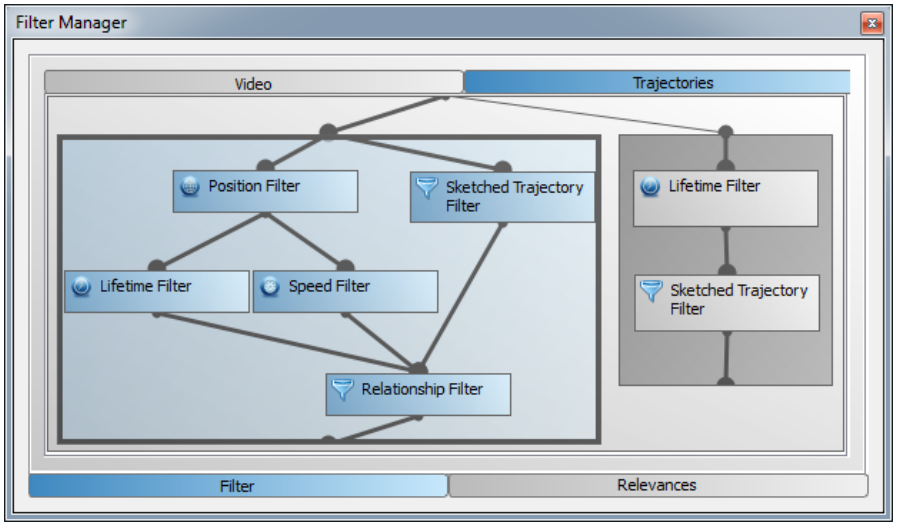


Figure 4.1 — The filter manager shows a combination of different filters in a filter graph. Connected nodes are conjunctions of filters and parallel routes denote disjunctions. Filters can be organized in containers (here: two containers) and activated or deactivated (see the right container). The filter manager dialog was partially developed in the context of a supervised diploma thesis by Engelhardt [89].

branches. To support the more complex usage of the filter graphs, branches can be encapsulated in containers and used as a single filter unit. To enable users to keep track of the semantics and purposes of the filter definitions, filters and containers can be annotated. These descriptions are given by users to explain the intention of the filter (such as "Exclude streets" or "Meeting") and provide an overview. In principle, there are two types of filter containers: container for crisp filters (e.g., white or black list filters), and fuzzy filter container. Containers for fuzzy filter definitions further feature defuzzification functionality and can be applied as relevance measure, which is discussed in Chapter 5. In fuzzy sets, intersection can be applied by any *t-norm* and union by the corresponding *t-conorm*. The most common t-norm and t-conorm for fuzzy sets are Zadeh's *min* and *max* [316] operators, which are applied by default. However, the users are not restricted to them and may apply any t-norm and t-conorm. An introduction to fuzzy sets and their operators can be found in the textbook of Siler and Buckley [264].

The evaluation of the filters requires distinct methods. For example, trajectories on the white (black) list are accepted (rejected) solely based on their unique object ID. Filtering by properties apply the t-norm to the query interval and the object property. Here, the property of the object is mostly considered on a global view, such as the mean velocity or the mean movement azimuth during the whole lifetime of a trajectory.

Other kinds of filter definitions requires a measure to compare the similarity of two trajectories, such as filtering by sketch. In this case, the sketched trajectory has to be compared to the trajectories in the video stream. The choice of the similarity measure is essential for the results and must be selected according to the particular purpose. Due to this importance, the next section introduces a novel and flexible metric to measure similarity between trajectories in the video visual analytics system.

4.1 Similarity Measures for Trajectories

The proposed approach generalizes the *lifeline context operators* by Laube et al. [182] by distinguishing between facets and similarity measures. Both, the facets and similarity measures are inspired by their work. In fact, the method provides three types of similarity measures for each facet. The types of supported similarity measures are *coverage* $\mathrm{D_c}$ (see Eq. 4.1), *distance between means* D_μ (see Eq. 4.2), and *distance between standard deviations* D_σ (see Eq. 4.3). The coverage measure evaluates the symmetrical overlap of facet (or feature) values $\mathcal{F}$ between two trajectories T and T' with samples t_i, t_j as

$$\mathrm{D_c}(T, T', \epsilon) = \frac{1}{2}\left(\sum_{t_i \in T} \frac{\mathrm{c}(t_i, T', \epsilon)}{|T|} + \sum_{t_j \in T'} \frac{\mathrm{c}(t_j, T, \epsilon)}{|T'|}\right) \tag{4.1}$$

with

$$\mathrm{c}(t, T, \epsilon) = \begin{cases} 0, & \min\limits_{t_k \in T}(\mathrm{d}(\mathcal{F}(t), \mathcal{F}(t_k))) < \epsilon \\ 1, & \text{otherwise} \end{cases}$$

The constant value ϵ represents the range of the facet value considered as overlap. The choice of ϵ depends on the sampling rate of the trajectories. In the case of surveillance videos, a fixed sampling rate in the range of 5–30 samples per second is assumed. $\mathrm{d}(\cdot)$ denotes the distance function, where the Euclidean distance, which can be multi-dimensional for particular facets $\mathcal{F}$ (e.g., velocity is one-dimensional; position is two-dimensional), is used.

Distance between means uses a segmentation of the compared trajectories into κ parts (i.e., the measure is *episodal* according to the terminology of Laube et al. [182]). The κ segments of each trajectory have equal duration and the features are linearly interpolated if necessary (i.e., if the number of sample points of the trajectory is not a multiple of κ). With segmentation function $\mathcal{S}$ and with $S_k = \mathcal{S}(T, \kappa, k)$, $S'_k = \mathcal{S}(T', \kappa, k)$ representing the set of samples of the k-th segment of trajectories T and T', the distance between means is defined by

$$\mathrm{D}_{\mu}(T, T', \kappa) = \frac{1}{\kappa} \sum_{k=1}^{\kappa} \mathrm{d}\left(\mathrm{E}(S_k), \mathrm{E}(S'_k)\right) \tag{4.2}$$

where $\mathrm{E}(S)$ is the arithmetic mean

$$\mathrm{E}(S) = \frac{1}{|S|} \sum_{t_i \in S} \mathcal{F}(t_i)$$

Using a variable number of segments, the users can choose the granularity of the temporal development of a trajectory's facet with respect to their goals. In the same way, the distance between standard deviations uses trajectory segments with granularity κ. With the segment variance function

$$\mathrm{Var}(S) = \frac{1}{|S|} \sum_{t_i \in S} \left(\mathcal{F}(t_i) - \mathrm{E}(S)\right)^2$$

the third similarity measure is defined as

$$\mathrm{D}_{\sigma}(T, T', \kappa) = \frac{1}{\kappa} \sum_{k=1}^{\kappa} \mathrm{d}\left(\sqrt{\mathrm{Var}(S_k)}, \sqrt{\mathrm{Var}(S'_k)}\right) \tag{4.3}$$

Figure 4.2 illustrates the different behaviors of the three similarity measures for the position facet. For concise description, a simplifying notation of the facet $\mathcal{F}$ and the corresponding similarity measure $\mathcal{S}$ is used: $\mathcal{F}[\mathcal{S}]$. Since each of these three measures accentuates different clustering properties of a single facet and complement each other, users may select one or multiple measures for a single facet, too. Additionally, the users also may want to combine multiple facets for filtering. Therefore, the particular similarity measures D_j of each facet can be combined with user-defined weights α_j in one distance term with the dialog depicted in Figure 4.3. In detail, the total distance $\mathrm{D}_{\mathrm{total}}$ between two trajectories is the sum of the distances of each facet and similarity measure D_j (see Eq. 4.4). Distances of every similarity measure and every facet are normalized to the range between zero and one, where the decrease of the similarity according to the differences can be designed as a function f_j by the dialog shown in Figure 4.4.

$$\mathrm{D}_{\mathrm{total}} = \sum_{j} \alpha_j \, \mathrm{f}_j(\mathrm{D}_j) \tag{4.4}$$

The reason why the introduced similarity measures with their according distance functions are favored over other established trajectory matching approaches, such as *dynamic time warping* (DTW) or *longest common subsequence* (LCSS), is their

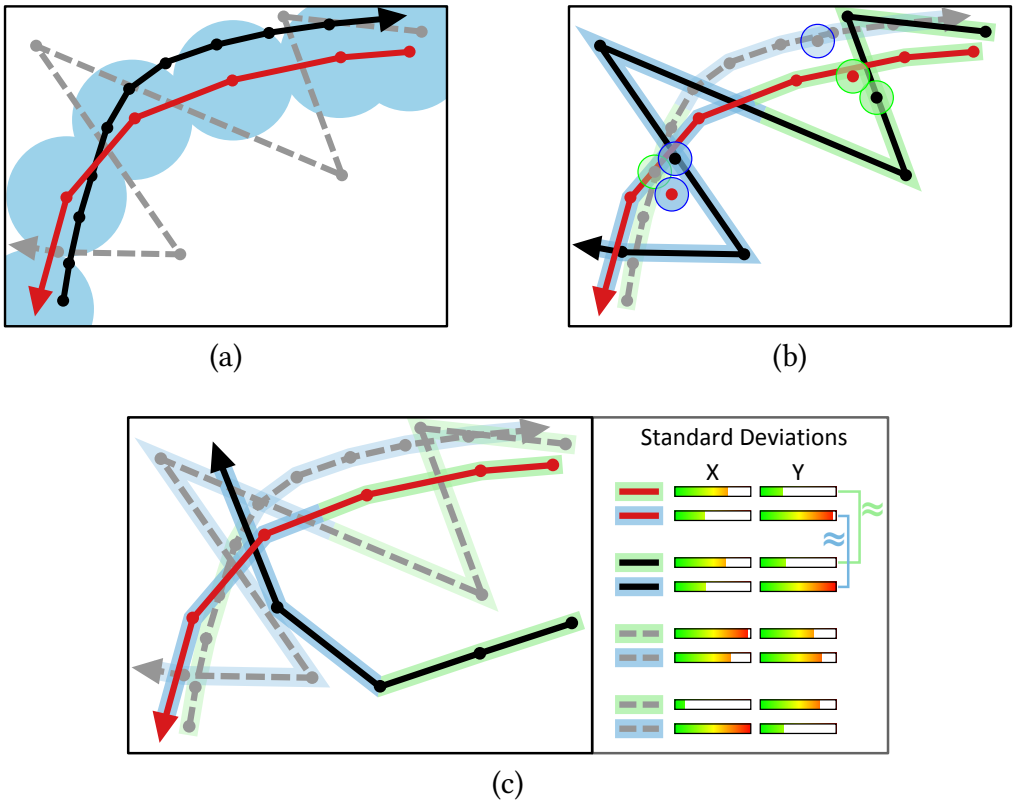


Figure 4.2 — Different effects of the three similarity measures for an example of the position facet. The dots on each trajectory show the samples points. Trajectories that are considered similar to the red trajectory with respect to the particular measure are depicted as solid lines in black, while dissimilar trajectories are shown in gray with dotted line patterns. The *coverage* measure (a) evaluates the overlap of ϵ-vicinities (blue discs—only depicted for the red trajectory) and sample points. *Distance between means* (b) calculates the sum of distances of the mean positions of the κ trajectory segments. The two-colored circles show the mean positions of a segment (outer color) of a trajectory (inner color). For the example, $\kappa = 2$ segments are used; the first depicted in green, the second in blue. The same segments are used in the example of *distance between standard deviations* (c). In this case, the differences between the standard deviations of each segment are summed up to form the distance measure. The standard deviations of each segment is illustrated on the left side of (c). Both trajectories considered in (a) and (b) are dissimilar here, and a new trajectory, which is similar according to distance between standard deviations measure, is added.

Figure 4.3 — Facet and similarity measure selection dialog. The users can select arbitrary combinations of facets and similarity measures. In the depicted setting, position[coverage], azimuth[mean], azimuth[standard deviation], and time[standard deviation] (i.e., lifetime) are selected (the velocity facet is deselected). The users choose also the number of segments (granularity) used for comparison (e.g., azimuth[mean]: 3 segments), their impact (weight) to the overall similarity (e.g., azimuth[standard deviation]: 0.2), as well as the ϵ for coverage measures (e.g., position[coverage]: $\epsilon = 1.0\,m$). Additionally, a minimum required similarity of a particular facet with its similarity measure can be applied as prerequisite (e.g., for position[coverage]: 0.8; if below, the total similarity will be 0). The edit buttons open the user interface to model the decrease of similarity function (see Figure 4.4). The dialog was developed in the context of a supervised diploma thesis by Engelhardt [89].

versatility. Dynamic matching approaches stretch the trajectory in one domain (e.g., temporal) to determine the best alignment in another domain (e.g., spatial). Since the approach uses multiple facets, a deformation of the trajectory to match one facet of a trajectory better would also change the distance of other features. If a global maximum of alignment (including all facets) would be used, each trajectory would be deformed in an unpredictable and for the user intransparent way. The coverage measure does not

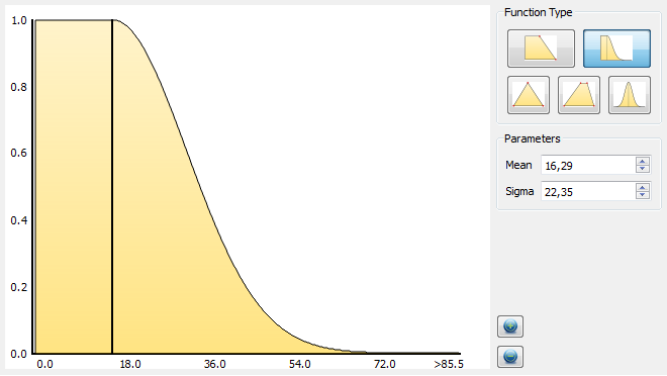


Figure 4.4 — Visual interface to design the similarity decrease function. The designed function determines the decrease of the similarity according to the chosen distance measure (i.e., D_μ or D_σ).

require any alignment at all, while the other measures align the trajectories relative to their lifetimes.

The facets supported in the system are:

Position. Position is one of the most important properties for the segmentation of a set of trajectories. The location of a trajectory at each sample point is obtained by the projection of the bottom center point of a blob's axis-aligned bounding box to the ground plane. This way, location is represented in a real-world metric rather than in perspectively distorted pixel metric. The position between two samples is linearly interpolated.

Velocity. Another facet is the velocity of trajectories on the ground plane, which is calculated as the difference between ground-plane-projected sample positions.

Azimuth. The azimuth of a trajectory represents its movement direction. This facet is useful to distinguish main paths of a set of trajectories. The azimuth of a trajectory is expressed as angle between the movement direction, calculated between two samples, and a defined reference direction. Due to the periodic behavior of the azimuth, the distance between two angles is defined as the lower angle between them (i.e., $d \leq 180°$).

Time. The time facet is relevant for filtering and clustering a set of trajectories according to their duration (lifetime) and temporal occurrence. Temporal occurrence of a trajectory is represented by the temporal mean of its samples, while the standard deviation of the samples' recording time expresses the trajectory's duration. Since

distance between means and distance between standard deviations are the only relevant similarity measures, the coverage measure is not considered in the context of the time facet. The combination of both similarity measures could be regarded as the temporal coverage of a trajectory, due to the continuity of the time dimension. Further, both similarity measures do not apply any segmentation to a trajectory, since segmentation just splits a trajectory into temporal parts. In the context of connected and temporally densely sampled trajectories from surveillance footage, such segmentation loses its purpose.

Object Class. The object class facet provides an easy way to distinguish between the types of tracked objects. The system only supports basic object classes that can be classified by an object's appearance, such as color, size, or local features. In the used datasets, three different classes are applied: persons, bicyclists, and cars. However, the object class is a categorical value, thus, none of the three similarity measures is appropriate. Object class similarity only depends on the binary distance. The binary distance is one if the classes of two trajectories are different, and zero otherwise.

4.2 Filter Definition

One contribution of this thesis is to support users in their filter definition process. Therefore, the user interface is designed to comply with the following four essential interaction guidelines:

1. Easy-to-use filter definition
2. Confidence-incorporated filter definition
3. Decision-guided filter definition
4. Filter feedback

In this section, the particular possibilities to define filters in the video visual analytics system are discussed in the context of these guidelines.

4.2.1 Easy-to-Use Filter Definition

Since filter specification is an important element in interactive exploration, it should be easy-to-use. The *International Organization for Standardization* (ISO) describes a set of usability heuristics for dialogs. One of these is the "Conformity with user expectations"[2]. This principle demands that an application should be consistent and

[2] ISO 9241-110:2006 Ergonomics of human-system interaction—Part 110: Dialogue principles; conformity with user expectations

Figure 4.5 — Filter formulation by example. All trajectories in the currently selected time windows are displayed and can be selected by a single mouse click. They can either be added to a white or black list, or be used as starting point for trajectory sketching (see Figure 4.10). The trajectory picker dialog was developed in the context of a supervised diploma thesis [89].

in accordance with the users' expectations. In our case, this implies that the filter formulation has to support the users in defining filter parameters according to their real-world associations. The input of locations, directions, distances, and so on has to be done in a manner that matches the knowledge and experience of the users with these attributes.

In particular, the location filter (see Figure 4.6) is defined by drawing in the image similar as done in applications such as Photoshop, and sample trajectories are selected in the current time window by a single mouse click (see Figure 4.5). For each filter, a keyframe is depicted in the background to convey the spatial context information to the user.

For the selection of the movement directions, it is important to put the movement azimuth in context to the image space. A compass-like visualization illustrates how directions are embedded, while a keyframe contributes scene context. The highlighted parts of the circle represent the desired and supported directions (see Figure 4.7). Hence, direct match between the movement azimuth and the scene is established. This facilitates that movement directions of desired trajectories can be easily estimated.

Interactions between moving objects can occur if they share nearby spatial locations at similar time. Thus, two attributes, the spatial distance and the temporal distance,

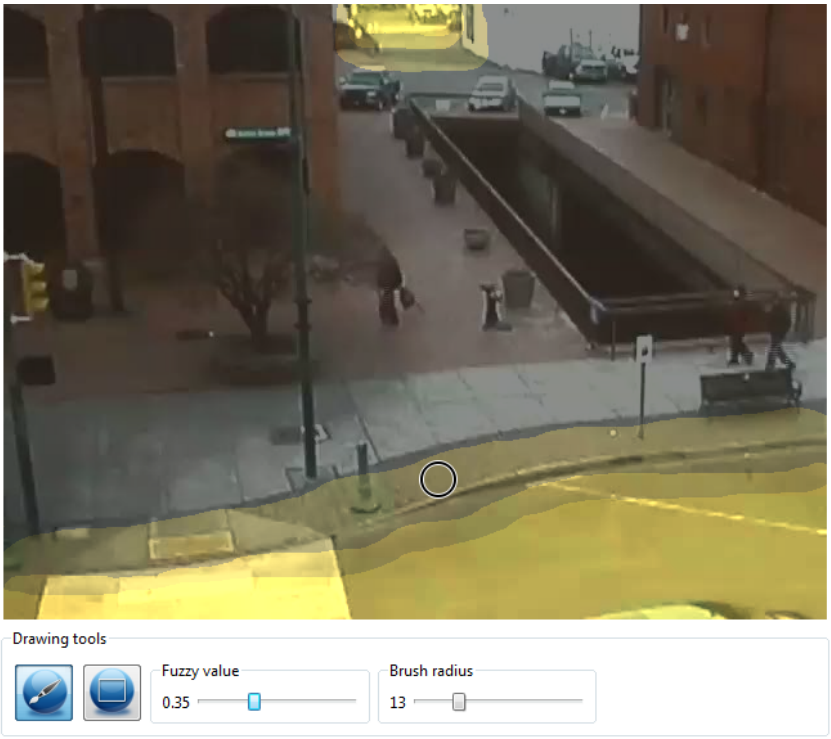


Figure 4.6 — Interface of the location filter. Spatial areas of interest (yellow) can be brushed on a keyframe similar as done in applications such as Photoshop. The fuzzy value between 0 and 1 that will be brushed is selected by the "Fuzzy value"-slider. The fuzzy value of an area of interest is depicted by saturation of the yellow color. This particular location filter can be used to include or exclude trajectories situated on streets.

are of interest for a relationship filter. Figure 4.9 shows the interface to provide the spatial distance of the relationship filter. To increase users' awareness of their defined distances at different locations, a distance circle is projected on the ground plane as the mouse hovers over the keyframe. The temporal distance of the relationship filter, as well as filters for lifetime or speed of trajectories, are fed into the system with a dialog similar to Figure 4.8.

Endert et al. [86] reported on observations of analysts working with the IEEE VAST 2009 Challenge data. One of the outcomes was that professional analysts like to start with intuitive GUIs to sketch filters and in this way explore data rapidly. Later, they prefer to enter exact values. The system supports both interaction modes: users can drag and drop values (e.g., movement azimuth and speed filter, red points in Figures 4.7 and 4.8) and brush their desired locations (location filter, Figure 4.6) to sketch filters roughly as well as enter the exact values (right panels of Figures 4.7 and 4.8).

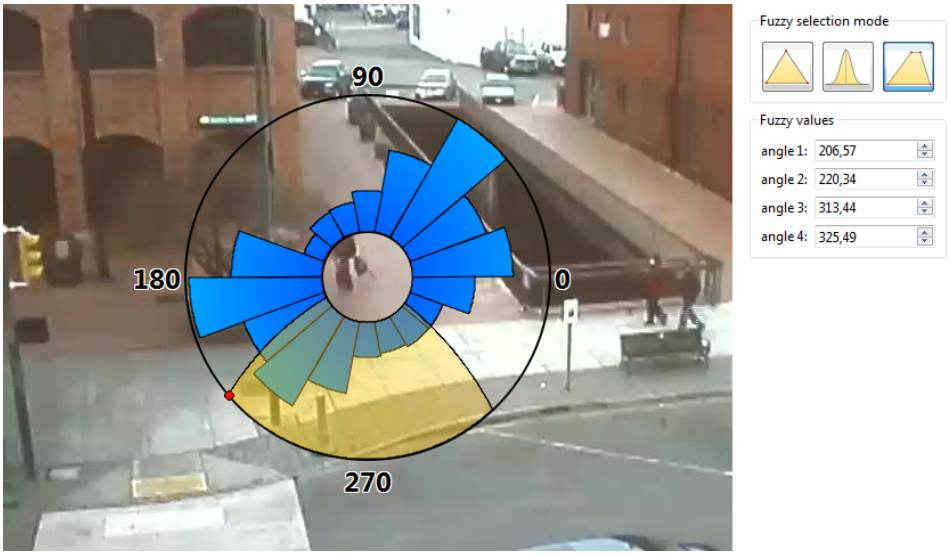


Figure 4.7 — Interface for movement azimuth selection. A keyframe is depicted for context information. The histogram (blue bins) shows the distribution of trajectories' directions. The yellow area represents the selected directions. On the right side, users can choose to model their uncertainty by different fuzzy selection modes, which is described in Section 4.2.1. This particular selection of the movement azimuth focuses on trajectories of people walking from top to bottom.

The filter definition by sketching trajectories is another implementation of this strategy (see Figure 4.10). The users can sketch a trajectory by simply adding particular positions it should traverse. Afterward, the trajectory can be adjusted in detail by defining speed and time stamp constraints of the whole trajectory or particular segments. The required level of detail depends on the choice of facets and similarity measures used to query for similar trajectories (see Figure 4.3). The system further eases up filter definition by providing automatic functions to resample the sketched trajectory according to the frame rate of the video (which is especially important for the coverage measure), as well as of trajectories that are selected by example for modification (too many sample points handicap fast sketch modifications).

Wolfson [309] introduced operators for retrieving trajectories that stand in certain relationships to a region: *always/everywhere* and *sometimes/somewhere*. The proposed location filter and the spatial distances of the relationship filter require trajectories to *start at*, *end at*, be *partially in*, or to be *completely in* a region. Wolfson uses the two discrete states to model uncertainty: *possibly* and *definitely*. In contrast, the proposed approach assigns trajectories to filters by a continuous *degree of membership* (DOM).

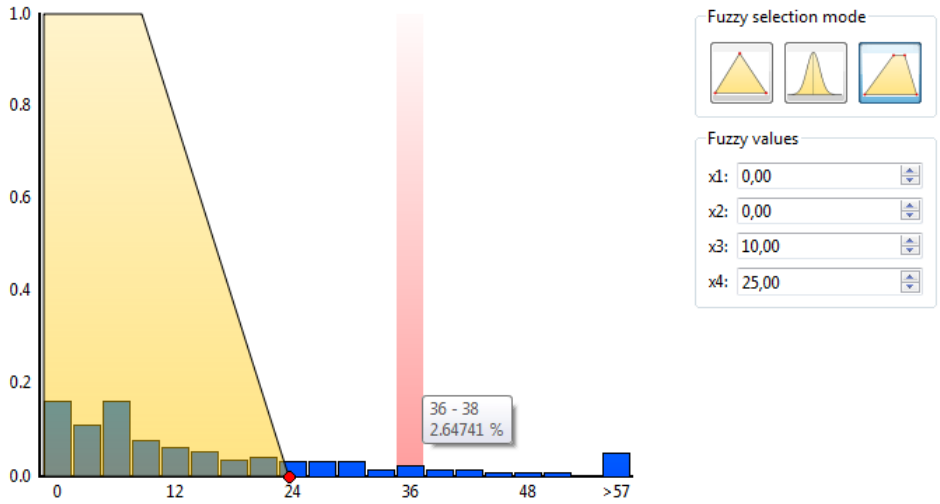


Figure 4.8 — Speed filter. A histogram (blue bins) shows the distribution of the mean speed of the trajectories. The yellow area represents the selected speeds. This particular speed filter can be used to focus on trajectories at walking speed.

4.2.2 Confidence-Incorporated Filter Definition

The human-centered and hypothesis-based filtering introduces a new level of abstraction. Since filters are associated with assumptions, they are a source of uncertainty, too. Filter definitions have to enable analysts to incorporate their confidence about hypotheses into the relevance feedback. In consequence, the relevance feedback must not strictly filter for trajectories, but has rather to cope with fuzzy decisions based on user-defined confidence values.

To address this issue, filters are represented by fuzzy sets that allow analysts to model their confidence by fuzzy membership functions. The video visual analytics system supports three different membership functions: Gaussian, triangle, and trapezoid set functions. These functions are the most common fuzzy sets functions [215], and they are modeled by only few parameters: two (Gaussian: mean and deviation), three (triangle), or four (trapezoid) (see Figures 4.7 and 4.8). For the relationship filter, a single-sided fuzzy membership function (termed *Z-function*) is used, such as the one in Figure 4.9. For the location filter (see Figure 4.6), the fuzzy values of arbitrary areas can be set separately. For the sketch filter, users can incorporate their filter definition confidence into the similarity decrease function (see Figure 4.4).

A trajectory's DOM to a property filter is calculated by the uncertainty originating from the *feature extraction* stage (data quality) and the confidence provided by the users. Therefore, the system integrates the product of an uncertain attribute of a trajectory and the fuzzy set function of a filter. The resulting interval of the DOM is between 0

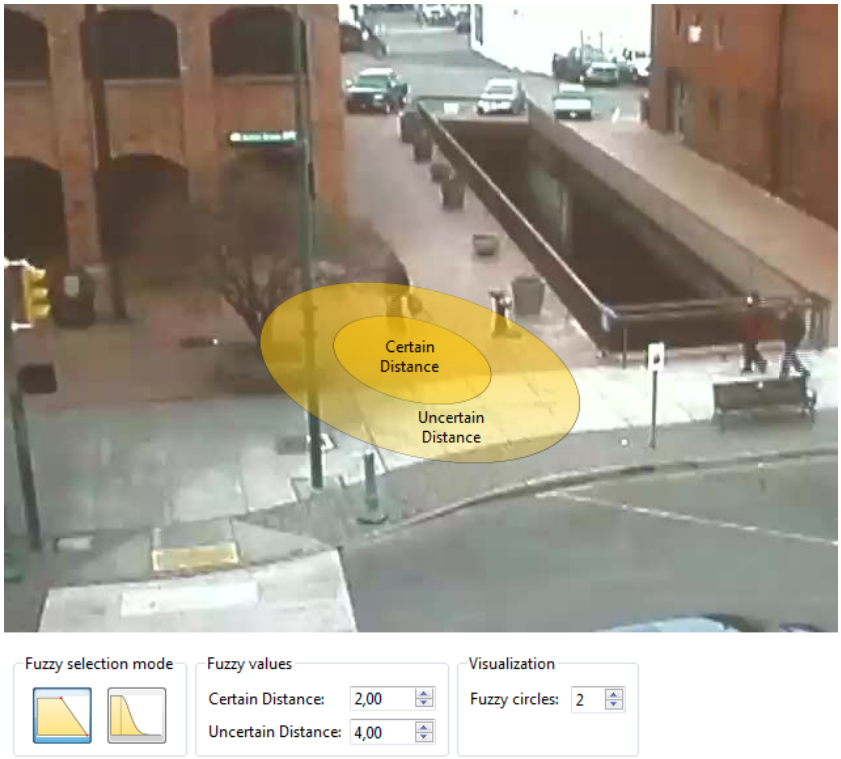


Figure 4.9 — Interface for spatial distance selection of the relationship filter. A keyframe is depicted for context information. Yellow ellipses indicate the certain and uncertain spatial distances projected on the ground plane at the mouse position helping users estimate distances. Please note that the relationship filter is independent of the absolute spatial location: only the relative distance between trajectories is considered.

and 1.

The property filters differ in the complexity and discretization in calculation of the DOM. While the location filter has to regard all three spatio-temporal dimensions, other property filters consider only a single attribute, for example the mean speed of a whole trajectory. The discretization is either introduced in the filter evaluation step (i.e., the uncertain attribute is evaluated at sample points) or is already available (e.g., pixels and frames). In contrast, the complexity and discretization of the sketch filter is determined by the user selected facets, similarity measures, and granularities.

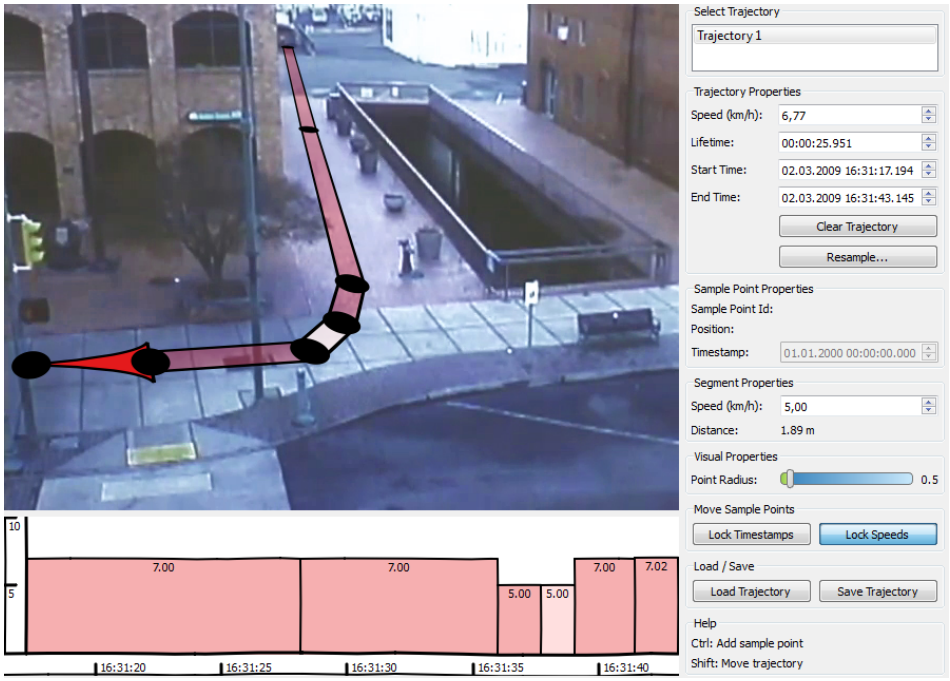


Figure 4.10 — Filter formulation by sketch. The filter interface shows a keyframe for context information and the trajectory sketches projected to the ground plane. A trajectory is sketched by adding sample positions (black dots). Afterward, the sample positions and segments can be further modified by drag and drop (position), or by configuring time and speed constraints. The bar chart at the bottom visualizes the speed of the particular segments and the temporal context. The facets used for filtering are specified in a separate dialog (see Figure 4.3). The sketch filter dialog was developed in the context of a supervised diploma thesis [89].

4.2.3 Decision-Guided Filter Definition

The research agenda of visual analytics identified the importance of "visual analytics systems to support the analyst in executing sound analytic technique routinely, facilitating insight and sound judgment in time-pressured environments and compensating for inexperience wherever possible" [282].

To support the analysts, the system supplies background information for decision guidance and situational awareness by presenting context-sensitive graphical statistics. Context sensitivity denotes the selection of the provided statistics according to the filters. For different filters and when appropriate, the system depicts normalized histograms of the filters' attributes. Due to the streaming structure of the video visual analytics pipeline, the histograms are updated continuously and show only the statistics of the features that have already passed the filters.

To conform to users' expectations, the appearance of the histogram is adapted to the filter. For example, the histogram of the movement azimuth filter is arranged in a circle in order to ease derivation of assumptions of trajectories' behavior. Investigating the histogram in Figure 4.7, one assumption may be that the people walking on the footway from right to left outnumbers the people walking into the opposite direction. This supports users to define relevance, derive hypotheses, and to create appropriate filters.

Further, such context-sensitive graphical statistics help reject previously made incorrect assumptions, for example that there are two predominant speed intervals: one for people and one for cars. The histogram of the speed filter shown in Figure 4.8 tells us that there are no sharp speed intervals. This distribution arises from cars decelerating and stopping at the signal light and bicycles at intermediate speed. Therefore, the assumption has to be rejected.

As already mentioned above, the approach complements the context-sensitive graphical statistics with keyframes for spatial context information where appropriate (see Figures 4.5, 4.6, 4.7, 4.9, and 4.10).

4.2.4 Filter Feedback

Feedback is essential for users to verify their hypotheses by filter definitions. Without filter feedback, users cannot know whether their filter formulations are too weak or too restrictive. More than that, users are left in the dark whether the filters operate as intended. They are also not aware if the filters cause side effects. Therefore, the video visual analytics system provides filter feedback of three different kinds.

First, due to the ability of fuzzy filters to forward the DOM of trajectories to the relevance measure stage, these DOMs can be mapped to visual properties of the visualizations (see Chapter 5), such as to the color of the trajectory tubes in the *VideoPerpetuoGram* (VPG) (see Chapter 6.4). Moreover, the defuzzification threshold of fuzzy filter containers can be lowered during filter definition to keep them "alive" in order to receive filter feedback.

The second kind of filter feedback is also provided by the visualization stage. Various visualizations support the users with global filter feedback, such the amount of remaining trajectories in the selected static time window (e.g., the ISS, see Chapter 6.3), or aggregated information of them (e.g., ISS and *chart view*). Hence, users get a feeling how restrictive their filter formulations are.

The third filter feedback shows for a selected trajectory the DOM to each single filter and container.

In the next section, a quite different approach to the filter definitions mentioned so far will be discussed: the ad-hoc training of specialized classifiers.

4.3 Ad-hoc Training of Classifiers

When analyzing complex and high-dimensional data spaces, appropriate model parameters are unknown and filter definition by a single example does not capture the large variety of appearances. In such cases, query by multiple examples can be useful, which can be seen as the training of a complex classifier. In this way, users can specify what they seek by integrating machine learning techniques into information visualization, as commonly recommended (e.g., by Shneiderman [263] or Chen [54]). However, in contrast to pre-trained classifiers as they are widely used in video analytics (e.g., person or car detectors, included in many video management systems), classifiers used to define filters within the visual analytics process have to be trained ad-hoc.

Let us consider the example of video surveillance operators who assume, after some initial analysis of video sequences, that a cyclist might have been involved in the case of a traffic incident they are investigating. Hence, they want to extract cyclists from video data to reduce the amount of video and to focus on promising parts for hypothesis verification. This example illustrates the need for training of new classifiers that can also be highly complex and specialized (e.g., hand-waving bicyclists with red helmets may be important in our example scenario). Since pre-trained instances of such classifiers are generally not available, the analysts have to define the filter by themselves. However, feature selection and model parameter definition for objects such as a bicyclist are too complex to be manually defined, even for domain experts with support by interactive visualization [298]. Hence, filter definition via query by examples may be a viable solution.

In contrast to traditional supervised training of a classifier, ad-hoc training involves new challenges. One of the challenges is to limit the *annotation costs* with respect to the large amount of annotated data is often required for proper training of a classifier. The *annotation quality* is also essential for training. This includes the precision of the sample selection (e.g., the precision of bounding boxes around objects of interest), the distribution of the samples (temporal and spatial, especially critical when suitable examples are rare), or the vagueness of the query idea the users have in mind (e.g., is a person on a trike also of interest?). Another challenge is the *classifier quality assessment* to determine an appropriate moment to stop training of a classifier. This is important to reduce the time spent for annotation and to receive a generalized classifier that does not overfit.

Inter-active learning, an extension to conventional active learning that directly involves human experts in the ad-hoc training process using the visual analytics methodology, addresses these challenges. The additional interaction introduced by inter-active learning is depicted by red arrows in Figure 4.11: the goal of this process is to create filters efficiently by leveraging the complementary strengths of human and machine, as outlined by Bertini and Lalanne [27] in the context of the knowledge discovery process.

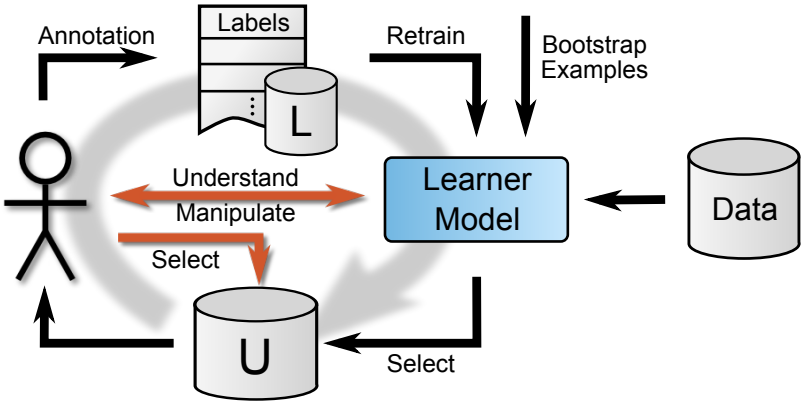


Figure 4.11 — In active learning, a (potentially bootstrapped) learner iteratively refines itself by posing queries from a pool of unlabeled data U to a (human) oracle that provides labels for the data L. Inter-active learning is an extension (red arrows) of active learning that further allows the human experts to integrate their background knowledge directly into the model.

In detail, inter-active learning efficiently approaches the goal of a well-trained classifier by iterating over the three basic steps: (i) assessment of the performance of the classifier, (ii) annotation of data instances and/or manipulation of the classifier model, and (iii) retraining of the classifier. Due to the tight connection between learner and user, visualization and human-computer interaction are, besides automatic methods, the central aspects of inter-active learning. Figure 4.12 illustrates this connection between the three major elements: learner model, user, and data. Furthermore, Figure 4.12 depicts the flow of information between the three elements by visualization, interaction, and automatic methods. The iterative interaction of these three components results in a visual analytics process that aims to refine the classifier model and its comprehension by the users.

The first two steps (i, ii), which involve the users, can further be divided into tasks the users may consider to process each cycle. For step (i), these tasks include assessing the success of the last training cycle (training feedback) and determining if a stopping criterion was reached (e.g., the model has already reached an appropriate level of quality or training does not improve the model anymore). Furthermore, by assessing the model's performance, users can build trust in their trained model and learn to know its strengths and weaknesses. Hence, they can incorporate the performance and uncertainty of "their" filters into their decisions within the analytical reasoning process. Finally, quality assessment also guides the users in refining the model. Users may detect overfitting of the model, too broad generalization, or low robustness to noise. These issues are tackled in the second step of the cycle.

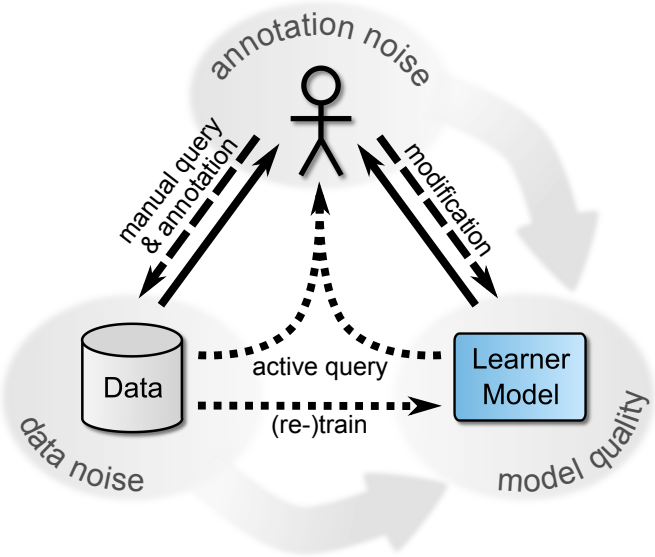


Figure 4.12 — Major components and information flow involved in the inter-active learning process. Solid lines depict information conveyed to the users by visualization, dashed lines represent user interaction, and dotted lines illustrate the flow of information triggered by automatic methods. Furthermore, the contribution of noise-affected data and labels to the model's uncertainty is depicted.

After the classifier model was analyzed in the first step, two ways to refine the model are available to the users in the second step (ii): data annotation and direct model manipulation. Both can be used to broaden the classifier model to accept a wider variety of data instances or to narrow the acceptance range. However, using direct model manipulation is recommended for generalization purposes only, and, in contrast, data annotation is suitable for both tasks. This recommendation accounts for the complex dependencies of high-dimensional data distributions. In such cases, it is often easier to tell the system what is wrong (e.g., overfitting of the model) than to define what is right. For data labeling, the users can choose which data regions they intend to annotate for model refinement. In this way, the users can efficiently integrate novel domain knowledge into the system. Labeling of data in regions near the decision boundaries helps increase the classifiers confidence, whereas labeling of data in regions far away from the decision boundary helps explore new regions of the data space and might reduce extensive class confusion. However, users can also rely on the classifier model to provide the most beneficial data instances for labeling utilizing active learning.

The proposed approach uses a *cascade of classifiers*, where each node consists of a committee of weak classifiers (boosted rectangle features), to predict class assignments of sliding windows in each video frame. Figure 4.13 depicts a typical workspace of the inter-active learning system that consists of several coordinated views. Left, the cascaded scatterplot (see Figure 4.13 (a)) shows the evaluation results of the cascade for performance assessment. In the cascaded scatterplot, the abscissa is divided into as many parts as number of nodes exist in the cascade, and each part corresponds to one node. In those parts, all data points are depicted that are not rejected until the associated node of the cascade. Green dots represent data points that pass the whole

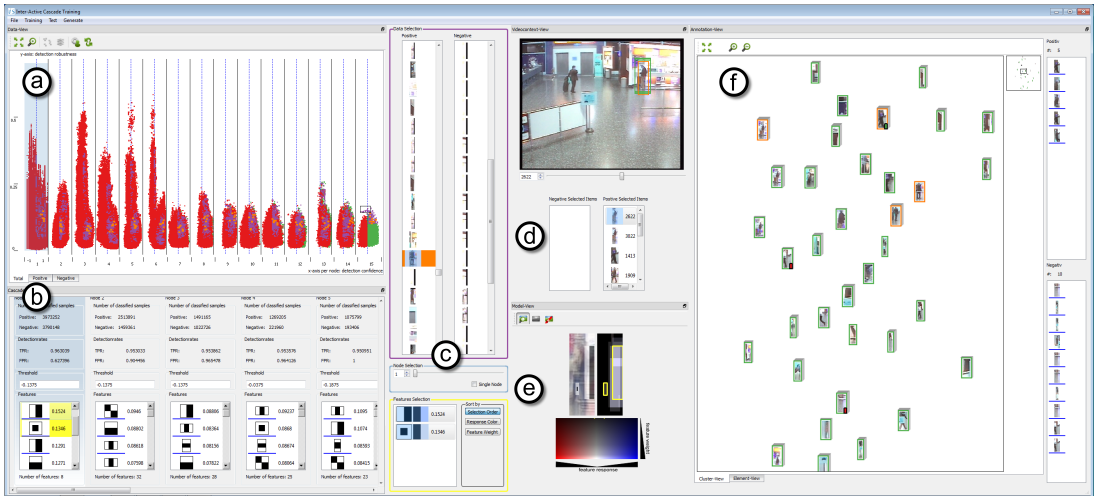


Figure 4.13 — Workspace of the inter-active learning system after learning with a couple of training examples: (a) cascaded scatterplot, (b) cascade information, (c) selection interface, (d) video context view, (e) visualization of classifier model, (f) annotation view.

cascade, red points are rejected. The x-location (inside the cascaded scatterplot part) depicts the classifier's confidence and the location on the y-axis its robustness. The blue dashed line shows the decision boundary, where data points close to the boundary are least confident. The users can zoom into the visualization to receive details and select data points for further investigation. Below the cascaded scatterplot, cascade information of the trained classifier model is shown (see Figure 4.13 (b)). Here, users can determine information about the training, such as the number of classified samples and the detection rates, and about the cascade, such as the features applied with their weights, and the nodes' thresholds. Moreover, direct model manipulation is supported by adding, removing, or modifying features and their properties. A selection of features can be visualized (see Figure 4.13 (e)) in context of a selection of data samples (see Figure 4.13 (c)) to judge feature responses. For fast labeling, the annotation view (see Figure 4.13 (f)) clusters visual similar samples together. Selected data instances in the annotation view are shown in their video context (see Figure 4.13 (d)).

Summarizing, the inter-active learning approach with the presented coordinated views includes the users' expertise for labeling and supports them in assessing the classifier model by visualization. Besides the annotation of manually or automatically selected data instances, users are empowered to adjust complex classifier models directly. The model visualization facilitates the detection and correction of inconsistencies between the classifier model trained by examples and the user's mental model of the class definition. Visual feedback of the training process helps the users assess the

performance of the classifier and, thus, build up trust in the filter created.

A detailed discussion of the inter-active learning approach for ad-hoc classifiers, including theoretical aspects, a detailed description of the views, and a usage scenario can be found in Höferlin et al. [130].

Chapter

5 Relevance Measure[1]

The idea behind the *relevance measure* stage is to evaluate the importance of data elements (e.g., video frames, trajectories) automatically according to a user-defined model. In contrast to filters, a relevance measure does not exclude data from further analysis. The assigned relevances can be utilized in the subsequent visual representation of the data elements to guide the analysts' attention to important areas in the data. The particular visualizations can also decide to hide irrelevant information completely from the users. This addresses perceptual scalability and facilitates situational awareness during analysis. Detailed description of the different mappings of calculated relevance and visual representation is provided in the next chapter. In addition, the relevances can be mapped to playback speed, for example, for *adaptive fast-forward* (see Chapter 5.1).

To define a proper relevance model, users can select different *relevance measures* and connect them as a relevance graph similar to the definition of filters (see Figure 4.1). The video visual analytics system supports two types of elementary relevance measures. The first type contains relevance ratings that are directly derived from the data elements' DOM to a fuzzy filter. The others include their own relevance model parameterized by the users.

Similar to filters, relevance measures are defined for particular data types, such as video frames or trajectories. However, relevances can be based on DOMs from arbitrary data types. For example, a relevance measure for the data type video was defined for the participation at the IEEE VAST Challenge 2009 that utilized the filter results of trajectories in the following manner: video frames receive a binary relevance according to the existence of trajectories with a DOM above a threshold. This means, video

[1] Based on Höferlin et al. [132, 137], and Höferlin et al. [127, 131] (Chapter 5.1).

frames that include trajectories that passed the filters, receive the relevance 1, and 0 otherwise. For the challenge participation, this relevance was mapped to both the playback speed of the video and to visual mappings in the visualization (i.e., to the VPG, see blue bar in Figure 6.30).

The next section discusses the objective of adaptive fast-forward for video sequences and briefly introduces two own methods to rate the relevance of video frames for this purpose.

5.1 Relevance Measures for Adaptive Fast-Forward

Fast-forwarding video can be considered as a temporal aggregation or summarization applied to raw video. From a perspective of video analysis, it readily fits in visual exploration strategies such as the visual information-seeking mantra [262] or the visual analytics mantra [158]:

- Video fast-forward can be employed in bottom-up processes, for example, in early stages of information foraging and filtering [236]. Here, this early-stage abstraction of the video data serves as a means of accelerating the process of video inspection for the user. In this context, fast-forward is used to obtain a first impression of the video footage quickly.
- Additionally, it can be used for top-down exploration. Top-down exploration typically starts at a highly aggregated level of visual representation but involves drill-down to details of the original data after possibly several iterations of zooming, filtering, and browsing. Thus, fast-forward video playback is the stage of data drill-down right before the raw video material is watched.

A typical property of unedited video material, such as surveillance footage, is the nonuniform distribution of activity: busy periods alternate with idle periods. Since *conventional fast-forward* plays the whole video at constant pace, users are overburden during busy periods and bored during periods with no activity (see Figure 5.1 (top)). Moreover, the users are kept busy by manually rewinding and adapting the video playback speed. A solution to alleviate this problem is *adaptive fast-forward*, which adapts the video playback speed automatically according to the relevance of each frame (see Figure 5.1 (bottom)). Consequently, annoying or irrelevant parts (e.g., static parts) of the video are reduced while emphasis is put on relevant parts (e.g., crowded parts).

The target footage of the proposed adaptive fast-forward measures is unedited video material without audio tracks. Common movies or TV broadcasts usually do not satisfy these criteria, since they are edited to condense the content appropriately, for example to narrate a story. Hence, watching such video data using the proposed adaptive fast-forward mode is neither recommended nor suitable, although it is possible.

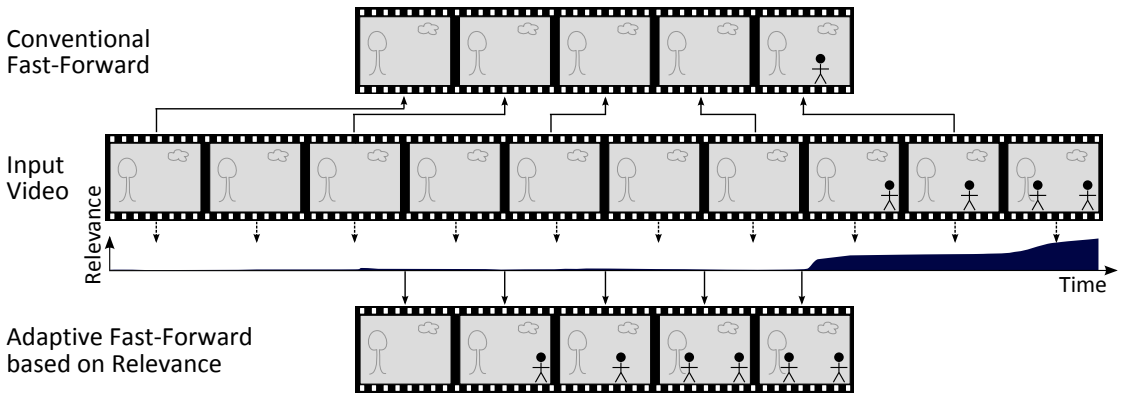


Figure 5.1 — Difference between traditional cue-play (top) and adaptive fast-forward (bottom). Both sequences are scaled to half the duration of the input sequence (mid). In adaptive fast-forward, the playback speed is adjusted according to a relevance measure while the conventional approach samples the sequence at a constant rate.

This gives rise to the question which relevance measures are appropriate for adaptive fast-forward. Peker et al. proposed an adaptive video fast-forward technique that adapts the playback speed of the video sequence relative to the present motion [228] and the visual complexity [227] (as combination of the spatial complexity and the motion) of the scene. An adaptive playback speed based on similarity to a target clip is described by Petrovic et al. [231]. One example application they propose for this type of adaptive video playback is a baseball game. The users feed the system with a target clip of the game. Scenes in which the game continuous are then displayed in normal speed, while scenes of game interruptions (e.g., showing spectators) are highly accelerated. Cheng et al. [57] designed an adaptive video player called SmartPlayer that adjusts the playback speed according to three factors: motion, manually defined semantic rules, and former playback preferences of the user.

The video visual analytics system assesses relevance according to motion activity level as proposed by Peker and Divakaran [227]. One drawback of this approach is that *static changes* (i.e., scene changes uncorrelated with any motion, such as blinking lights), which are common in surveillance video due to time lapse (see Figure 5.2), cannot be sufficiently handled. This measure also has difficulties with *video noise*, for instance, because of coding artifacts, or as consequence of sensor noise in dark environments due to high gain settings.

Therefore, two own methods to rate relevance in video frames are introduced: an information-based relevance measure [127], and a relevance measure that is based on a learned visual attention model [131]. The next sections briefly outline the two approaches.

Figure 5.2 — Three subsequent frames of a temporally subsampled surveillance sequence with arbitrary optical flow vectors due to static changes.

5.1.1 Information-Based Relevance Measure

The information-based relevance measure calculates the information gain between two successive video frames based on Shannon's information theory. In this approach, temporal information of a video sequence is formulated as symmetrized Rényi divergence between the temporal noise distribution and the frame difference distribution. The proposed approach is able to handle static changes and video noise in contrast to approaches that utilize motion [227].

The motivation for this approach is that the assumption that subsequent frames F_1 and F_2 remain constant (i.e., $|F_1 - F_2| = 0$) in a scene where no changes appear does not hold for real video data. The sensor as well as the encoding process introduce noise to the signal. Therefore, video data is modeled as additive combination of the signal S carrying the actual information and the temporal noise N. Hence, the absolute frame difference D is:

$$D = |F_1 - F_2| = |\Delta S + \Delta N| \tag{5.1}$$

With this prerequisite, the approach defines the information gain (symmetrized α-divergence) between two successive frames as

$$\mathcal{D}_\alpha(D||N) = \frac{1}{2}\hat{\mathcal{D}}_\alpha(D||M) + \frac{1}{2}\hat{\mathcal{D}}_\alpha(N||M) \tag{5.2}$$

with $M = \frac{1}{2}(D + N)$ and the α-divergence $\hat{\mathcal{D}}_\alpha$ (a generalization of the Kullback-Leibler divergence [176] proposed by Rényi [250] for generalized probability density functions) defined as

$$\hat{\mathcal{D}}_\alpha(D||N) = \frac{1}{\alpha - 1}\log_2\left(\sum_i \frac{p(d_i)^\alpha}{p(n_i)^{\alpha-1}}\right) \tag{5.3}$$

of order α for $\alpha > 0$ and $\alpha \neq 1$. Here, $p(d)$ is the discrete difference image distribution, and $p(n)$ denotes the estimated noise probability distribution. The index i indicates a particular bin of the distribution histogram. Rényi describes the measure $\hat{\mathcal{D}}_\alpha$ as "the

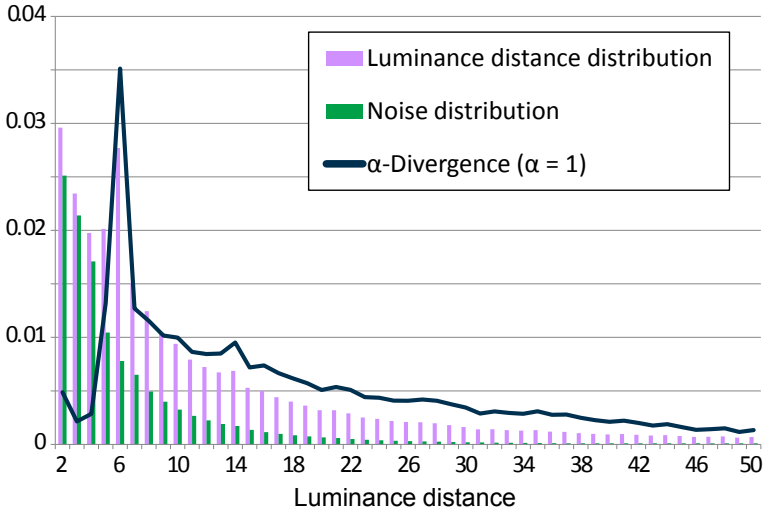


Figure 5.3 — Terms of Rényi divergence (α-divergence) between the noise distribution and absolute frame difference distribution. The first two bins as well as the last 200 bins were omitted.

information of order α obtained if the distribution of N is replaced by the distribution of D" [250]. The behavior of the Rényi divergence according to a noise distribution and an absolute frame difference distribution is depicted in Figure 5.3.

The main advantages of the proposed formulation of information gain are its robustness against noise, suitability for low frame rates, and its high computational efficiency. Moreover, users have additional control over the accentuation of this information measure by the parameter α that emphasizes certain parts of the distribution ratios.

Detailed information about the approach, including its derivation, theoretical aspects, information about the temporal noise estimation, and a comprehensive evaluation including comparable results of the proposed method with state-of-the-art approaches on different videos and a qualitative user study can be found in Höferlin et al. [127].

5.1.2 Relevance Measure Based on a Learned Visual Attention Model

The focus of visual attention is guided by salient signals in the peripheral field of view (bottom-up) as well as by the relevance feedback of a semantic model (top-down) [308]. As a result, humans are able to evaluate new situations very fast, with only a view numbers of fixations.

The idea of this relevance measure is to train a *visual attention model* from fixation data captured by an eye-tracker. Based on this model, parts of surveillance videos that

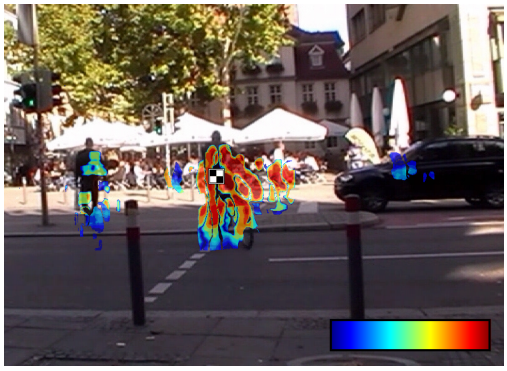


Figure 5.4 — Saliency map calculated by the presented approach. Salient regions are illustrated by a color-coded overlay from blue (low saliency) to red (high saliency). The predicted fixation regions are compared to a real fixation (black/white box) recorded by an eye-tracker.

are likely to attract visual attention are predicted. The relevance of a frame is thus defined as the visual saliency: frames with only few potential fixation points that can be surveyed fast are rated less relevant than frames with many. Example prediction results of the trained visual attention model on a single frame are depicted in Figure 5.4.

Based on the video footage and recorded fixation data, a discriminative visual attention model is created that consists of a cascade of classifiers. Figure 5.5 depicts the basic workflow of training and application of the visual attention model. Each classifier consists of a set of rectangle features selected by AdaBoost [292]. The cascade of boosted rectangle features became very popular for object detection, after it was successfully applied to face detection by Viola and Jones [292]. In particular, this approach is known for its fast computation utilizing an acceleration structure called *integral image* in combination with a cascade of classifiers with gradually increasing complexity. Classifiers at the beginning of the cascade are kept simple. Their goal is to reduce inexpensively the large amount of sliding windows that do not contain the searched object category, while keeping all windows with potential detections for the subsequent, more complex classifiers.

To obtain the examples required to train the visual attention model, the approach relies on fixation data from eye-tracking. A *Tobii T60 XL* eye-tracker was used to record overt visual attention when free-viewing different stimuli (i.e., without a specific task). The training and test videos show outdoor environments at daytime and with continuous activity of pedestrians and/or cars, which are typical for video surveillance.

The relevance measure utilizes the learned visual attention model to calculate the area covered by potential fixation points as measure of a frame's relevance.

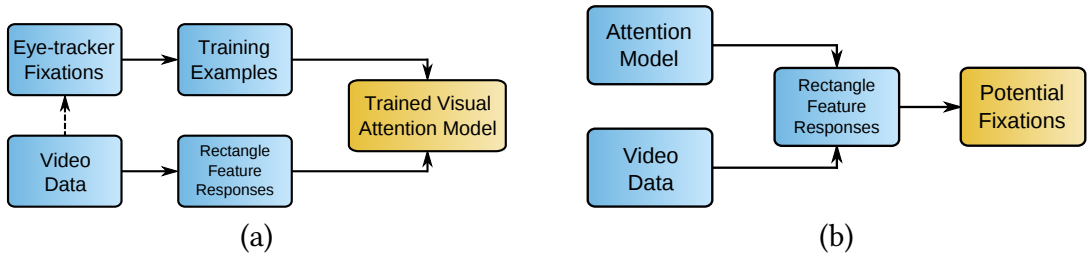


Figure 5.5 — Schematic workflow of the training (a) and application (b) of the proposed visual attention model. Arrows with solid lines show the workflow; the dashed line depicts the dependency between video and eye-tracking data.

Details about the video stimuli, the probands, and further preparation of the data are provided in Höferlin et al. [131]. Additionally, details about the training, the applied features, and the performance of particular feature combinations can be found there.

5.1.3 Comparison of the Supported Relevance Measures for Adaptive Fast-Forward

This section provides a short discussion and summarization of the performances of the three relevance measures supported by the video visual analytics system in the context of adaptive fast-forward: motion activity, Rényi divergence, and the visual attention model.

The training dataset for the visual attention model consisted of four videos with different resolutions, durations, and encodings. Additionally, perspective and captured objects vary from video to video. Hence, this experiment also indicates that the playback speed adaption using the presented visual attention model is to some extent insensitive to a specific training dataset. To improve robustness, fixations recorded from multiple subjects were used. The ratio of positive to negative examples was chosen 3:4, since experiments indicated slightly improved performance when more negative examples are used than positive examples (details provided in [131]).

The relevance feedback of the three methods was calculated on four video clips, which are different to the video clips for training of the visual attention model. These videos were also used in the user study of Höferlin et al. [127]. Three of the videos, termed *Crowded Airport*, *Airport*, and *Noisy Airport*, originate from the i-LIDS multi-camera tracking scenario. They are encoded with the Motion JPEG Video (MJPA) codec at a resolution of 720 × 576 px, and 25 fps. The *Noisy Airport* sequence is a version of the *Airport* sequence with added Gaussian noise. The *Night* sequence is an uncompressed monochrome video that was captured at night with a resolution of 656 × 494 px and

Crowded Airport *Night* *Airport* *Noisy Airport*

Figure 5.6 — Example frames of the video sequences used for adaptive fast-forward experiments.

15 fps. The sequence includes regions with low contrast and dominant noise from high gain settings. Example frames of the videos are depicted in Figure 5.6.

The results of the three measures can be roughly summarized as follows:

- Motion activity, Rényi divergence, and the visual attention model perform well on the *Crowded Airport* and *Airport* sequences.
- Motion activity fails to adapt the playback velocity of the *Noisy Airport* sequence, whereas Rényi divergence and the visual attention model can cope with noise.
- Motion activity and Rényi divergence are unable to adapt the playback speed of the *Night* sequence due to noise (motion activity) and low contrast (Rényi divergence), the visual attention model is also appropriate for this scenario.

The results of the relevance measures applied to the four scenarios are depicted in Figures 5.7 and 5.8.

Especially, the performance of the visual attention model in periods of no activity is remarkable. In these periods, the baseline is consistently located close to zero relevance, as it is expected. In contrast to that, the other methods assign some amount of importance to these periods and especially the Rényi divergence jitters strongly around its baseline.

The Rényi divergence is robust to noise to a certain degree. Nevertheless, the comparison of the relevance feedback of *Airport* (Figure 5.7 (center)) and *Noisy Airport* (Figure 5.7 (bottom)) indicates that the learned visual attention model preserves the relevance signal better under the influence of noise. For scenes with noise, motion activity is not an appropriate measure. The Rényi divergence has problems with scenarios featuring low contrast. Motion activity is not appropriate for time lapse video due to static changes (see Figure 5.2 and [127]).

The visual attention model is the only method that can cope with the high noise and low contrast scenario posed by the *Night* sequence. Figure 5.8 points out that only the

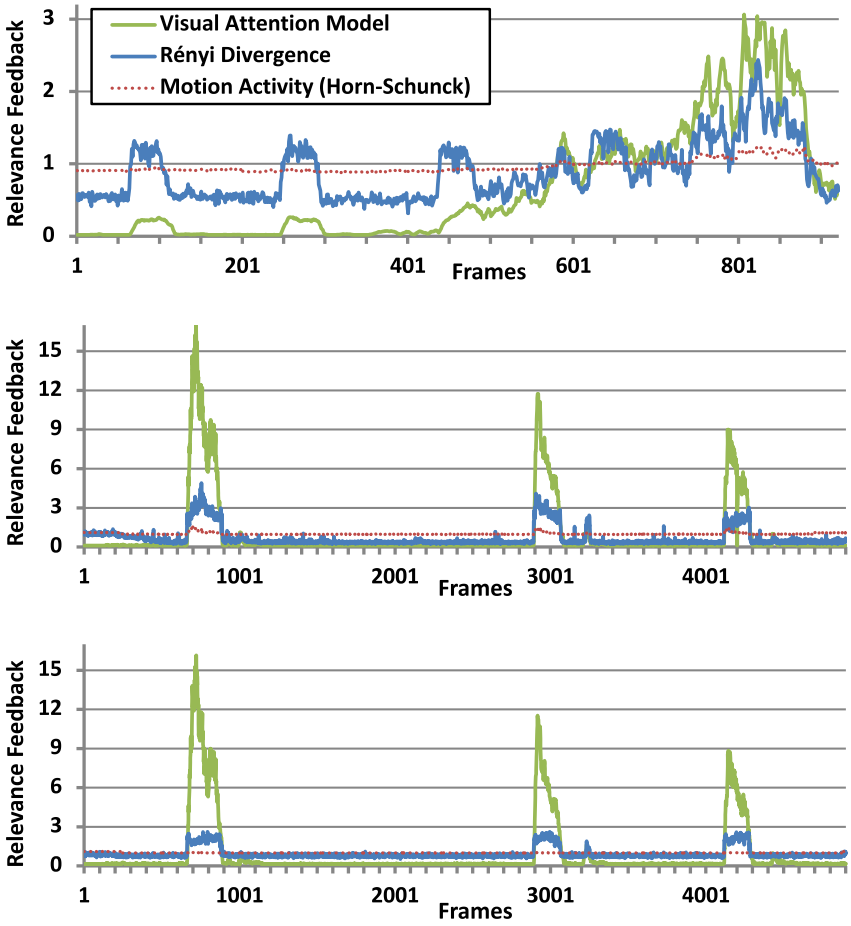


Figure 5.7 — Relevance feedback of the compared methods (normalized to an expectation value of 1, i.e., the playback time of an accelerated sequence is the same for all methods, only acceleration of particular periods vary) for the sequences: *Crowded Airport* (top), *Airport* (center), and *Noisy Airport* (bottom).

visual attention model provides the expected result: high relevance at periods where people are present in the scene. Sample videos that show a direct comparison of the different methods are available on the project's homepage[2].

Further evaluation results and a qualitative user evaluation are provided in the original publications [127, 131].

[2] www.vis.uni-stuttgart.de/index.php?id=vva

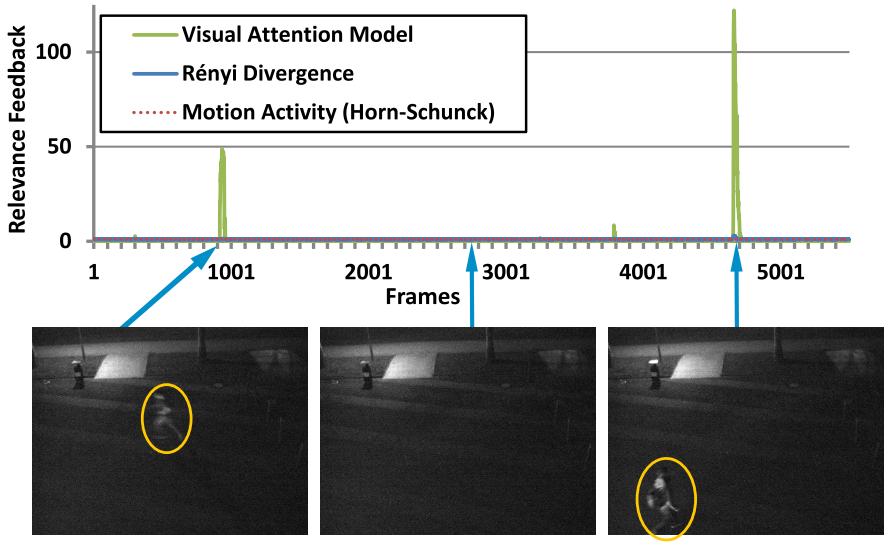


Figure 5.8 — Relevance feedback of the compared methods (normalized to an expectation value of 1) for the *Night* sequence. The visual attention model is the only approach that identifies relevant movement in this sequence with high noise and low contrast.

Chapter

6 Visualization[1]

The *visualization* stage is responsible for preparing and communicating information it to the *human analyst*. *Multiple coordinated views* provide complementary perspectives to particular facets of the data and enable the analysts to explore the dataset [251]. The analysts can select and combine these views according to their specific task requirements. Data streams of all views are temporally synchronized and can be controlled by the users via the *timeline* (see Figure 6.1 (a)). Selected data instances are highlighted in all views (brushing and linking). Further details about the selected data instances are available by the selection manager, which also allows transferring these instances as *relevant information* artifacts into the *reasoning sandbox*. Moreover, each visualization allows exporting specific views on the data as *pattern artifacts* into the *reasoning sandbox* for further sense making.

Data dimensions. Video data is naturally represented in spatio-temporal dimensions. However, it depends on the task which data dimensions are important for analysis. When searching for instances of cause and effect, the temporal dimension becomes more relevant than spatial dimensions. Other tasks, however, consider the spatial dimensions (e.g., detection of access to a forbidden area), additional properties (such as movement velocity or assignment to an object class), or a combination of them (e.g., spatio-temporal dimensions to detect encounter of multiple entities). To achieve task scalability and to support the users in exploratory pattern discovery, it is therefore important to provide different views that show various perspectives of the data.

[1] Based on Höferlin et al. [132], Borgo et al. [31, 32] (Chapter 6.1), Höferlin et al. [127, 139] (Chapter 6.2), Höferlin et al. [138, 140] (Chapter 6.3), Höferlin et al. [137] (Chapter 6.4), Höferlin et al. [136] (Chapter 6.5), Burch et al. [46] (Chapter 6.6), and Höferlin et al. [126, 128] (Chapter 6.7).

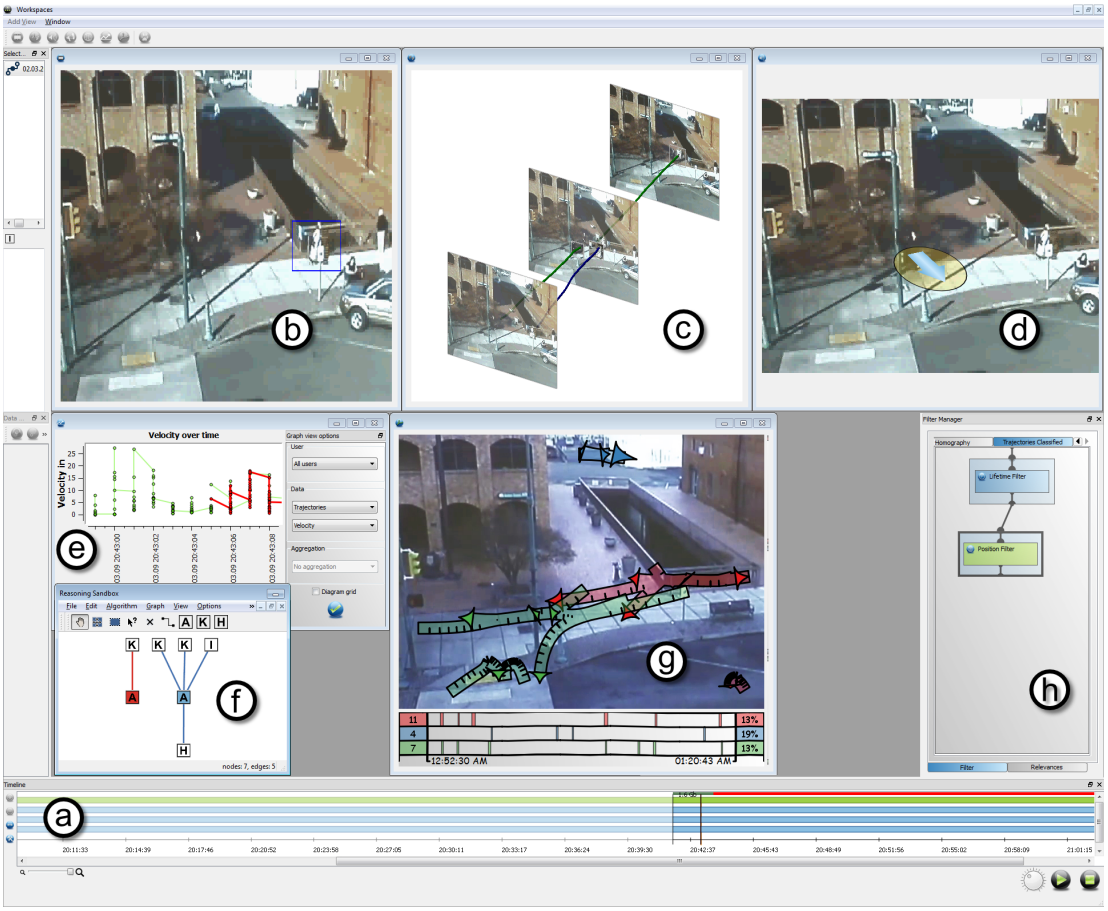


Figure 6.1 — Screenshot of the video visual analytics system that shows several views on the data: (a) timeline; (b) video player; (c) VideoPerpetuoGram (VPG) (Chapter 6.4); (d) auditory display (Chapter 6.7); (e) chart view; (g) interactive schematic summaries (ISS) (Chapter 6.3). The reasoning sandbox is depicted in (f) (Chapter 7), and the filter and relevance graphs are shown in (h) (Chapter 4). Large screenshots of most of the particular views are depicted in the corresponding chapters.

The proposed video visual analytics system provides complementary visualizations of different data dimensions.

The *timeline* view (see Figure 6.1 (a)), for example, illustrates the temporal context of different data streams by depicting their sampling intervals as bars. Additionally, the intervals of selected data instances, such as trajectories, are highlighted. Such visualization is useful to show both the temporal "location and duration of intervals. One can also see how intervals are related to each other" [7].

Object trajectories extracted from geo-referenced video sources or directly captured by

GPS devices can be displayed by the *map view* [129]. The representation in geographical context helps identify patterns that span over a large area and possibly over multiple cameras. This view represents a connection between geospatial and video visual analytics.

Data mining and data aggregation. Depending on the analysis task and the availability of exact search target definitions, the analysis process features more or less exploratory characteristics. If the task is, for example, the search for a vaguely specified search target in offline analysis, a first goal may be to gain an overview of the data. This may be achieved by building a mental model of typical pattern and structure in the data using data exploration techniques. The proposed video visual analytics system therefore features tight integration of visual data exploration and automatic data mining. Different methods in data mining can be distinguished: classification, regression, clustering, summarization, dependency modeling, and change or deviation detection [93]. Within the *visualization* and data mining stage, methods of clustering and summarization are mainly applied. Change detection can be achieved by interaction with the visualizations and methods of classification are primarily used in the *filtering* stage.

An example that combines automatic clustering of trajectories with a schematic visualization of the generated model is the ISS view (see Figure 6.1 (g)), which is discussed in detail in Chapter 6.3. The view applies trajectory bundling to summarize calculated trajectory clusters as visual feedback of the data mining results. Further, scalable video data exploration is achieved by scatter/gather browsing of the trajectory clusters. Cluster selection automatically adds a trajectory filter to the dataset and can be used to explore the dataset either by common structure or by comparison to the trained model.

Besides filtering, aggregation is the method of choice to enable scalable analysis of large data as well as facilitating visual pattern discovery by providing views of different scales on the data. Visualizations that apply data aggregation, such as the ISS, enable hierarchical exploration of the data from coarse to fine. Visual exploration and data aggregation further support the analysts' abilities of pattern discovery and thus the formulation of new ideas and mental models of the data.

Another example of the usage of aggregation techniques to leverage the analysts pattern mining abilities is the *chart view* [129] visualization (see Figure 6.1 (e)). This view depicts time series (e.g., of properties of trajectories, or additional one-dimensional time-dependent data streams) by different types of standard charts, and supports several granularities of temporal aggregation.

Another type of aggregation that is typically applied to enable data scalability and visual pattern mining is the reduction of the resolution of the presented data. Resolution reduction allows inspecting a larger area of the data space and puts emphasis on coarse

structures in the data. Most views of the proposed system support the adaption of either the spatial or the temporal resolution, or both. Despite the general adaption of spatio-temporal resolution of different visualizations, the representation interval of streamed and dynamically displayed data elements (i.e., video frames) can be adjusted. Besides conventional fast-forward video playback at constant pace, the proposed system also provides relevance-based sub-sampling of the video stream, called adaptive fast-forward (see Chapter 5.1). The display duration, either at constant sampling interval or adapted to relevances, is controlled by a central heartbeat mechanism to ensure temporally synchronized views on the data.

Situational awareness. Situational awareness is an important aspect in all parts of the analysis process. In the context of video fast-forward, two aspects become important to maintain the situational awareness of the users: the support of object identification and of motion perception, as will be discussed in Chapter 6.2. Furthermore, the current playback speed has to be communicated to the users if the playback speed is adapted to the frames' relevances [127]. Since all fast-forward visualizations have strengths and drawbacks in specific scenarios, we provide a variety of different fast-forward visualizations, which will be introduced and evaluated in Chapter 6.2, from which the analyst can choose the most suitable one according to task and data.

The proposed visual analytics system further addresses the issues of situational awareness arising when analyzing a large amount of dynamic video data. To cope with the perceptual deficits of humans, such as change blindness and inattentional blindness, the system provides a couple of data representations that support situational awareness. Inspired by the multiple-resource theory [303], the system provides both visual data displays and displays using another communication modality: auditory displays for video data (see Figure 6.1 (d)). Dependent on the level of data abstraction, either sonification of low-level video data [128] or sonification of higher level features, such as trajectories [126] can be applied. These sonification methods are briefly introduced in Chapter 6.7.

Another view on the video data that facilitates situation assessment is the VPG (see Figure 6.1 (c)), which is discussed in Chapter 6.4. The VPG displays a particular period of a continuous video stream in its spatio-temporal dimensions, using a sliding window. Hence, a 3D video volume is rendered, where time is extruded in the third dimension. Sparse sampling of video frames, additional illustration of extracted trajectory features, and viewpoint navigation increases visibility and allows inspection of interaction and activity patterns in the data. This dynamically summarizing visualization of a short sliding period of the video combined with its extracted features alleviates change and inattentional blindness.

Furthermore, several views, such as the VPG and the conventional video player, allow mapping the measured relevance of particular data elements on several attributes

of their visual representation. For example, the VPG allows mapping the relevance value of a trajectory to its display color, and object segments in the video player are surrounded by a color-coded bounding box. In the same way, the DOM of a particular data element to a defined fuzzy filter can be mapped to the color of its visualization. This allows displaying the filter confidence of a data element. Moreover, other uncertainties, such as the positional uncertainty of trajectories (calculated by the *feature extraction*) can be either modeled by relevance measures or directly visualized by the VPG as a semi-transparent blur surrounding the visual trajectory representation (Chapter 6.4). This helps the analysts be aware of their data quality.

The remainder of this chapter is organized as follows: the next section provides a review of state-of-the-art video visualization methods. Chapter 6.2 introduces a variety of fast-forward visualization methods including comprehensive evaluation. In Chapter 6.3, the ISS are discussed in detail. Then, VPG extensions for the application for video visualization of tracked moving objects are described in Chapter 6.4, before a feasibility study of the applicability of video visualization to snooker skill training is presented in Chapter 6.5, which also extends the original approach of the VPG. Afterward, a more general visualization for dynamic data, the *layered TimeRadarTrees*, is briefly discussed (Chapter 6.6). The visualization chapter closes with a brief discussion of two sonification approaches to support situational awareness for video surveillance by means of auditory displays.

6.1 Related Work on Video Visualization

Obtaining a quick overview of a video is an important task in many applications. Whether analyzing surveillance videos, obtaining a quick overview of a sports match, or selecting a movie to watch from a large DVD collection, watching the entire sequence is usually not an option. Instead, a quick summary of the crucial events happening in the video is desired. This can be done by summarizing the video by a number of short sequences, like in a cinema trailer, or by creating an image narrating the story. In some situations, meaningful information, such as motion flow, can be extracted and depicted in a way that helps the viewer recognize certain patterns or unusual events. These techniques are collectively referred to as video visualization.

In this section, video visualization techniques are categorized according to the content and format of the output media. This classification is based on the taxonomy depicted in Figure 6.2. As proposed by Borgo et al. [32], further classification possibilities are by goals, by input information, and by levels of automation.

This section only reviews video visualization, which has the goal to provide users with a tool to aid their intelligent reasoning while removing or alleviating the burden of viewing videos. A survey of video-based graphics, which aims to make use of

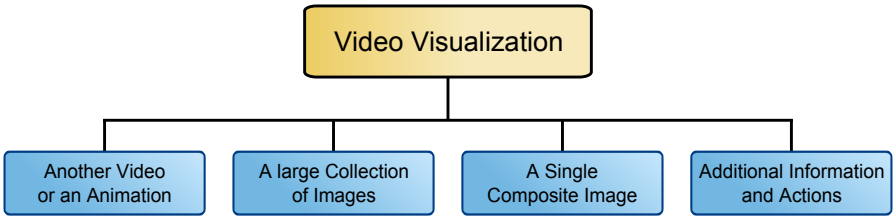


Figure 6.2 — Video visualization taxonomy: classification by output data types.

video content in creating computer-generated imagery for artistic appreciation and entertainment, can be found in Borgo et al. [32]. Moreover, a large body of literature addresses the issue of time-dependent data in general and its visualization. Background information of this topic is provided by a recent survey by Aigner et al. [7].

First, techniques for keyframe selection are discussed, followed by the first class of video visualization techniques that examine methods to generate new videos as an output media that are more "cost-effective" to view in comparison with the original videos. The two following sub-sections concentrate on common methods to summarize videos using keyframe storyboards, either by a large collection of images or by a single composite image. This is followed by a review of techniques for generating abstract visualization, where information in the temporal feature space is displayed to aid summarization and understanding of a video.

6.1.1 Keyframe Selection

According to the segments of a video (see Chapter 3.1) image-based video visualization commonly operates on the three lower levels: frames, shots, and scenes. For example, several frames might be selected and presented to the user, or the contents of a shot or a scene might be summarized in a single image. A crucial step for all these applications is the selection of keyframes, i.e., representative frames of the video. In the following, different keyframe selection techniques are outlined first. Afterward, different depiction methods are discussed before a number of techniques that incorporate additional information into keyframes to enhance understanding are considered.

As mentioned before, keyframe selection is typically the first step in image-based video visualization. Keyframe selection means that we are looking for a set of images that optimally represents the contents of the video according to a specified criterion, such as "find a representative image for each shot". As in most optimization procedures, two different strategies can be pursued when choosing relevant images. Either, a maximum number of frames is given or an error rate to be met. The maximum number criterion is commonly used when dealing with limited resources. For example, when the keyframes are to be displayed on a single page or transmitted to a mobile device at

a low transmission rate. The error rate approach is applied when looking for the best set of images meeting the optimality criterion. In both techniques, manipulating one parameter affects the other. Commonly, the number of keyframes and the error rate are correlated, i.e., if we allow a larger number of keyframes, the error will drop; and if we increase the allowed error in the second technique, we will obtain more images. Hence, when choosing a strategy, we have to decide what is more important: a fixed number of images or a limit on the error.

No matter which technique is chosen, in both cases an optimality criterion has to be defined. The simplest one would be to uniformly select images from the movie, but this might easily lead to missing short key sequences or several depictions of long uninteresting scenes. Truong and Venkatesh [285] classified a number of partly overlapping criteria for the optimization, which can be summarized in the following five categories:

- *Sufficient content change*: Choose keyframes so that they mutually represent different visual content. With the error criterion, we sequentially go through the video and select a frame as keyframe whenever it largely differs from the previous keyframes. Alternatively, we can look for the n frames that represent sequences of equal variance.
- *Maximum frame coverage*: Select keyframes such that they represent a maximum number of frames that are not keyframes.
- *Feature space analysis*: Treat each frame as a point in high-dimensional feature space. One optimization strategy is based on point clustering, where the keyframes are the representative points of the clusters. Alternatively, the video can be seen as a path in high-dimensional space connecting subsequent frames and we look for a simplified path with minimal error.
- *Minimum correlation*: Choose keyframes such that they feature a minimum amount of correlation between each other.
- *"Interesting" events*: Consider semantics and try to identify keyframes with high information content. Methods in this category might analyze motion patterns, look for faces, or high spatial complexity.

6.1.2 Another Video or an Animation

The class *another video or an animation* considers a group of techniques that alleviate the problem of watching videos without leaving the video output domain. There are three different approaches, differing in the way they maintain the content of the video.

The first category contains *video navigation* techniques. Here, the full content of the video is maintained. Content control and time compression are achieved via video

browsing approaches and fast-forward techniques.

Within the second category, *video montage* and *video synopsis*, a new video with a shorter duration is created by combining different spatial and temporal video parts. Spatial and temporal context information may be lost using this technique while the occurring actions are preserved.

The third category covers *video skimming* techniques, which skip uninteresting parts of the video to create shorter clips with the purpose of video abstraction. Due to the absence of whole video parts, time compression is achieved by the cost of information loss. However, the available parts maintain spatial context information.

Video Navigation

Many proposals have been made regarding the problem of watching videos in a timesaving and efficient manner. According to Li et al. [190], basic video browser controls include *play, pause, fast-forward, seek, skip-to-beginning*, and *skip-to-end* of video. Li et al. [190] added enhanced controls. The most important features include the support for modifying the playback speed between 50 % and 250 % of the original speed while preserving the pitch of the audio, an automatical pause removal feature that enables the user to remove parts of the video where pauses in continuous speech occur, and the possibility to select shots of the video to jump to their temporal positions.

Ramos and Balakrishnan [242] focused on controlling videos with pressure-sensitive digitizer tablets. Besides fading in/out annotations and several interaction possibilities, they present a variation of the fish-eye view called *twist lens* to seek in video streams. The time line slider consists of several sampled frames semi-occluded by each other. If the user coarsely selects a frame and increases the pressure, the slider is smoothly morphed around this frame into a sinusoidal shape. The occlusion of the frames in the vicinity of the selected one is decreased and an accurate selection of the time position is feasible.

Schoeffmann and Boeszoermenyi [255] created a time line slider as a combination of an arbitrary number of navigation summaries. This enables users to see several content abstractions of the video in the time line at one glance. Navigation summaries can be visited frames, dominant colors, frame stripes or a motion layout.

Another possibility to browse through videos is given by direct object manipulation approaches (e.g., [165, 111, 81, 113, 155]). To browse videos in this way, objects and their movements are extracted in a pre-processing step. Afterward, objects can be picked in the video window. The video is directly scrubbed by moving the selected object to another position. Kimber et al. [165] and Girgensohn et al. [111] also allow scrubbing by object manipulation on a floor plan.

As mentioned above, fast-forward is a basic control for video browsing. Wildemuth et al. [304] evaluated how fast too fast is. They recommended showing every 64^{th} frame of a video for fast-forward surrogates. Even at lower speeds, the user's abilities in object recognition (graphical), action recognition, linguistic comprehension (full text), and visual comprehension decrease. This problem leads to different approaches to adapt the video playback speed by video content as discussed in Chapter 5.1.

Ballan et al. [20] use SLAM techniques to generate a 3D reconstruction of the footages captured by typical consumer video cameras. The work is an example of how SLAM techniques can achieve real-time performance in the creation of free viewpoint video transition and, hence, allow users interactively exploring video from different viewpoints. The approach heavily relies on color-priors: foreground objects, for example, are required to have specific shapes or colors to avoid artifacts in the reconstruction.

Video Montage and Video Synopsis

Kang et al. [153] introduced a technique for video abstraction called *video montage*. They extract visual informative space-time portions from video and merge these parts. Their technique changes the temporal and the spatial occurrence of the information and results in a shorter video clip with condensed information.

One of the method's drawbacks is the loss of spatial context. Methods that preserve spatial positions were proposed by Rav-Acha et al. [244] and Pritch et al. [238, 239]. In their approaches, objects are detected, tracked, and temporally rearranged. The recomposed video shows different actions, occurring at different temporal positions, at the same time. Even if the trajectory of the object has a long time duration it is cut into several pieces, all displayed at the same time.

Video Skimming

The goal of video skimming is to create a short summarization of a given video stream. Therefore, less interesting parts of the video are discarded. The process builds upon the previously described *keyframe selection* (see Chapter 6.1.1).

Truong and Venkatesh [285] identified a five-step process for automatic video skim generation. For some video skimming techniques, particular steps are skipped or combined in a different variation, but the basics remain. These five steps are *segmentation* (extract shots, scenes, events, parts of continuous speech, etc.), *selection* (choose "interesting" parts for summarization), *shortening* (reduce the time duration for the selected parts further, for example, by cutting), *multimodal integration* (combine skims for different features such as image, audio, and text into the final skim), and *assembly* (temporally arrange independent video skim parts, for instance, chronological).

Correa and Ma [66] introduced *video narratives* that are single compositions of dynamic mosaics organized along a linear timeline. The system supports speed-varying skimming of videos as well as the generation of storyboard or dynamic video summaries.

The field of video skimming covers a huge research area; further information is provided by Truong and Venkatesh [285].

6.1.3 A Large Collection of Images

The easiest direct depiction of keyframes is the storyboard technique, where equally sized images are arranged on a regular grid, for example, three by four images on a page [19]. This technique can be extended to allow for different levels of temporal detail when presenting the keyframes in a hierarchical manner [186, 271]. At the top level, a single frame represents the entire film and at the lowest level, all frames are included. Although easy to apply and understand, both techniques have the disadvantage that they do not provide information about the relevance of individual snapshots. To include such semantics, the images can be scaled according to their importance to the video [315, 288]. Yeung and Yeo [315], for example, use the number of frames being represented by a keyframe, which is equivalent to the subset's length, to scale the keyframes of a sequence and arrange them according to predefined design patterns in a video poster. The illustration of several video posters in temporal order summarizes the content of a sequence. Barnes et al. [22] presented another approach to video summarization called *tapestries*, merging the structure of DVD chapter menus with the timeline representation of video editing tools.

6.1.4 A Single Composite Image

All methods in the previous category have in common that they do not alter the contents of the individual keyframes. Reassembled depictions, in contrast, combine the contents of several images to create a new one. An early goal in this area was to reconstruct the background of a scene. Methods to achieve such a reconstruction [149, 275, 187, 151], sometimes called *mosaics*, combine several successive video frames and reconstruct the scene while correcting for camera movement and zooming. *Salient stills* [279] extend this technique and add additional information about temporal changes. Therefore, salient regions of interest are extracted and seamlessly arranged on the background such that the temporal structure of the video content is preserved. A similar approach is followed by Pritch et al. [239], who concentrate on the simultaneous depiction of events happening at different times in the video.

An alternative approach is taken by techniques that extract relevant subsections of the keyframes and reassemble the sub-images to form a new image. The *stained-glass* visualization [59] first arranges the important components on a page and fills the

gaps in between with image data according to a Voronoi tessellation of the data. This approach was extended in the *video collage* [211] and *auto-collage* [253] algorithms, where a combination of template-based arrangement and an energy minimization algorithm is used to find good locations for the different sub-images. While the first method concentrates on boundaries of arbitrary shape, the second one concentrates on seamless transitions between the different sub-images.

6.1.5 Additional Information and Actions

In the last video visualization category, methods that add additional information to the representation are summarized.

Enhanced Stills

A well-known approach is the *schematic storyboards*, where annotations are added to illustrate the movement of persons or the camera [112]. Nienhaus and Dollner [222] take a similar approach using additional dynamics glyphs. Further image-based video visualization that enhance the raw data are graph-based approaches that depict, additionally to the keyframes, the interaction between different characters or the use of different scenes in a graph [15].

Video Abstraction

In some cases, abstract attributes, such as changes in a scene, changes between frames, motion flow, and pixel clusters, can be depicted visually to aid the understanding of a video using only one or a few visualizations. Such visualization may not display objects in an intuitive manner, but the abstract visual representation can convey temporal attributes more effectively than discrete keyframe displays.

A popular approach interprets video data as a space-time volume, e.g., as published by Fels and Mase [94]. Here, the spatial axes x and y are combined with time as the third axis. Within this representation, they define *cut planes* to intersect the video volume. Cut planes can be arbitrarily defined to watch the video in a different way. Normally watching video in this context is nothing but applying using a cut plane parallel to the x-y axes that is moving along the z-axis. The principle of cut planes through a video volume were refined for other applications like cut outs or non-photorealistic rendering [166].

Daniel and Chen proposed to employ volume visualization techniques to visualize the video volume with the aim of summarization [72]. They transformed the video volume into other shapes, for example, a horseshoe view, to convey more information. A change detection filter was applied and the results were displayed in the volume. Within this visualization, several visual patterns can be identified that indicate events,

such as changes that remain for a period, walking with moving arms, or an opened door.

Chen et al. [55] investigated *visual signatures* as abstract visual features to depict individual objects and motion events. To this end, they applied and evaluated flow visualization techniques to video volume visualization. Example visual signatures they used to evaluate their approach are a temporal visual hull, a color-coded difference volume, glyphs, and streamlines, where a sphere moves toward the upright corner of the image frame.

A further enhancement was done by Botchen et al. [35]. With the VPG, they presented a focus and context design of video visualization, combining keyframes and trajectories into a visualization. They created a visual representation of a continuous video stream in a manner similar to an electrocardiogram or a seismograph that enhances the video volume visualization approach additionally with semantic annotations. The visualizations in Chapter 6.4 and 6.5 are based on this approach and develop it further.

In the next section, visualization techniques for video fast-forward are introduced and evaluated.

6.2 Video Visualization for Fast-Forward

Conventional as well as adaptive fast-forward approaches necessitate the possibility to play video faster, but the frame rates cannot be increased arbitrarily due to the physical constraint of video display devices (usually 60–200 Hz). For this reason, higher accelerations are typically achieved by discarding frames (*frame-skipping*). Frame-skipping affects the human perceptual performance even at normal playback-speed (in this context termed time-lapse). Keval and Sasse [164] experienced a strong decrease of crime detection performance for time-lapse video presentation in their experiment and Scott-Brown and Cronin [256] pointed out that video in time-lapse format disrupts motion perception, and thus, increases change blindness. The authors of both works illustrate the importance of proper object/event identification and motion perception for video surveillance and the challenges that may arise by time-lapse video. However, the psychological influence of frame-skipping in fast-forward scenarios has not been investigated yet and remains an open research question.

In this section, three video visualization techniques based on fast-forward are introduced, evaluated, and compared against the state-of-the-art method frame-skipping (Figure 6.3 (a)). The developed methods are *temporal blending* (Figure 6.3 (b)) that addresses the motion disruptions introduced by frame-skipping, *object trail visualization* (Figure 6.3 (c)) that leverages a combination of frame-skipping and temporal blending, and *predictive trajectory visualization* (Figure 6.3 (d)) that supports motion perception by augmenting the video frames with an arrow that indicates the future object trajectory.

A controlled laboratory user study (n = 24) was conducted to determine the trade-off between support of object identification and motion perception, two properties that have to be considered when choosing a particular fast-forward visualization. The evaluation covers user performance and user experience according to the taxonomy of Lam et al. [179]. The evaluation focuses on the trade-off between motion perception and object detection and identification to account for the main challenges of fast-forward visualization in the context of video surveillance. Although the perception of animation has been playing an important role in visualization research (see, for example, the textbook by Ware [296]); this work is—to the best of my knowledge—the first user study to evaluate different visualization techniques for compressed time rendering of motion images.

Since playback-speed adaption of adaptive video fast-forward methods hinders correct object speed estimation in video [127], users must be made aware of the adaption factor used for adaptive video fast-forward. Communication of the current playback-speed should not distract attention from the video. Moreover, it should be presented in a way that matches users' expectations. To convey the playback-speed information, a *speedometer* was utilized by Cheng et al. [57]. Besides a modified version of this speedometer, this section introduces two new visual representations for playback-speed (*color frame*, *analog VCR fast-forward*) and evaluates their performance in terms of subjective effectiveness, level of distraction, and workload.

6.2.1 Fast-Forward Video Visualization Approaches

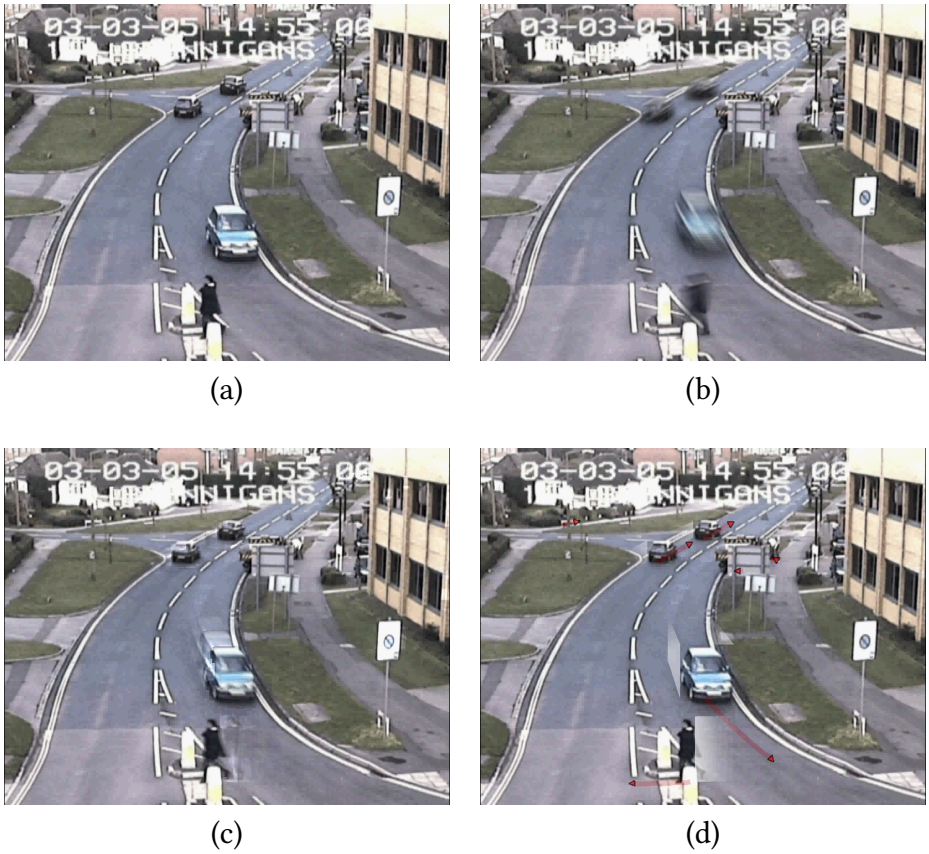
In general, fast-forward video visualization has the objective to communicate the information of a certain number of frames from a source video, n_{src}, within n_{dst} frames in the destination video. If we use the same frame rate for the destination video that we have in the source video, then the relation between these quantities depends solely on the fast-forward acceleration factors a_i, which can vary frame-wise in the case of adaptive fast-forward: $n_{\text{dst}} = \sum_{i=1}^{n_{\text{src}}} 1/a_i$.

Frame-skipping

The typical approach to boost the playback speed in video fast-forward is to discard as many frames as required to obtain the desired acceleration factor [127]. In detail: The j-th destination video frame f_{dst}^{j} is the i-th frame of the source video. The indices $i \in I$ are determined by Eq. 6.1. All other frames are skipped (see Figure 6.4).

$$I = \left\{ i \,\middle|\, \exists j : \; j \leq \sum_{k=1}^{i} \frac{1}{a_k} \;\wedge\; j > \sum_{k=1}^{i-1} \frac{1}{a_k} \right\} \tag{6.1}$$

Since the original video frames are displayed in frame-skipping, the appearance of objects is preserved and should be as well observable as in the original video (see



(a) (b) (c) (d)

Figure 6.3 — Comparison of four fast-forward video visualization techniques: (a) frame-skipping, (b) temporal blending, (c) object trail visualization, and (d) predictive trajectory visualization. A sample video is available at the project's website: `www.vis.uni-stuttgart.de/index.php?id=vva`.

Figure 6.3 (a)). Another advantage is the low computational cost. If an appropriate video codec is chosen and the number of keyframes is adequate, the video can be accelerated nearly without limits: the number of frames considered for visualization remains constant.

Temporal Blending

The temporal blending approach is developed to alleviate the interruptions of frame-skipping including their perceptual issues. It is motivated by the human visual system, which summates visual stimuli over time [21, 115]. Fast moving objects thus appear blurred (see Figure 6.3 (b)). This effect can be observed, for example, by swinging torches at night, as the integration time of the human eye depends on the luminance of the environment. Motion streaks from such blurring can support the perception of

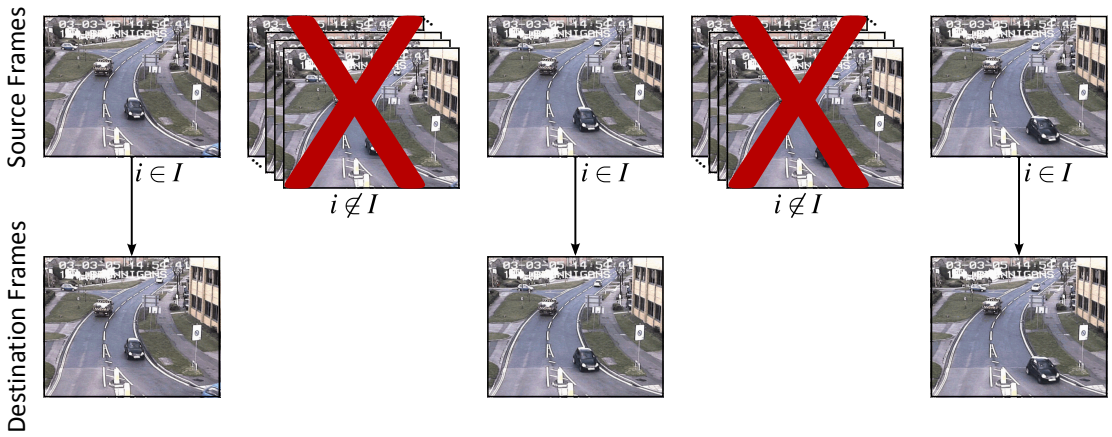


Figure 6.4 — Frame-skipping. To achieve the desired frame rate, source frames are discarded.

motion direction as reported in psychophysical experiments [107]. Temporal blending emulates the blurring effect by integrating the frames in a similar way.

Recorded video footage is typically gamma-encoded, which needs to be considered. The original reason for the gamma correction is the nonlinearity of CRT monitors: if we double the voltage of the signal sent to the display device, the radiometric (physical) brightness does not double. To address this problem, most of the video and image capture devices internally gamma encode the signal. The linearly scaled chromaticity stimuli (linear RGB) are nonlinearly transformed to sRGB. If those nonlinear values are now presented to the user on a CRT monitor, the intrinsic gamma decoding characteristic automatically re-transforms the signal to linear RGB (Figure 6.5 (b)). For this purpose, a gamma pre-correction is included in other display devices like LCDs. In the simplest case, the gamma-corrected (R', G', B') values are calculated by $(R', G', B') = (R^\gamma, G^\gamma, B^\gamma)$. Images are typically encoded by the camera with $\gamma = \frac{1}{2.2}$ and decoded by the display device with $\gamma = 2.2$. A similar argument can be made for other color systems like YIQ or YUV, known from video systems. For more background information on gamma correction and color systems see Fairchild [91] and Poynton [237].

Originally, the observed scene is not changed by an imaging–displaying process as depicted in Figure 6.5 (a). To achieve physiologically correct integration, the approach blends the frames in linear RGB space. Therefore, the input frames (which are already gamma-encoded) are first gamma-decoded back to linear RGB. Now correct blending is possible. After this step, the resulting images are gamma-encoded again (see Figure 6.5 (c)).

The integration time for the visualization depends on the acceleration factors a_i. In

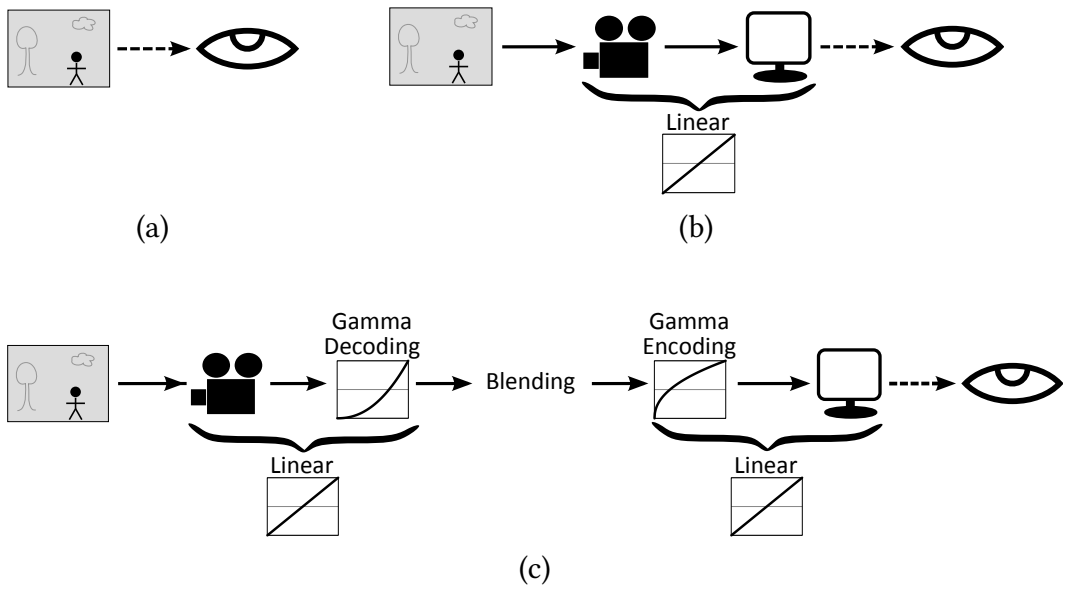


Figure 6.5 — (a) Visual stimuli arrive the eye unchanged. (b) Videos and images captured by a camera are gamma-encoded. The monitor reverses the nonlinear transformation (gamma decoding). (c) Artificial integration of visual stimuli according to the human vision system. Before blending, the image is transformed to linear RGB and finally gamma-encoded again.

detail, each source frame f^i_{src} is added with the weight $1/a_i$ to the destination frame ($f^j_{\text{dst}} = \sum_i \frac{1}{a_i} \cdot f^i_{\text{src}}$) until the sum of weights is one ($\sum_i \frac{1}{a_i} = 1$). To satisfy this, the weight of a source frame $1/a_i$ may be split and the source frame will affect multiple destination frames. The whole rendering process is depicted in Figure 6.6.

From the perspective of computer graphics, temporal antialiasing by temporal blending is usually called *motion blur* [296] (see Navarro et al. [221] for a review of the state-of-the-art of motion blur rendering). In contrast to frame-skipping, temporal blending communicates object movement by integrating all available frames. However, object identification may become difficult due to the blurred appearance of objects (see Figure 6.3 (b)).

Object Trail Visualization

Object trail visualization is developed to combine the advantages of frame-skipping (object identification) with the support of motion perception of temporal blending: multiple entities of objects are visualized simultaneously with increasing transparency (see Figure 6.3 (c)). The appearance is inspired by *dynamic stills* [50], in which a whole video clip is summarized as one static image. The difference is that the duration that is

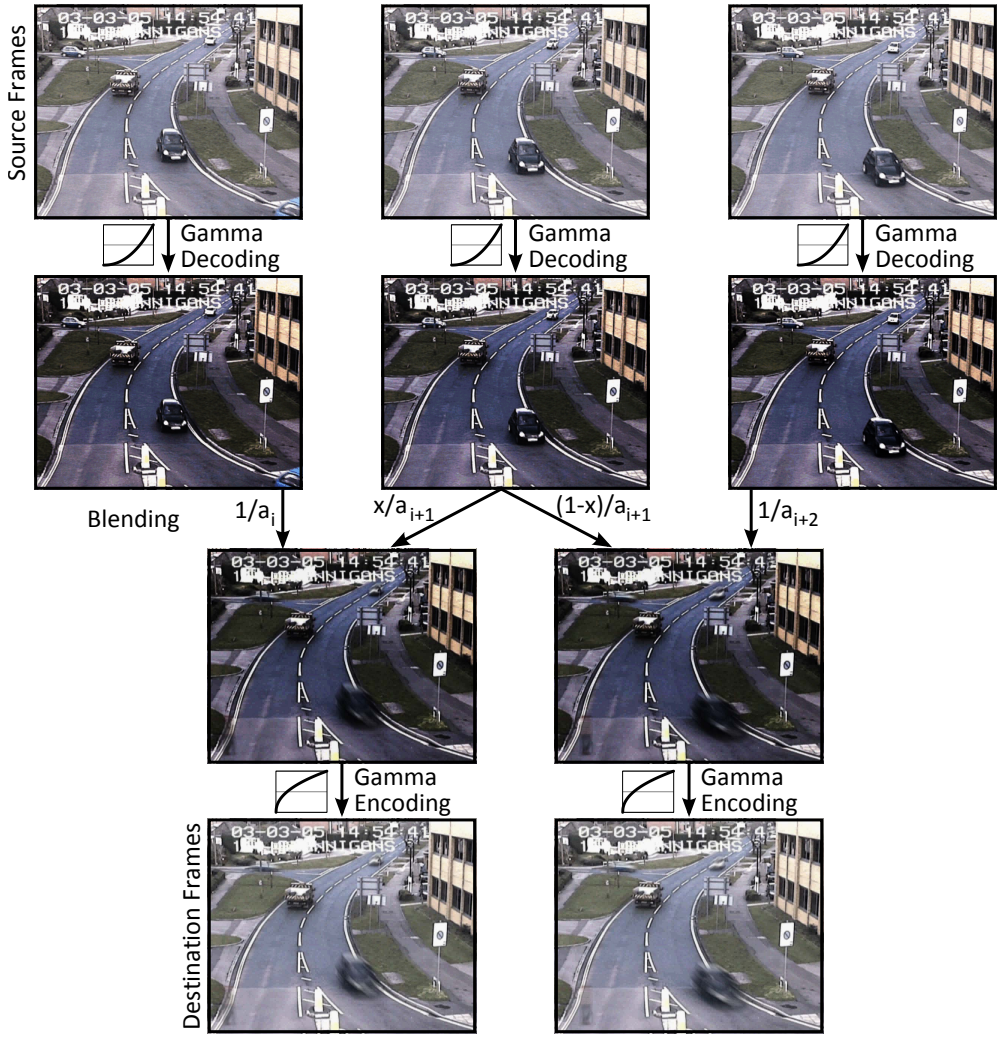


Figure 6.6 — Temporal blending. Source frames are gamma-decoded (top) before they are blended with respect to their acceleration factors a_i (center). After blending in linear RGB space, the resulting frames are gamma-encoded again (bottom).

summarized is much shorter and used for each destination frame. This effect is also known from Microsoft's *pointer trails*, which can be used since Windows 3.1, or from high-density cursors [24] in order to enhance the visibility of mouse movement. The approach renders older entities of an object with higher transparency, similar to the *salient video stills* [278]. The main difference between object trail visualization and the two mentioned approaches (dynamic stills and salient video stills) is that object trail visualization does not summarize whole sequences in a single image: it creates many summarizations—one for each destination frame.

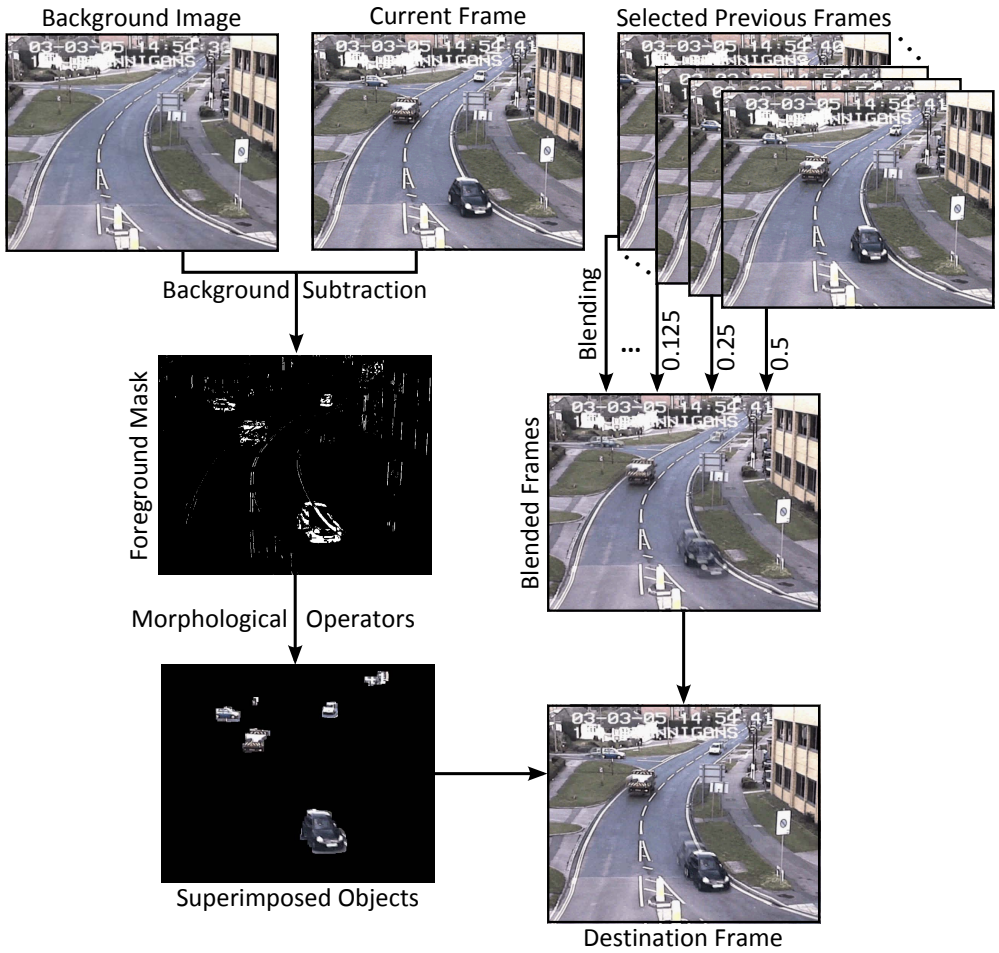


Figure 6.7 — Rendering process of object trail visualization. Left: object enhancement responsible for proper object identification; right: frame blending responsible for proper motion perception; bottom right: destination frame with superimposed objects on the blended frames.

The rendering process has two stages (see Figure 6.7), which can be computed in parallel: frame blending and object enhancement. The first stage is dedicated to movement perception; the second stage facilitates object identification.

Stage 1: Frame Blending. A particular number of frames ($m - 1$) are blended. The distance k between the frames is determined according to the acceleration factors a_i. The blended image is calculated by $f^j_{\text{blend}} = \sum_{o=1}^{m-1} w_o \cdot f^l_{\text{src}}$, where $l = i - o \cdot k$ and the normalized weights w_o of older frames are reduced according to $w_o = \frac{2^{-o}}{\sum_{\omega=1}^{m-1} 2^{-\omega}}$ (see also Figure 6.7).

Stage 2: Object Enhancement. The blended image is superimposed by objects

present in the recent frame. First, the approach extracts the background image by the running Gaussian average method [233], then it calculates a foreground mask by background subtraction and thresholding, and finally applies morphological operators.

Besides that, the method highlights the object movements by additionally blending the object masks of the previous frames semi-transparently: older entities receive a higher luminance here (see Figure 6.3 (c)). Optionally, the movement emphasis can be disabled (as in Figure 6.7).

Predictive Trajectory Visualization

The predictive trajectory visualization is designed—similar to object trail visualization—to allow for good object identification and motion perception. It uses abstract trajectory illustration to improve the perception of moving objects. A special characteristic is that the visualization contains movement information from past and information about future object movements alike. It is inspired by storyboarding techniques [112] and *augmented keyframes* [52], which use arrows to show movements in still frames.

In detail, it consists of a keyframe that is chosen in the same way as in frame-skipping, a tail for each object that depicts motion from the past, and arrows to forecast the objects' positions in future (see Figure 6.3 (d)). The design decision to use arrows to indicate future movements is founded in related work of flow visualization: arrows are widely used to indicate motion [300]. The rendering process is illustrated in Figure 6.8 and utilizes extracted and tracked objects with their bounding box information (for details on object extraction and tracking see Chapter 3).

In predictive trajectory visualization, objects are visible for a longer duration: their arrows are visible before they enter the scenery. Thus, the observer can identify the objects earlier. However, this benefit comes at the expense of visual clutter, which may occur in videos with many moving objects.

6.2.2 Adaptive Fast-Forward Playback Speed Visualization

In adaptive video fast-forward, the playback velocity of each frame varies according to a measure of relevance. A user study by Höferlin et al. [127] showed that it is difficult to differentiate between variations in video playback speed and changes of the movement velocity of objects. To interpret video material correctly, it is indispensable to estimate the velocities of objects. An event including a moving person, for example, would be interpreted completely different if the observers realized that the person is running or if they imagined that the person is walking. To avoid such misinterpretations, the participants of the user study "suggested to add visual feedback to increase awareness of playback speed" [127].

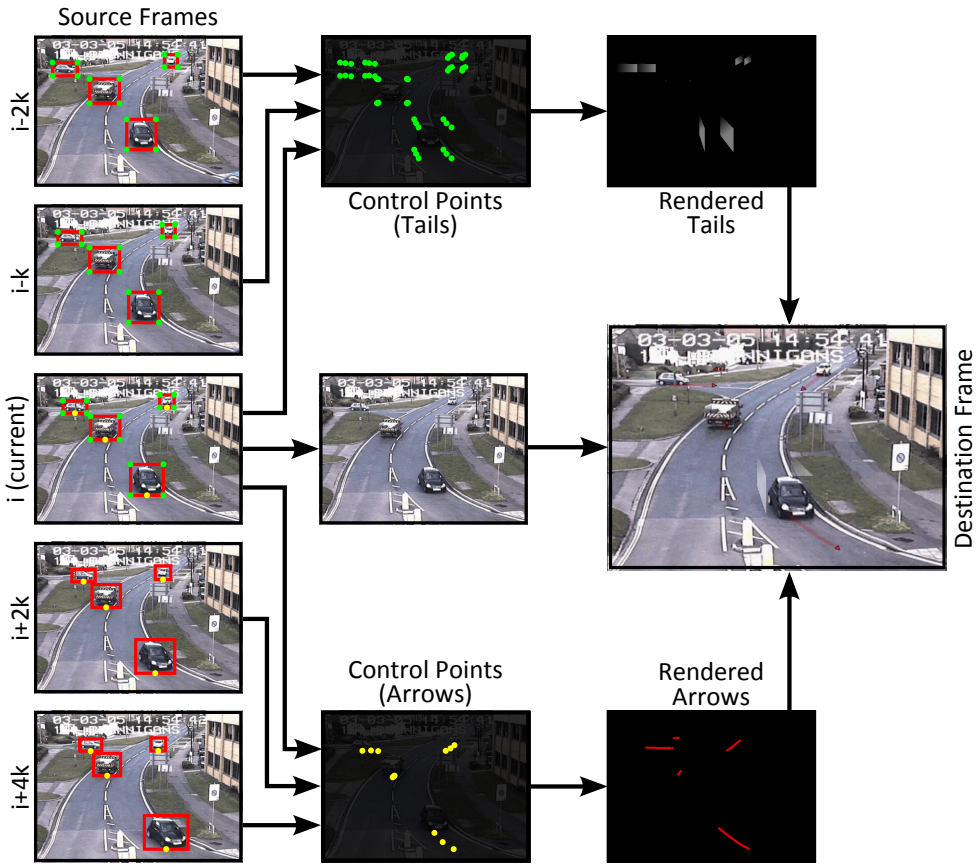


Figure 6.8 — Rendering process of predictive trajectory visualization. The current frame i is selected as in frame-skipping. For tail rendering (top), the edge points of the objects' bounding boxes (green points) are used from the current frame i and the two previous frames $i - k$ and $i - 2k$. For arrow rendering (bottom), the center bottom points of the objects' bounding boxes (yellow points) are used from the current frame i and the two future frames $i + 2k$ and $i + 4k$ (doubled frame distance). The distance k can be adapted to the playback speed (here: $k = 5$). Each group of three temporally shifted control points is used to create a quadratic Bézier curve. The tail is rendered by two surfaces that show a semi-transparent fading white gradient. Each is defined by the two vertical Bézier curves coming from the top and bottom vertices of one of the bounding box sides. The arrowshaft (defined by its Bézier curve) is superimposed by an arrowhead pointing in the direction of motion.

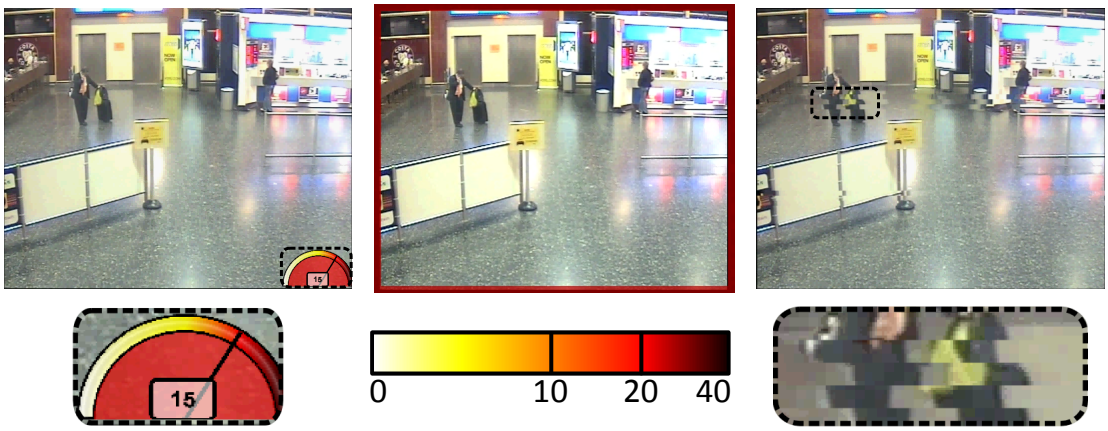


Figure 6.9 — The three proposed adaptive fast-forward playback speed visualizations. Left: speedometer; center: color frame; right: analog VCR fast-forward. The color scheme is depicted in the bottom center.

Therefore, three possibilities to indicate the playback speed of adaptive fast-forward are discussed in the following: the *speedometer* (Figure 6.9, left), the *color frame* approach (Figure 6.9, center), and the *analog VCR fast-forward* visualization (Figure 6.9, right).

Speedometer

The idea to communicate the velocity of adaptive fast-forward by a speedometer was applied by Cheng et al. [57]. They show a needle that swivels to the current playback speed as well as the numeric value. For the user study, this visualization is adopted and enhanced by color mapping (see Figure 6.9, left). Therefore, a heated body scale color map is applied, which is appropriate for ordinal data [33]. The applied variant is the Matlab's color map *hot*, depicted in Figure 6.9 in the bottom center.

The speedometer communicates the playback speed by a metaphor known from cars. Since the peripheral vision has low spatial resolution compared to the fovea [117], identifying the exact position of the needle or reading the numeric speed value requires visual focus of the observer. Since attention should be given to the video, a half-circular area is added at the center of the speedometer that shows the current speed by the color from the color map. The area is larger than the thin needle and the displayed digits, and thus, is better identifiable by peripheral vision.

Color Frame

In the color frame visualization, the video is enclosed with a thick frame that is inked according to the playback velocity (see Figure 6.9, center). The applied color map is

Figure 6.10 — Analog VCR fast-forward visualization. Pixels inside the black rectangles are randomly shifted to generate horizontal distortion lines. The playback speed is indicated by the number of distortion rows and the distortion width.

identical with the one used for the speedometer. This visualization is implemented and included in the user study since it does not require visual focus, and thus may reduce the required attention to assess the playback speed. In contrast to the speedometer, only a coarse impression of the playback speed can be provided by the color frame.

Analog VCR Fast-Forward

The third playback speed visualization, analog VCR fast-forward, is based on the characteristic horizontal distortion lines that occur for fast-forwarding analog video data (see Figure 6.9, right). The approach maps the playback speed to the amount and thickness of the distortion rows, as shown in Figure 6.10. Horizontal distortion lines are generated by shifting the pixels inside the black rectangles of Figure 6.10 into a random direction (left, right) with a random magnitude (denoted by horizontal arrows): $\Delta_x = -1^{\text{rand}() \mod 2} \cdot (\text{rand}() \mod \widehat{\Delta}_x)$, with the maximum amplitude $\widehat{\Delta}_x = 10\,\text{px}$. The shifting direction and magnitude vary between frames. The playback speed is mapped in 10 levels to the distortion width (up to 3) and the amount of distortion rows (up to 3), where the absence of distortion lines indicates original playback speed.

In contrast to the approaches using color mapping, the users do not need to learn any color scheme. The visualization uses a known metaphor, and thus, can be interpreted without training. Similar to the color frame visualization, the playback speed can be recognized without focusing on particular screen regions. Nevertheless, the visualization can be characterized as qualitative visualization since it is difficult to identify the exact playback speed. Additionally, the video signal is distorted, which may negatively influence the perception of the video.

6.2.3 User Study

In the user study, the four fast-forward visualizations are compared in terms of object identification (measured objective and subjective) and motion perception (subjective), and the three playback speed visualizations in terms of playback speed feedback (subjective). Three different videos are used in which artificial search targets have been partially added. The user study applies a within-subjects design.

Hypotheses

A trade-off between support for object identification and motion perception for the fast-forward visualizations and a trade-off for the speed visualizations between conveying precise information and inducing less distraction is expected. Thus, the following hypotheses are tested:

- **Hypothesis 1 – Object identification.** It is expected that frame-skipping shows the best results in supporting object identification because it preserves the original video frames without distortion or superimposed information inducing visual clutter. This hypothesis is also motivated by findings from perception research that indicate that there is no active deblurring mechanism for motion perception in the human visual system [48]. Although object trail visualization and predictive trajectory visualization augment information and modify the video frames, it is expected that they show good results. At high playback speed, the time an object is depicted at a particular position lasts longer for the object trail visualization (previous entities fade out), which should improve identification. Predictive trajectory visualization highlights objects, which attracts the attention of the users. However, too many objects may cause visual clutter. Temporal blending may show the worst results: motion blur at high playback speed may hinder correct identification of search targets.
- **Hypothesis 2 – Motion perception.** By discarding frames, frame-skipping may disrupt motion perception. Both temporal blending and object trail visualization additionally merge objects of the current frame with previous instances, which may improve motion perception. Predictive trajectory visualization includes information about the past and future movement of objects. Therefore, it is hypothesized that it shows the highest performance in motion perception.
- **Hypothesis 3 – Playback speed feedback.** The speedometer is expected to be the most efficient visualization for playback speed feedback due to its detailed information representation. However, since it requires visual focus, it may be most distracting. The color frame visualization has least influence on the video scene and shows only marginal distraction. The color mapping can represent a wide range of possible fast-forward accelerations; thus, it may be the best

trade-off between visual distraction and the accuracy of conveyed information. Estimating the pace of playback via analog VCR fast-forward does not require visual focus. However, this benefit comes at the expense of distorted video rendering and a rather coarse granularity of speed feedback. Thus, it is expected to be the least preferred visualization.

Each hypothesis is tested separately by a specifically designed task.

Stimuli and Tasks

In the experiments, the independent variable of interest is the visualization type. Object identification and movement perception were tested with the fast-forward visualizations frame-skipping, temporal blending, object trail visualization, and predictive trajectory visualization. Playback speed feedback was evaluated with the speed visualizations speedometer, color frame, and analog VCR fast-forward. Each participant had to perform each of the three tasks **T1**, **T2**, and **T3**. Each of the three tasks was designed to check one of the three hypotheses.

In this study, three videos were used to compare the visualizations with each other. The videos are benchmark datasets used in different areas, such as tracking, and have in common that they show real-world scenarios and originate from CCTV cameras. Video **V1** (from the BEHAVE dataset[2], resolution: 640 × 480 px, 25 fps) and video **V2** (from the AVSS dataset[3], resolution: 720 × 576 px, 25 fps) were used for the two tasks concerning the fast-forward visualizations (**T1** and **T2**). These videos were chosen to account for different amount of activity and complexity of movement. The first video (**V1**) is less complex and less populated than **V2**, it includes less perspective distortion, and covers a smaller area. Video **V3** (from the multi-camera i-LIDS dataset[4], resolution: 720 × 576 px, 25 fps) was used for the playback speed visualization task. This video was chosen since it was also used to evaluate relevance measures for adaptive fast-forward [127, 131] (see also Chapter 5.1). The tasks the participants had to perform were:

- **T1 – Object identification** *(objective and subjective)***.** The participants had to watch **V1** accelerated by factor 20 and **V2** accelerated by factor 10, using each of the four visualizations. These acceleration factors were determined during the pilot study. The stimuli had lengths of 45 s and 106 s, respectively. As search target, animated cartoon figures were artificially inserted into the videos, walking

[2] BEHAVE Interactions Test Case Scenarios
http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/
[3] i-LIDS dataset for AVSS 2007
[4] i-LIDS multi-camera tracking scenario dataset
http://www.homeoffice.gov.uk/science-research/hosdb/i-lids/

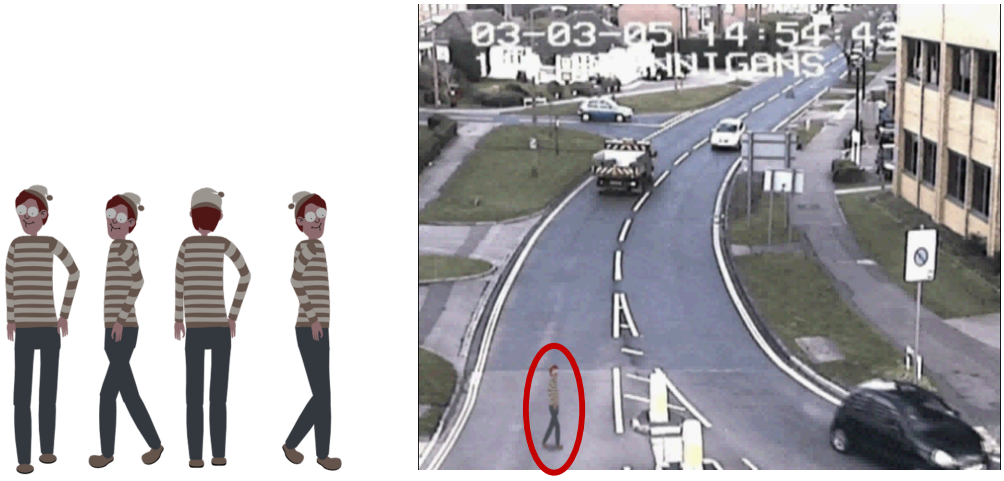


Figure 6.11 — In **T1**, the search target to be identified was a cartoon figure (left) that was added to the video. Right: Example frame of **V2** with the artificially added search target (marked red).

through the scene like normal persons (see Figure 6.11). The participants had to detect and recognize the cartoon figures. The brightness and contrast of the cartoon figures were adapted to the video to avoid pre-attentive perception. Comic figures have the advantage that they can be easily rendered into an existing scene without recapturing the video and feature a good recognition value. The detection had to be indicated by pressing a buzzer. A detection was counted to be correct if the buzzer was pressed while the cartoon figure was present in the sequence or had left not earlier than two seconds before. Each appearance of the cartoon figure was only counted once. Additionally to this objective measure, the visualizations had to be rated by the participants. The spatial and temporal positions of the cartoon figure varied to avoid learning effects—in total, four different versions of each video (**V1** and **V2**) were used. The cartoon figure appears 13 times in each version of **V1** and 8 times in each version of **V2** (the appropriate numbers were determined in the pilot study). To counter-balance the experiment, the presentation order of the visualizations was permuted. To balance the video versions, version 1 of the video was always shown first, then version 2, and so on. This assured that each visualization was presented the same number of times with each version of the video and potential side effects arising from different difficulties of the variants were avoided.

- **T2 – Motion perception** *(subjective)***.** Each combination of pairs of the different visualizations (i.e., six visualization pairs) were presented and had to be compared by each participant. For each pair, two visualization stimuli of 10 s were

successively presented with a pause of 3 s in between. Three of the pairs showed a snippet of **V1**, for the other three pairs a snippet of **V2** was used. The order and the composition of the pairs of the presented visualizations were permuted to counter-balance the experiment. The participants had to judge for each of the pairs which performed better in terms of motion perception. There was also the option to rate the performance of the two visualizations as equal. Paired comparisons were used since the number of items from which to choose increases the cognitive burden and negatively influences the quality of the results [109].

- **T3 – Playback speed feedback** *(subjective)***.** The participants had to watch **V3** in adaptive fast-forward three times with each of the three speed visualizations applied. After the presentation of the stimuli, the participants had to rate each visualization in terms of effectiveness of playback speed feedback, perceived workload, and distraction from the video content. The presentation order of the visualizations was counter-balanced between the participants.

Pilot Study

Before conducting the study, the study design was checked by a pilot study with four participants. The pilot study allowed identifying potential issues with the tasks and stimuli. It turned out that **V1** with an acceleration factor of 16 and 9 search targets was too easy for the first task. The acceleration factor was increased to 20 and four additional search targets were inserted to the scene.

Environmental Conditions and Technical Setup

The user study was conducted in a laboratory isolated from outside distractions. The room was artificially illuminated and only objects relevant for the study were contained inside. The participants were instructed to turn off their mobile phones.

The videos were presented in full screen on a laptop with a 16-inch monitor and a screen resolution of 1366×768 px. The distance between the participants and the screen was 50–80 cm. A video player was implemented that allowed to record the user input during the first task, in which the participants used a buzzer to indicate a detection of the search target.

Participants

The study was designed as within-subjects study and was conducted with 25 participants. Due to the lack of interest of one participant, the test results of 24 participants are considered. Seven of those participants were female and 17 were male, but gender was not considered as relevant factor. The average age was 25.5 years; the youngest participant was 20 and the oldest 31. Most of the participants were students from the

University of Stuttgart. Six participants studied at other universities or had already graduated. Half of the participants were students of computer science or software engineering. The others had majors in humanities or other subjects of study. All participants were tested with a Snellen chart and had normal or corrected-to-normal vision. The experiment took 45–55 min, depending on the particular speed of each participant. Each participant was compensated with EUR 10.

Study Procedure

The participants were first asked to fill out a questionnaire about their gender, age, and field of study.

Then, a short tutorial (about 6 min) introduced the fast-forward visualizations using snippets of **V1**. Afterward, a video snippet with a cartoon figure was shown to the participants to explain task **T1**. The video snippet was not reused in the tasks and it was presented in original playback speed. The first task was performed using **V1**. The participants watched all four visualizations and had to identify the cartoon figures. The start of the playback of each visualization was initiated by the participants. Hence, small breaks of individual length were possible between the visualizations.

After completing the task, the participants were asked to fill out a questionnaire about the effectiveness of the visualizations (*subjective effectiveness*), how comfortable they felt with the visualizations (*comfort*), how useful they considered each of the visualizations (*usefulness*), and the *effort* it made to watch them. For this rating, a 10-point Likert scale was provided. Furthermore, they were asked which visualization they preferred for performing the task (forced choice). Finally, the possibility was provided to give free comments about the visualizations. The same procedure was repeated with **V2**.

For task **T2**, the participants were briefed on paying attention to object movement. The visualizations were presented pairwise, using snippets of **V1** and **V2**. After the presentation of a pair, the participants had to decide which visualization supported motion perception better, or wheter both performed similar. After **T2**, the subjects were asked to comment on the visualizations.

Before the experiment with the third task was conducted, the different playback speed visualizations were introduced in a short tutorial (about 4 min in length). Then, one stimulus for each of the three visualizations (calculated on **V3**) was successively presented to the participants. The duration of each stimulus was approximately 1 min. Afterward, the participants were asked to fill out a questionnaire to judge the playback speed visualizations in terms of the *effectiveness* of the speed feedback, the *effort* required to interpret them, and the *distraction* from video they induced, on a 10-point Likert scale. In the end, they also pointed out their preferred visualization technique (forced choice) and could comment on all speed visualizations.

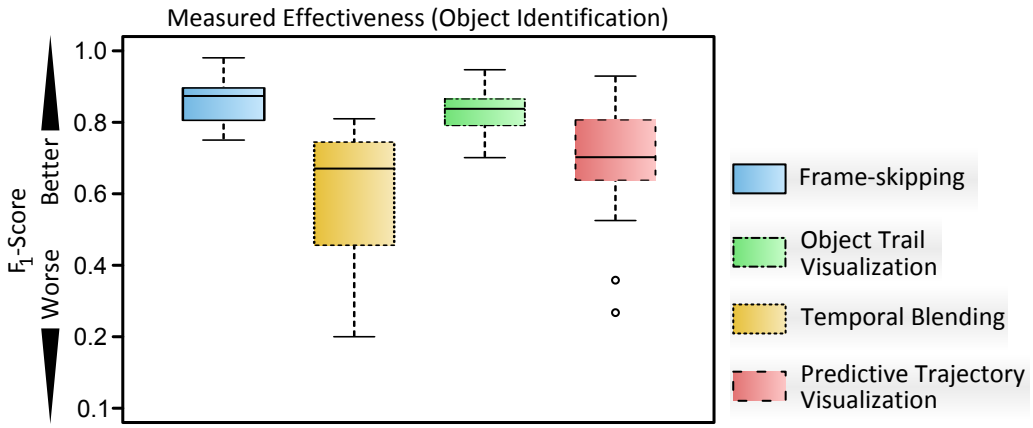


Figure 6.12 — F_1-scores for object identification (**T1**).

There was a "Give Up" option throughout the study; however, it was not used by the participants. The time to perform the tasks was limited by the duration of the videos. There was no time limitation between the videos, so the participants decided when to continue.

6.2.4 Study Results

The statistical analysis includes the results of 24 participants. The significance of the results are tested with non-parametric tests since not all results are normal distributed. For statistical computing, the software from the *R Project*[5] was used.

Results of the Object Identification Task T1

The results of task **T1** are divided into the measured effectiveness (objective results) and the results of the questionnaire (subjective results).

Results of the Measured Effectiveness. To measure the performance of the participants in task **T1** (object identification), the F_1-scores ($F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$) are calculated by determining the precision and recall using the buzzer input. The boxplots of the F_1-scores of all subjects for each visualization are depicted in Figure 6.12.

The non-parametric Kruskal-Wallis test shows a significant effect of the visualization type on object identification ($H(3) = 46.63$, $p < 0.005$).

Post-hoc pairwise U-tests confirm that frame-skipping (median of F_1-scores: 0.87) and object trail visualization (0.84) show significantly better results ($p < 0.005$, Bonferroni-

[5] The R Project for Statistical Computing. URL: http://www.r-project.org/

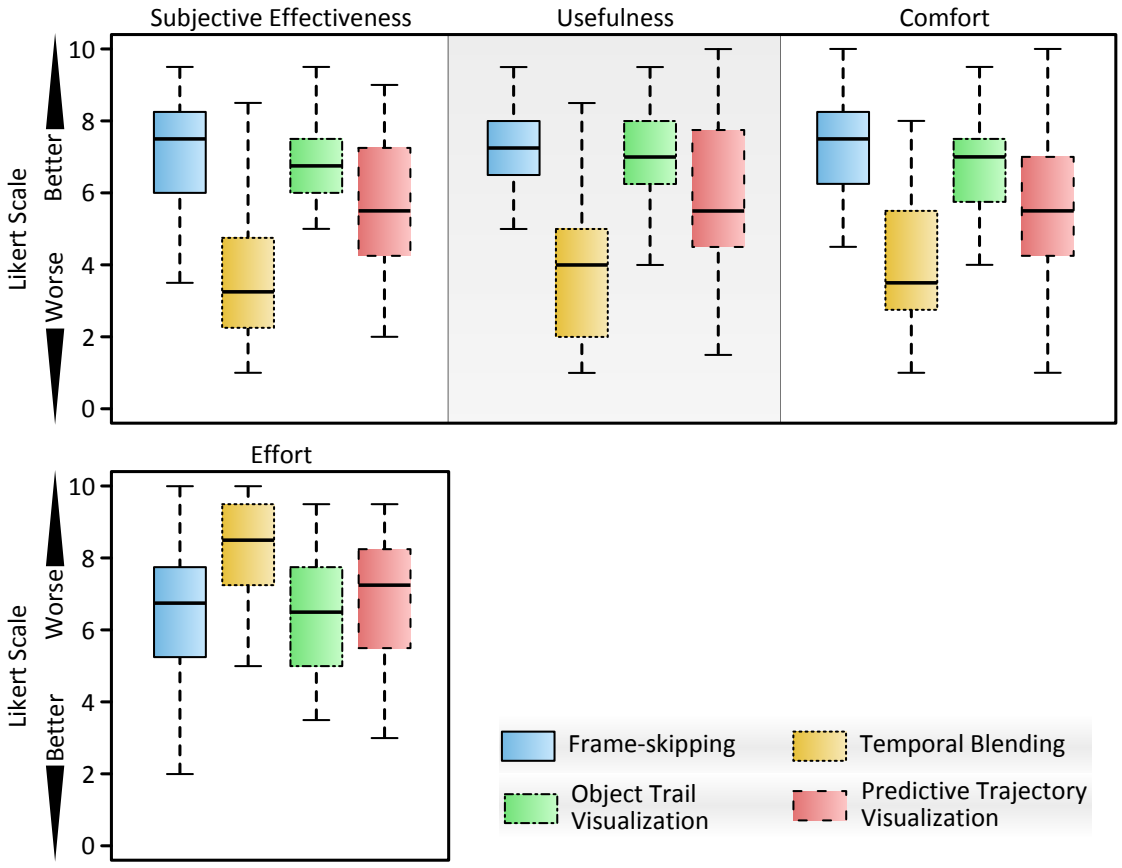


Figure 6.13 — Boxplots of the results of the questionnaire of task **T1** (the whiskers represent the lowest / highest values within one and a half times interquartile range to the median).

corrected for multiple comparisons) than the other two visualizations (temporal blending: 0.67, predictive trajectory visualization: 0.70). According to the comments of the participants, the latter visualizations suffer from information overload.

Frame-skipping and object trail visualization do not show any significant differences. However, some subjects described the trails to be useful for identifying the objects.

With temporal blending and predictive trajectory visualization, it was more difficult for the participants to identify an object. The test showed no significant differences between these two visualizations. According to the participants' comments, the objects in the video were too blurry when visualized by temporal blending. Some subjects also had problems with predictive trajectory visualization. They reported that the visualization provided too much information to focus on object identification. Nevertheless, they also mentioned that this visualization was useful in cases where only few objects

were present.

As outlined in Hypothesis 1, frame-skipping showed the best results in object identification and object trail visualization was able to produce similar results. However, the hypothesis that predictive trajectory visualization can achieve results comparable to object trail visualization has to be rejected. According to the median, temporal blending performed worst, but it was not significantly worse compared to predictive trajectory visualization.

Results of the Questionnaire. Figure 6.13 depicts boxplots that summarize the questionnaire results of task **T1**:

- **Subjective effectiveness.** The Kruskal-Wallis test shows that the visualization type had a significant influence on the rating ($H(3) = 35.66$, $p < 0.005$), and the U-test reveals significant differences ($p < 0.05$, Bonferroni-corrected) between all visualizations except for two pairs: frame-skipping / object trail visualization and predictive trajectory / object trail visualization. The boxplots in Figure 6.13 (top left) show that the subjective effectiveness and the measured objective effectiveness (see Figure 6.12) in object identification exhibit qualitatively similar results. However, in comparison to the other methods, the effectiveness of temporal blending was rated much worse than the objectively measured results indicate.
- **Usefulness.** The visualizations show significant differences (Kruskal-Wallis: $H(3) = 36.91$, $p < 0.005$) in terms of usefulness. Predictive trajectory visualization (median: 5.5) was considered significantly less useful than frame-skipping (median: 7.25, U-test: $p < 0.05$, Bonferroni-corrected). According to the comments of the participants, predictive trajectory visualization was only useful as long as a small number of objects were present in the scene. No significant difference was found between frame-skipping and object trail visualization (median: 7.0). Temporal blending was rated the least useful with a median of 4.0 and with significant differences to all other visualizations (U-test: $p < 0.05$, Bonferroni-corrected).
- **Comfort.** Significant differences (Kruskal-Wallis: $H(3) = 33.51$, $p < 0.005$) in the evaluation of comfort were found among the visualizations. Frame-skipping—as the familiar method to watch fast-forward videos—was considered the most comfortable to watch (median: 7.5). Frame-skipping performed significantly better than predictive trajectory visualization and temporal blending (U-test: $p < 0.05$, Bonferroni-corrected). A quite similar rating with significant difference to temporal blending was received for object trail visualization (median: 7.0; U-test: $p < 0.05$, Bonferroni-corrected). However, there was no significant difference between frame-skipping and object trail visualization. The least comfortable

Table 6.1 — Scores of the pairwise motion perception ranking (task **T2**).

	FS	TB	OTV	PTV	Sum
Frame-skipping (FS)	-	31	31.5	25	**87.5**
Temporal Blending (TB)	17	-	20	14.5	**51.5**
Object Trail Visualization (OTV)	17.5	28	-	13.5	**59**
Predictive Trajectory Visualization (PTV)	23	33.5	34.5	-	**91**

visualization was temporal blending (median: 3.5). It performed significantly worse than the others, except for predictive trajectory visualization (median: 5.5).

- **Effort.** There were significant differences (Kruskal-Wallis: $H(3) = 15.36$, $p < 0.005$, U-test: $p < 0.05$, Bonferroni-corrected) between frame-skipping (median: 6.75) / temporal blending (median: 8.5) and between object trail visualization (median: 6.5) / temporal blending. According to the medians, temporal blending demands highest effort, predictive trajectory visualization follows with a median of 7.25, but there were no significant differences between predictive trajectory visualization and the other three visualizations.

For the forced choice question in task **T1**, which visualization the participants prefer, 43 % answered with frame-skipping, 27 % preferred predictive trajectory visualization, 23 % object trail visualization, and 7 % temporal blending.

Results of the Motion Perception Task T2

A ranking of the pairwise compared visualizations was generated according to Kendall [161]: a visualization receives one point for each won comparison, and half a point for each draw. The resulting scores of the pairwise results are depicted in Table 6.1, summarized results in Figure 6.14. Following the test of Kendall [161] for paired comparisons with draws, the coefficient of agreement u shows a significant but relative low accordance between the participants ($u = 0.07$, $\chi^2(6) = 26.90$, $p < 0.05$).

Predictive trajectory visualization won most of the comparisons, as Hypothesis 2 has outlined, and received a score of 91. Surprisingly, frame-skipping is second with 87.5 points. Expected advantages of temporal blending (last rank, 51.5 points) in motion perception, and object trail visualization (third, 59 points) could not be confirmed.

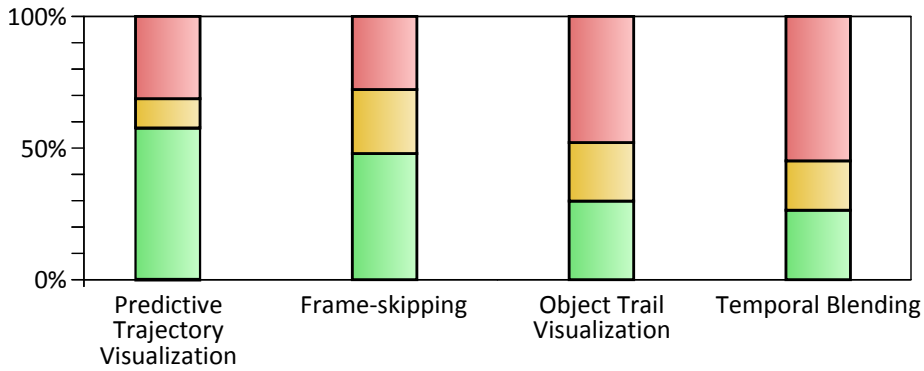


Figure 6.14 — Summary of the motion perception ranking (task **T2**). From left to right: The first, second, third, and fourth place of the ranking. Green/bottom: won; yellow/middle: draw; red/top: lost.

Results of the Playback Speed Feedback Task T3

The Kruskal-Wallis test showed no significant effects of the visualization type on the given answers. Figure 6.15 shows the results of the questionnaire for this task. Based on the participants' free comments and the medians, these results can be summarized with the following observations:

- **Effectiveness.** The participants attested analog VCR fast-forward the highest effectiveness in playback speed feedback (median: 8.0). Many participants explained their rating by their familiarity to such visualization from VCR. In terms of playback speed feedback, the speedometer also showed reasonable performance (median: 7.0).
- **Effort.** The participants rated analog VCR fast-forward also best according to the median in terms of effort while watching (median: 3.0). The effort required to keep track of the speed with the speedometer was higher according to its median (5.0). Following the comments of the participants, a reason for that originated from the frequent changes of focus between the video and the speedometer. The participants reported that the frequent color changes of the color frame visualization (median: 6.0) resulted in flicker, which led to stress.
- **Distraction.** The visualizations only marginally (according to the medians) differ in terms of distraction. The medians of the visualizations are 5.5 (speedometer), 5.0 (color frame visualization), and 4.5 (analog VCR fast-forward). The participants mentioned the frequent changes of focus between the video and the speedometer as a problem.

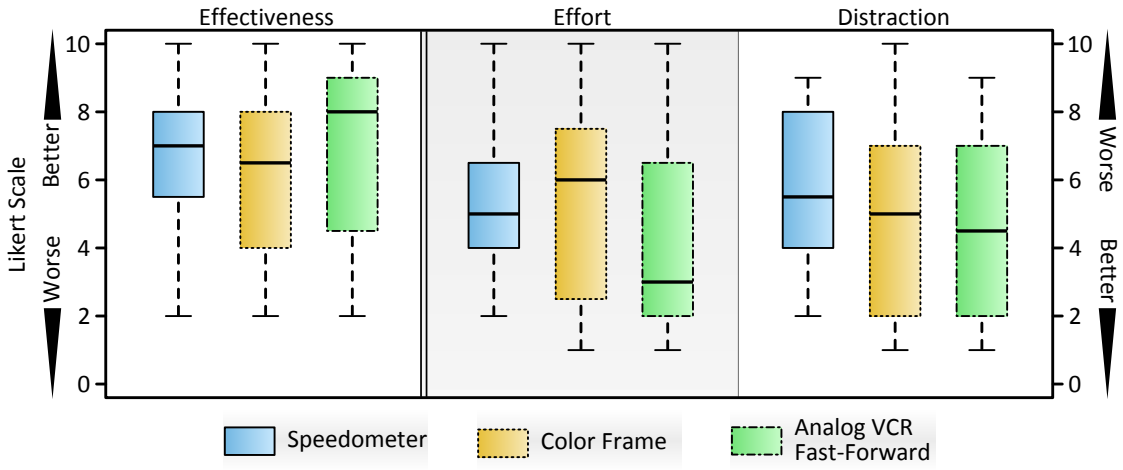


Figure 6.15 — Boxplots of the results of the playback speed visualizations (task **T3**).

For playback speed visualization, 40 % of the participants preferred the speedometer in the forced choice question, 36 % the analog VCR fast-forward, and 24 % the color frame visualization. As assumed in Hypothesis 3, the speedometer was the preferred visualization for speed estimation. However, the color frame visualization was not as good as expected, probably because of the induction of stress by flicker. It was remarkable that the issue of distortion and coarse playback speed feedback for analog VCR fast-forward was not considered that problematic as assumed. Nevertheless, the difference between the visualizations is relatively small, thus, no significant differences could be detected by Kruskal-Wallis test as mentioned above.

6.2.5 Conclusion

The user study showed some remarkable results concerning the performance in object identification of the different fast-forward visualizations. Frame-skipping was the preferred method for object identification and performed also well in motion perception. This result is contrary to the initial hypothesis that motion perception would be negatively affected by discarded frames. Temporal blending failed in both tasks: object identification and motion perception. It was especially surprising that most of the participants rejected this visualization also for the task of motion perception. The object trail visualization showed comparable results to frame-skipping in the object identification task. Nevertheless, the trail of an object did not improve motion perception. Although predictive trajectory visualization provides the most information on object motion (it was ranked best for this task), the additional information results in visual clutter and hinders object identification in crowded scenes. Hence, one recommendation that can be given is to choose either frame-skipping or predictive

trajectory visualization depending on the particular task. Due to the dependencies of the visualization on different characteristics of the video stimuli, the video visualizations should be evaluated in detail according to the amount of action (i.e., crowded vs. empty scenes) and according to different playback speeds, in future work. Especially the results of temporal blending indicated that this visualization depends on the acceleration factor; a reason could be that faster playback causes stronger blur. Another direction for future work is the adaption of predictive trajectory visualization according to the number of objects present in a scene in order to reduce visual clutter. Moreover, switching automatically to the most promising visualization, according to the current characteristics of video stimuli, may support users.

The results of the evaluation of playback speed visualizations for adaptive fast-forward can be summed up as follows: feedback of playback speed was considered best for the speedometer visualization. However, the participants of the study criticized the need to switch constantly the visual focus between the video and the speedometer. The color frame, in contrast, induced stress by visual flicker, which limits its application to cases of non-permanent usage. Analog VCR fast-forward seems to be the best solution to perceive speed feedback while watching video in adaptive fast-forward despite its coarse feedback. Especially in the context of playback speed feedback, the results of the study point out the strong benefit of using established metaphors in visualization. Although analog VCR fast-forward distorts the video signal and the speedometer visualization requires the switch of visual focus, their acceptance was surprisingly high.

The conducted user study focused on real-world videos that originated from surveillance cameras. For the first task, the video was superimposed by an artificial object, a cartoon figure, which should be identified. Although much effort was put in adapting the brightness and contrast of the objects to appear as normal persons, it is impossible to rule out completely the effects from this choice of stimulus. Therefore, future work should also feature real objects for the identification task. One direction for future work is also to generalize the results to entirely different video footage. Future directions for the playback speed visualizations include the evaluation of further solutions, such as a progress bar or a rotating bar. Such visualizations seem promising due to good motion perception in the peripheral field of view.

6.3 Interactive Schematic Summaries

This section presents a scalable method for exploratory video navigation utilizing the trajectories of moving objects: the *interactive schematic summaries* (ISS). The approach is suitable for exploratory browsing tasks, such as the inspection of video sequences with the objective to gain insight into structure, characteristics, and trends of object

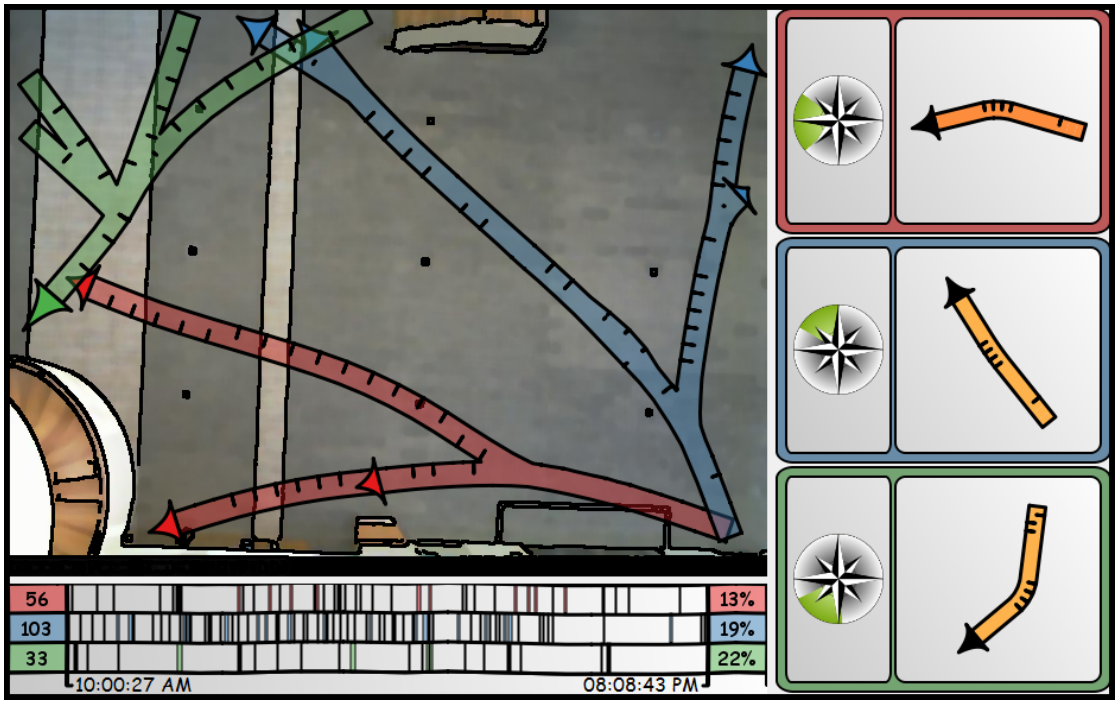


Figure 6.16 — Interactive schematic summary. Three clusters of trajectories are summarized. The arrows in the *spatial context view* (top left) indicate major paths. The *temporal context view* below displays the temporal coverage of the clusters (middle), the number of trajectories (left), and the diversity (right) of the clusters. The *facet showcase view* on the right depicts the following facets: azimuth[coverage] (partially filled wind rose), azimuth[mean] (directions of the arrow segments), and velocity[mean] (color of the arrows) of the cluster medoids. A legend of the visual mappings of the facet showcase view is provided in Figure 6.29.

movements. The ISS browsing technique is an example of the tight integration of visual data exploration and automatic data mining in the *visualization* stage.

In detail, the approach adopts *scatter/gather* browsing [70], which is known from document retrieval, and applies this method to trajectories extracted from video data. In the scatter stage, a set of trajectories is automatically arranged into a small number of coherent subsets according to the ensemble of features (i.e., the facets of the object movement) selected by the users. Users may select one or more subsets to be gathered (by union) into a new set of trajectories, which will be scattered again. By this iterative browsing process, users hierarchically refine their results or broaden them by navigating back to previous scatter levels. Users may also vary the facets selected for clustering to improve the browsing experience. According to Taylor et al. [276], facets are "clearly defined, mutually exclusive, and collectively exhaustive aspects,

properties, or characteristics of a class or specific subject". For video exploration, the users can select arbitrary combinations of the facets and similarity measures introduced in Chapter 4.1, which allow scattering the trajectory facets according to different occurrences. In this way, video sequences relevant to the users are easily retrieved and insight into the characteristics of the sequence is gained. This process is also closely related to visual information seeking mantra [262].

The subsets (i.e., clusters) are represented by a schematic visualization of clusters of trajectories that is efficient in providing an overview of movement characteristics in video (see Figure 6.16). The visualization is called *schematic summary* since it represents a "short summary" of clusters—expressed by a metaphor of scatter/gather it can be regarded as a "dynamic table of contents". The representation is schematic due to the simplifying way of visualization. Thus, the users are aware of a loss in accuracy caused by the reduction of complexity. The objective of these simplifications is to reduce visual clutter. With the same aim, cartoon rendering to video context information is applied, and the *trajectory bundling* method is developed to simplify cluster representation further. The presented visualization also comprises illustration of the selected features' distributions for each cluster. Amongst others, the visual language that is introduced is capable of communicating the characteristics of arbitrary combinations of facets and similarity measures while alleviating visual clutter. This gives users an overview of the homogeneity and feature constellation within a cluster.

In contrast to interactive query and filtering techniques (e.g., the methods introduced in Chapter 4 or by Aravecchia et al. [13]), the method is far more exploration-oriented and can be used as initial browsing step prior to query approaches. Compared to recent methods that use summaries for navigation [255], playback-speed adaption (see Chapter 5.1 and 6.2), or video synopsis [239], the approach allows much faster exploration of video data and offers a higher scalability due to hierarchical browsing. Hence, this video browsing technique closes the gap between these approaches.

A related approach that also uses clusters of trajectories has recently been presented by Lee and Bailer [185] in the context of media production. In their approach, trajectories are grouped using DTW. Clusters are represented by a set of video frames, each superimposed with a moving object's bounding box and trajectory. Navigation allows the selection of clusters and their re-clustering using the same or another feature. However, concise representation of the whole cluster and support for quick overview of the distributions of the features used for grouping the subsets is missing. This is a major focus of the work presented in this section.

6.3.1 Faceted Browsing Example

First, two exemplary browsing sessions using the ISS in typical video surveillance environments are described. The first example is based on video data provided with

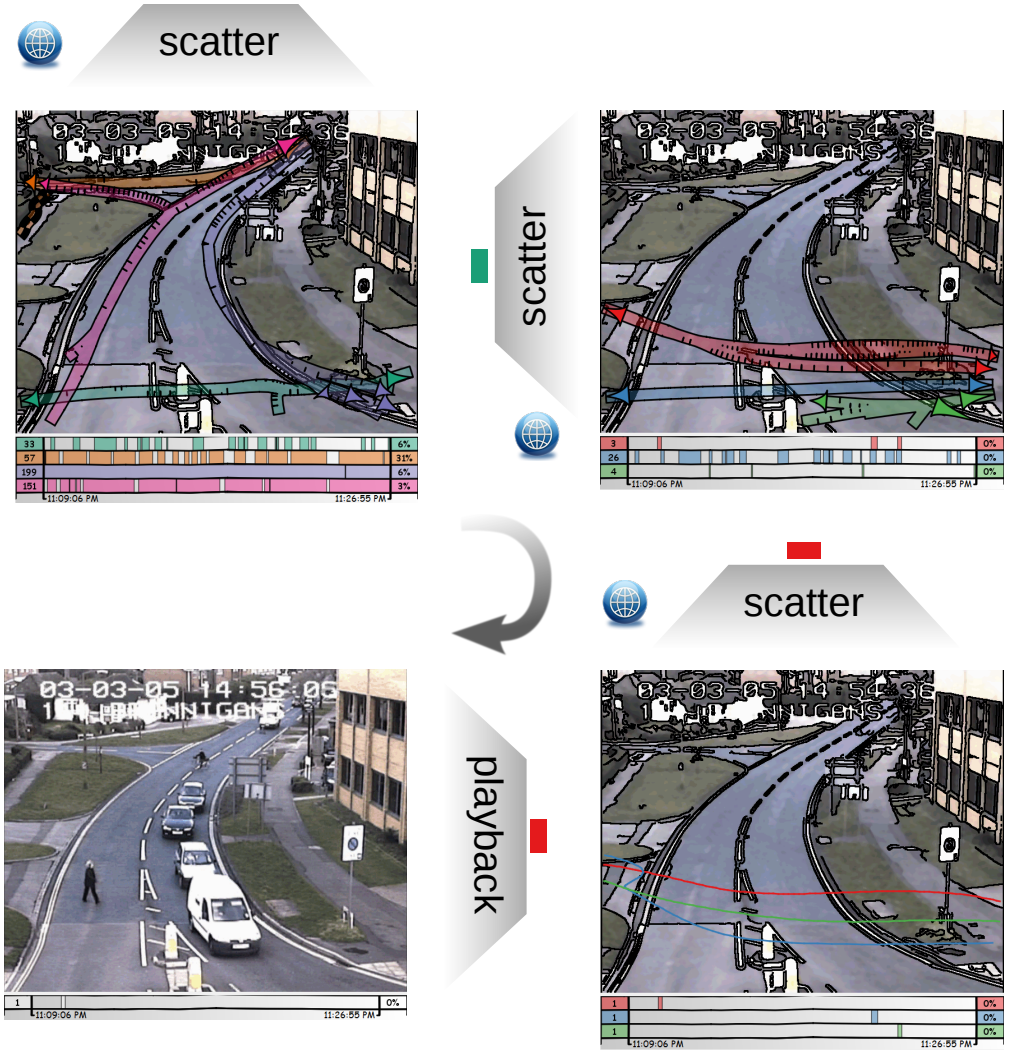


Figure 6.17 — Browsing example for a video sequence from the AVSS'07 *parked vehicle* challenge. Colored boxes (green, red, and red) indicate the cluster selection by users in the gather step. Trajectories of these clusters are scattered into new subsets. Symbols on the blue spheres illustrate that only the position facet was applied in the scatter step. Finally, users watch the period of the video associated with the selected trajectory in playback mode.

the AVSS'07 *parked vehicle* challenge[1]. It is an example of a structured environment because it mainly contains vehicles driving along the road. Since the regular behavior of traffic is known, users may identify relevant situations (e.g., dangerous maneuvers

[1] i-Lids dataset for AVSS 2007

or pedestrians crossing the street in undesired areas) easily. Figure 6.17 illustrates a browsing process that begins with an initial scatter of the trajectories into four clusters. Schematic arrow bundles outline the characteristics of the trajectories within the subsets. Context of the trajectories is provided by a schematic background image of the environment. The users feel comfortable with the number of clusters, since a general overview is well provided. The clusters exhibit the two main directions of road traffic and the movement of people crossing the street at the refuge island. The 4^{th} cluster (orange arrows) is to some extent disordered (junk cluster); this is indicated by its high diversity (depicted as percentage in the temporal context view). Based on prior knowledge, the users easily spot relevant movement. By selecting a cluster, its trajectories are gathered and immediately re-clustered into three new subsets with reduced number of trajectories. Hovering over a cluster highlights its schematic representation to improve separation in visually cluttered areas. After a few steps of iterative drill-down, the set of trajectories in which the users are interested is retrieved and playback of the according video sequence is initiated.

The second example is based on the *Edinburgh Informatics Forum Pedestrian Database* [205]. It contains 1975 trajectories captured on June 4^{th}, 2010 within the *Informatics Forum* (a rather unstructured indoor environment without predefined movement paths). This example illustrates a characteristical browsing session of users that want to find out what happened that day.

A first scatter of the trajectories into six clusters provides a coarse overview of the movement of people during that day (see Figure 6.18 (a)). Since the users choose a combination of position and azimuth facets, the complete set of trajectories is split into subsets according to their main directions. Hence, often used paths and hot spots are unveiled by the visualization. The facet visualization on the right, next to the spatial context view, shows the medoid's direction of each cluster. In this example, the users are interested in movement toward the entrance/exit of the building in the lower left corner. They set the number of clusters for the next browsing step to four. By selecting the cluster representatives (purple, green), their trajectories are gathered and re-scattered into four new clusters. The trajectory distribution depicted in the temporal context view, below the schematic summary of the spatial context view, exhibits that most people left the building between 12:30 pm and 7:00 pm that day. The spatial context view further reveals the main paths of people leaving the building. Roughly speaking, the people coming from the upper levels of the Informatics Forum split into a group taking the stairs (mainly orange cluster) and another group taking the elevator in the upper right part (green cluster). According to the trajectory distribution depicted in the temporal context view, one can roughly estimate that the first group is about three times as big as the group taking the lift.

The users now navigate back to the initial scatter representation and re-scatter the whole dataset into six new clusters considering another facet of the trajectories: their

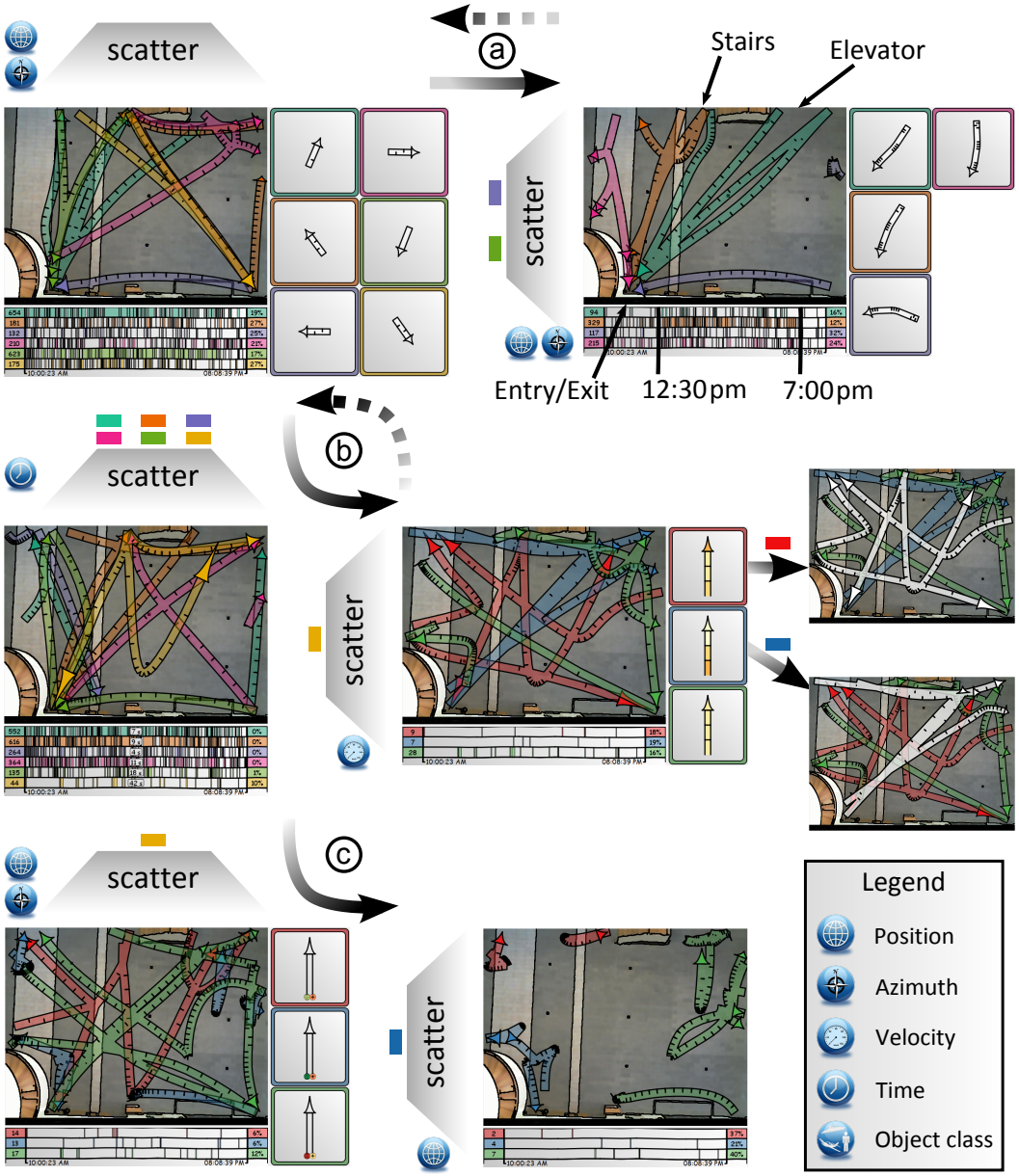


Figure 6.18 — Browsing example using the *Edinburgh Informatics Forum Pedestrian Database* [205]. Colored boxes indicate the cluster selection by the users in the gather step. Trajectories of these clusters are then scattered into new subsets and displayed with new colors. Symbols on the blue spheres illustrate the facets applied in each scatter step. The users first navigate in the direction marked with (a). After returning to the initially scattered layout, they re-scatter the whole dataset displaying another facet. They then browse into the branch marked with (b) and finally navigate into the last branch, labeled with (c).

temporal standard deviation. Hence, the dataset is clustered with respect to the duration of the trajectories. After gathering the cluster of trajectories with the longest lifetime, the users re-scatter them into three clusters regarding their velocity. Summarized into four segments, the colors of the arrows in the facet showcase view depict the velocity of the trajectories in each cluster. Together with the schematic summaries of the clusters, these velocities exhibit three typical classes of movement behaviors of trajectories with long duration: slowly moving people (green cluster), people that initially walk fast, but have to wait for the lift afterward (blue cluster), and people walking fast, then slowing down while they change their direction and finally head on fast again towards their new destination (red cluster). To improve understanding, highlighted selections of the blue and red clusters are also depicted in Figure 6.18 (b), which are accessible to the users by mouse hovering.

To gain further insight into the movement of people that produced the 44 trajectories with long duration, the users navigate back to the view of the dataset scattered with respect to the time facet. Next, they gather the trajectories of the dark yellow cluster and scatter them into three clusters using position and azimuth facets with the *distance between standard deviations* similarity measure (see Chapter 4.1) as depicted in Figure 6.18 (c). The colors of the glyphs in the facet showcase view represent the extent of positional and directional deviation of the trajectories (the applied color-coding is depicted in Figure 6.29). The users decide to gather the trajectories of the blue cluster, which have low positional deviation. Scattering these trajectories into three clusters according to their position exhibits that these tracks are mainly produced by people waiting or loitering in different areas of the Informatics Forum.

6.3.2 Trajectory Clustering

The proposed method requires trajectories that are extracted from video according to Chapter 3.2. The trajectory information can be preprocessed (or buffered as static time window for stream processing) to avoid negative effects on the responsiveness of interaction (see also the discussion in Chapter 2.2 for the potential conflict of sequential processing of data and aggregation techniques). The *Edinburgh Informatics Forum Pedestrian Database* already provides trajectory information.

For clustering, the approach allows the users to choose from three different similarity measures for each facet of an object's movement, which are both discussed in detail in Chapter 4.1. The similarity measures are applied in the clustering process to calculate the distances between trajectories. As mentioned in Chapter 4.1, the provided similarity measures are *coverage*, *distance between means*, and *distance between standard deviations*; the supported facets are *position*, *velocity*, *azimuth*, *time*, and *object class*. The default browsing mode allows the users simply selecting the facets for clustering. If users demand for higher control for scattering, additional details of the similarity measures used

Figure 6.19 — Facet selection interface. Large symbols on the left depict the different facets (legend of the symbols depicted in Figure 6.18). Further details on the similarity measure (i.e., type and granularity) are available after selection of a facet. The green spheres illustrate selected facets. In this example, position facet using the coverage measure is selected together with the all three similarity measures of the velocity facet.

for the respective facet can be switched on, such as the type of similarity and its level of detail (i.e., the granularity κ). By interactively controlling these parameters, users are able to formulate comprehensively their intensions for re-scattering. For example, if users want to have the main paths clustered, they can choose position[coverage]. If the clusters should also consider the movement directions, they could add azimuth[mean] with $\kappa = 1$. Even explorations that are more sophisticated are possible: clustering loitering people is possible by combining position[standard deviation] (low for loitering people) and azimuth[coverage] (high), for example. The user interface for the selection of facets and similarity measures is depicted in Figure 6.19.

The similarity measures are finally utilized to cluster the trajectories in each scatter/gather iteration according to the selected facets. To achieve responsive interaction, the approach can process all stages offline except for clustering. Since clustering is not in the focus of this thesis, only a brief discussion of the methods is provided in the following.

The responsiveness of the proposed video browsing method is sensitive to the choice of the trajectory clustering approach. Due to re-clustering in each scatter/gather iteration, time complexity of the clustering approach becomes relevant. Another requirement the clustering method has to meet is the possibility to specify the number

of clusters. This way, users can choose the number of subsets according to prior knowledge about the scene as well as control the amount of visual clutter produced by the cluster representations. Partitional clustering algorithms, such as k-means, meet these requirements and are efficient in clustering large-scale data sets [313]. For example, k-means has a time complexity of $O(k \cdot |T|)$ (with $|T|$ being the number of trajectories). Thus, it is favored over hierarchical clustering algorithms, which also allow for manual specification of the number of clusters, but typically exhibit a time complexity of $O(|T|^2)$. Additionally, partitional clustering approaches, such as k-means, allow for pre-calculation (prior to browsing) of the trajectory distance matrix, in contrast to hierarchical clustering methods, which normally require cluster-to-cluster distances to be calculated within clustering. The proposed method applies the k-medoids approach, which is related to k-means, but differs in using a data element (i.e., a trajectory) as cluster representative instead of the mean value in feature space. In executed tests, k-medoids took 47 ms for the 1975 trajectories of the dataset from the second example (single threaded on an Intel i7-2600K CPU, time without determining medoids (also linear complexity [194])).

Several enhancements of clustering algorithms have been developed for the purpose of scatter/gather browsing in the context of text document retrieval. Cutting et al. [70] came up with a combination of hierarchical and partitional clustering for scatter/gather, called *Buckshot*. The idea was to use the hierarchical approach on a small but representative subset of the data, to generate appropriate seeds for the fast partitional method. Fast online clustering algorithms for scatter/gather browsing have recently been proposed by Liu et al. [194] and Ke et al. [156]. They achieve a fast cluster generation in online phase by utilizing a cluster tree that they calculated in an offline phase by hierarchical clustering. Further information about clustering of trajectory data is provided by the comprehensive survey of Liao [191].

6.3.3 Schematic Visualization

For interactive scatter/gather video exploration, the visualization of the cluster information is fundamental. First, an appropriate visual representation of the clusters is crucial for scatter/gather browsing since users base their decisions and further browsing intentions mainly on this visual information. Besides this, context information has to be communicated to enable users to interpret events correctly. Temporal information of the trajectories in the clusters is also relevant since video data is inherently time-dependent. Moreover, the visualization should unveil the facet characteristics of each cluster selected by the users in the current scatter stage. To meet these requirements, the approach introduces the schematic summaries consisting of three linked views:

1. the *spatial context view* (see Figure 6.16 top-left);

2. the *temporal context view* (see Figure 6.16 bottom-left);
3. and the *facet showcase view* (see Figure 6.16 right).

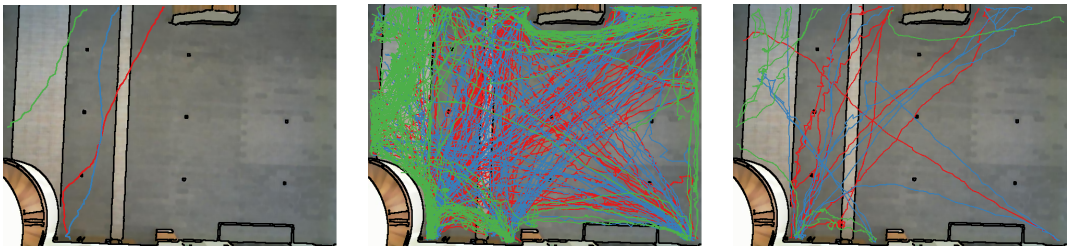
Hovering with the mouse over a cluster in any view highlights the cluster in the others. Further, the clusters can be gathered in each view.

A visualization that convey all available information in detail would be inadequate for the exploration task. For example, visualizing all trajectories of a cluster would result in immense occlusion and visual clutter, and thus, hinder users to gain insight (see Figure 6.20 (b)). Instead, a novel simplifying visualization is proposed that is capable of reducing visual clutter while highlighting the main characteristics of the video. Schematic rendering techniques are employed to emphasize the sketchy nature of the visual cluster representation.

Cluster Representation

A suitable visualization of trajectory clusters is the key element of ISS. For this reason, a novel illustration of trajectory clusters is proposed in this section. Before describing the rendering method, some general requirements of cluster representatives and their visualization are discussed, and it is shown how the approach copes with these requirements.

The major objective of cluster representatives is to convey the structure of the trajectories. The representation should be small-sized to avoid occlusion with illustrations of other clusters. Visualizing only the medoid of the cluster minimizes visual clutter but does not reveal the whole spectrum of trajectories of the cluster (see Figure 6.20 (a)). Anyhow, the medoids are the most important representatives of the cluster as the facets of each trajectory in the cluster were compared with the facets of the medoid for clustering. For that reason, the proposed method shows the medoid in the facet showcase view. In contrast to visualizing only the medoid, drawing all trajectories of a cluster leads to occlusion of trajectories of the same cluster and of others. Such visualization hinders interpretation of clusters (see Figure 6.20 (b)). By selecting a set of representative trajectories for the spatial context view, the approach conveys a general overview while visual clutter is kept low (see Figure 6.20 (c)). Nevertheless, the visual representation of Figure 6.20 (c) is still prone to occlusions and difficult to understand (which will lead to *trajectory bundling* that is introduced later). Additionally, the quality of the summary depends on the trajectory selection for cluster representation, and users may pursue different goals, therefore. According to the analytical task, one may be interested in understanding the main routes and hence ignore outliers. In other cases, users may care more about the diversity of trajectories and would like to select clusters with divergent content. To satisfy both kinds of goals, users can choose if trajectories should be selected randomly (i.e., uniformly distributed) or in a way that



(a) medoids (b) all (c) selection

Figure 6.20 — The number of trajectory representatives is a trade-off between visual clutter and completeness of representation.

maximizes dissimilarity of representatives. Additionally, the approach gives the users control over the number of representatives depicted.

Another requirement of the cluster representation is to convey the directions of the trajectories. Directions are not only relevant if the azimuth facet is chosen. They are also implicitly considered for the position facet, if distance between means is chosen with more than a single segment, i.e., we have a temporally ordered set of positions. An example is shown in Figure 6.20, where the main difference between the red and the blue clusters is that the respective trajectories point in opposite directions. Moreover, direction information is important to interpret the features in the showcases correctly: the users should be provided with the information which part of the trajectory represents the beginning and which represents the end. Typically, directions of trajectories are indicated by arrows [291].

Finally, the visual representation should meet users' expectations and be easy to understand. A cartoon-like illustration serves this purpose, because users are already familiar with it. It has also been demonstrated that such visualization communicates the content of video effectively [112]. Winnemöller et al. [307] showed in a user study that such abstract visualization noticeably increases the recognition speed and additionally enhances the memory ability of users. Further information about artistic and illustrative stylization is provided in the textbook by Strothotte and Schlechtweg [270].

Spatial Context View

The *spatial context view* has two objectives. First, it has to provide an overview of the spatial distribution of trajectories inside the different clusters. The second goal is to convey spatial context information. The spatial context view is always visible, regardless which facets were chosen for the scatter step.

To create a schematic summarization of trajectories that provides a good overview

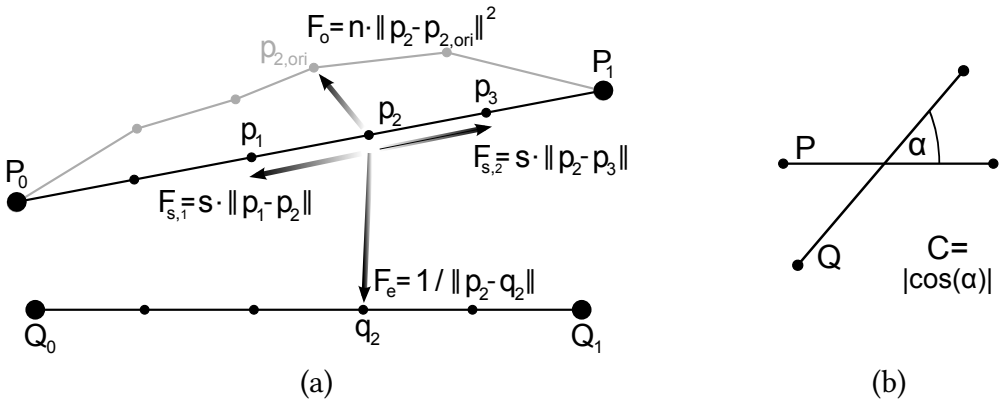
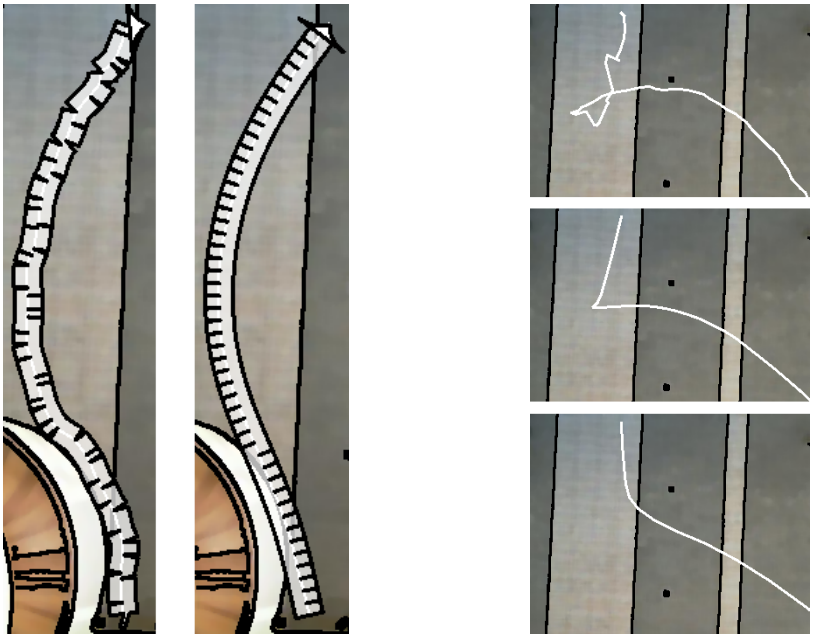


Figure 6.21 — Forces affecting trajectories P and Q. (a) The spring force F_s (made up of $F_{s,1}$ and $F_{s,2}$), one fragment of the electrostatic force F_e between sample point p_2 and q_2 (without its constant e), and the original location affinity force F_o exerted on sample point p_2. The gray spline illustrates the original trajectory P prior to exertion of forces. (b) Angle compatibility C between trajectories.

and prevents visual clutter, the approach uses a rendering algorithm consisting of two stages. In the first stage, trajectories of a cluster are simplified and homogenized by a novel method termed *trajectory bundling*. In the second stage, an arrow representation is rendered for the bundled trajectories.

Trajectory Bundling. As mentioned above, the visual representation of a cluster is still subject to occlusion. Figure 6.20 (c) reveals positional variations for rather similar trajectories (e.g., green trajectories in the top left corner). Small variations of their traces increase visual clutter and cause overdrawing and occlusion of video context information. In browsing scenarios, getting an overview is often more important than the retrieval of particular details. For instance, a small variation of any trajectory in Figure 6.20 does not change the perception of the prominent paths that the users observe. For this reason, this section introduces trajectory bundling, a novel visual approximation of the cluster that reduces the area required for its visual representation. This enables the user to understand main characteristics and prevents visual clutter.

The trajectory bundling technique is inspired by *force-directed edge bundling* (FDEB) [142], developed to reduce edge clutter of general graphs. Basically, FDEB uses a physics model to shift subdivision points (additionally introduced points on an edge) iteratively into the direction of a force. This force is made up of the *spring force* F_s, the *electrostatic force* F_e, and four edge compatibility measures. For trajectory bundling, the approach adapts these forces and extends them by the *original location affinity force* F_o. The exerted forces on a sample point of a trajectory are sketched in Figure 6.21 (a).



(a) smoothing (b) removal of outgrowths

Figure 6.22 — Effect of spring force F_s: (a) smoothing for schematic arrow rendering, (b) removal of outgrowths to reduce visual clutter.

The spring force was originally used to control the stiffness of an edge [142]. Trajectories are unlike edges in a graph no straight lines, but can describe arbitrary paths of moving objects. These paths suffer from jitter introduced by inaccuracies of the tracking algorithm. The spring force adapted for trajectory bundling smooths each single trajectory (important for schematic arrow rendering, see Figure 6.22 (a)) and simplifies them by removing outgrowths that would lead to visual clutter (see Figure 6.22 (b)). The spring force of a sample point p_i is defined as: $F_s = s \cdot (\|p_{i-1} - p_i\| + \|p_i - p_{i+1}\|)$, where s represents a constant to control the extent of smoothing and outgrowth reduction. In contrast, FDEB utilizes a spring constant to control stiffness that depends on the number of subdivision points. Please note that the spring force the approach uses has completely other objectives than the spring force used in FDEB. The counterpart to the spring force of FDEB in trajectory bundling is the original location affinity force. The proposed method defines this force as $F_o = n \cdot \|p_i - p_{i,ori}\|^2$, where n is a constant to control the affinity to the original location and $p_{i,ori}$ is the original location of sample point p_i. This force controls the stiffness according to the properties of trajectories (i.e., the arbitrary original shape in contrast to an originally straight edge). The original location affinity force is used to balance the impact of the other forces (F_s and F_e) to maintain the qualitative appearance of the original trajectory. The force F_o shows quadratic growth with the distance between the original position of a sample point and

its position of the current iteration. Hence, the extent of deformation is attenuating and the characteristics of the trajectory are preserved.

The electrostatic force F_e is responsible for bundling of trajectories: sample points of different trajectories attract each other in spatial vicinity. The proposed electrostatic force is similar to the one defined in FDEB. They define it as $F_{e,FDEB} = \sum_{Q \in E} \frac{1}{\|p_i - q_i\|}$, where E is the set of all edges without P. In FDEB, a subdivision point p_i is influenced only by points with the same subdivision index i. This approximation does not differ much from a visual point of view in the context of edges [142], but has a strong influence in case of trajectories. Due to complex traces of trajectories, sample points with different indexes are often closer to each other than the pair of sample points with the same index. This might stem from different numbers of samples or opposing directions of the trajectories. Therefore, the electrostatic force for trajectory bundling should consider all sample points of other trajectories. The approach defines the electrostatic force for trajectory bundling as $F_e = e \cdot \sum_{q \in T} \frac{1}{\|p_i - q\|}$, where e is a constant to control the attraction of trajectories and T is the set of all sample points of trajectory representatives in a cluster excluding the sample points of P.

A couple of edge compatibility measures were introduced in FDEB to control the amount of interaction between edges according to the relations between the edges. For trajectory bundling, only the angle compatibility measure is appropriate. In general, parts of trajectories that are mostly parallel should be bundled together and parts almost perpendicular should not affect each other. Therefore, the approach adopts the angle compatibility measure defined as $C = |\cos(\alpha)|$, where α is the angle included between two trajectories (see Figure 6.21 (b)).

The fusion of the aforementioned forces with the compatibility measure leads to the resulting force exerted on sample point p_i at each iteration step:

$$F_{p_i} = F_s + F_o + \frac{C}{F_e} = s \cdot (\|p_{i-1} - p_i\| + \|p_i - p_{i+1}\|) + n \cdot \|p_i - p_{i,ori}\|^2 + e \cdot \sum_{q \in T} \frac{|\cos(\alpha)|}{\|p_i - q\|} \quad (6.2)$$

The start and end points of the trajectories are handled separately (the nodes of FDEB are not affected at all). Applying the resulting force F_{p_i} to start and end points would lead to traction of all trajectories to the barycenter, which is typically near the center of the image. Nevertheless, similar start and end points need to be merged, too: although objects enter (or leave) the scene through the same door, the origin of their trajectories may vary to some extent. The approach accounts for this issue by applying modified forces to start and end positions: $F_{p_{s/e}} = F_o + e_{s/e} \cdot \sum_{q \in T_{s/e}} \frac{|\sin(\beta)|}{\|p_{s/e} - q\|}$, where $T_{s/e}$ is the set of all start and end points of trajectories in the cluster, $\sin(\beta)$ is a modified angle

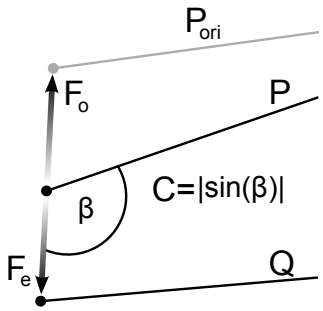
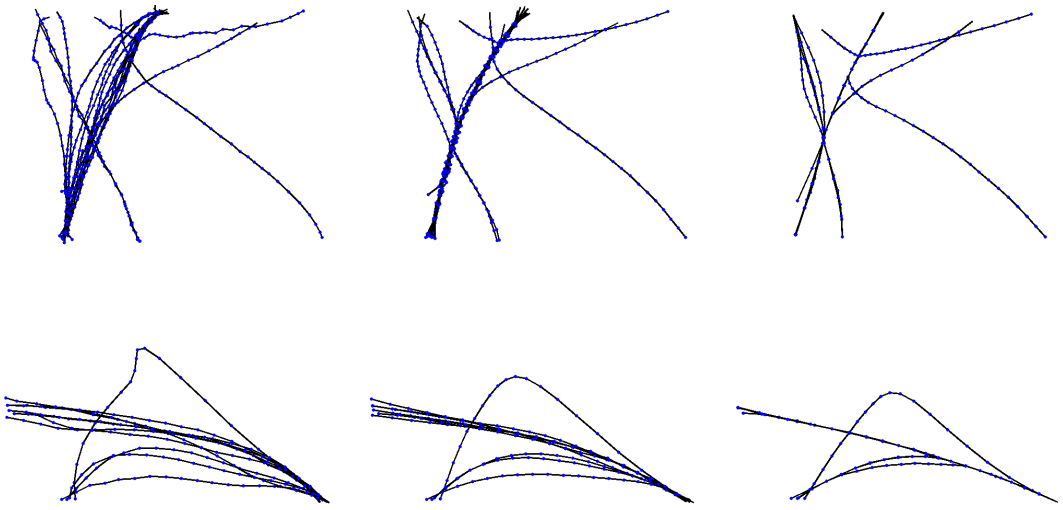


Figure 6.23 — Forces and compatibility measure applied to start and end points.



Iteration 0 Iteration 15 Iteration 150

Figure 6.24 — Results of trajectory bundling. The upper row shows a cluster with high diversity and 20 representatives. The lower row displays a rather homogeneous cluster with 10 representatives.

compatibility for start and end points (see Figure 6.23), and $e_{s/e}$ is a constant to control the attraction of the start and end points.

The proposed method also simplifies the original iterative refinement scheme according to the properties of trajectories. In each of the I iterations, it calculates the forces, multiplies them by a relative step size s_i, and then applies them (resulting unit: pixels) to the sample points. The relative step size at iteration i is defined as $s_i = 1 - \sqrt{\frac{i}{I}}$ and is decreasing for convergence. Visual results of trajectory bundling are illustrated in Figure 6.24. Additionally, the area coverage of cluster representations is measured, which serves as indicator for visual clutter and the potential of overdrawing of multiple cluster representations. Trajectory bundling is evaluated with the *Edinburgh Informatics Forum Pedestrian Database* dataset used in the second browsing example. This is the same dataset used to display the visual results in Figure 6.24. For evaluation, the

trajectories are rendered by splines with a thickness of 2 pixels (similar to the representation of trajectories in Figures 6.20 and 6.24) and the area covered by this visualization is checked to measure visual clutter reduction on screen. On average, the evaluation shows a decrease of area covered of more than 38 %, if trajectory bundling is applied (configuration of bundling forces similar to those used for Figure 6.24). Only a marginal dependence between the maximal number of cluster representatives and the average area coverage was experienced. Further, area coverage is improved by more than 50 % for clusters with at least three representatives (> 70 % with seven representatives). This originates from the fact that nearby similar parts of trajectories are visually merged together. Even about 10 % of drawing area was saved when simplifying merely a single trajectory by the proposed bundling approach. Further, an increase of the trajectory bundling performance with increasing positional homogeneity of trajectory clusters was experienced. This effect especially arises in later browsing iterations (using the position facet), since cluster diversity is reduced every scatter/gather step.

Schematic Arrow Rendering. After applying trajectory bundling to the cluster representatives, the approach renders a visual cluster representation in cartoon style. For this purpose, the trajectories are combined into one or more arrows that can have multiple start and end points. For rendering, the painting routines of the Qt library[7] are used. Some results are displayed in Figure 6.25.

Arrowshaft Rendering. The arrowshaft is rendered by successively unifying rectangular segments, one for each pair of adjacent sample points. Only the final segment of the trajectory is excluded and will be rendered as arrowhead. The orientation and height of the rectangles are defined by the according pair of sample points. The width of the rectangle is predefined and set to 10 pixels in the examples. By contracting the length of the rectangles, gaps are created. These gaps produce hatchings in the bending area because of a black contour of the rectangles. The arrowshaft is rendered semitransparent in order to preserve underlying information from other clusters and background context information. For better distinction, clusters are drawn in different colors.

Arrowhead Rendering. Arrowheads are attached to the arrowshafts to convey direction information. Here, the approach first merges end segments of trajectories that are in close vicinity to avoid occlusion of arrowheads of the same cluster. The resulting arrowhead points into the mean direction and may increase in width and length according to positional variations of the end segments.

Providing Context Information. Spatial context information is important to interpret trajectories and their clusters properly. Therefore, the approach provides video

[7] http://qt.digia.com/

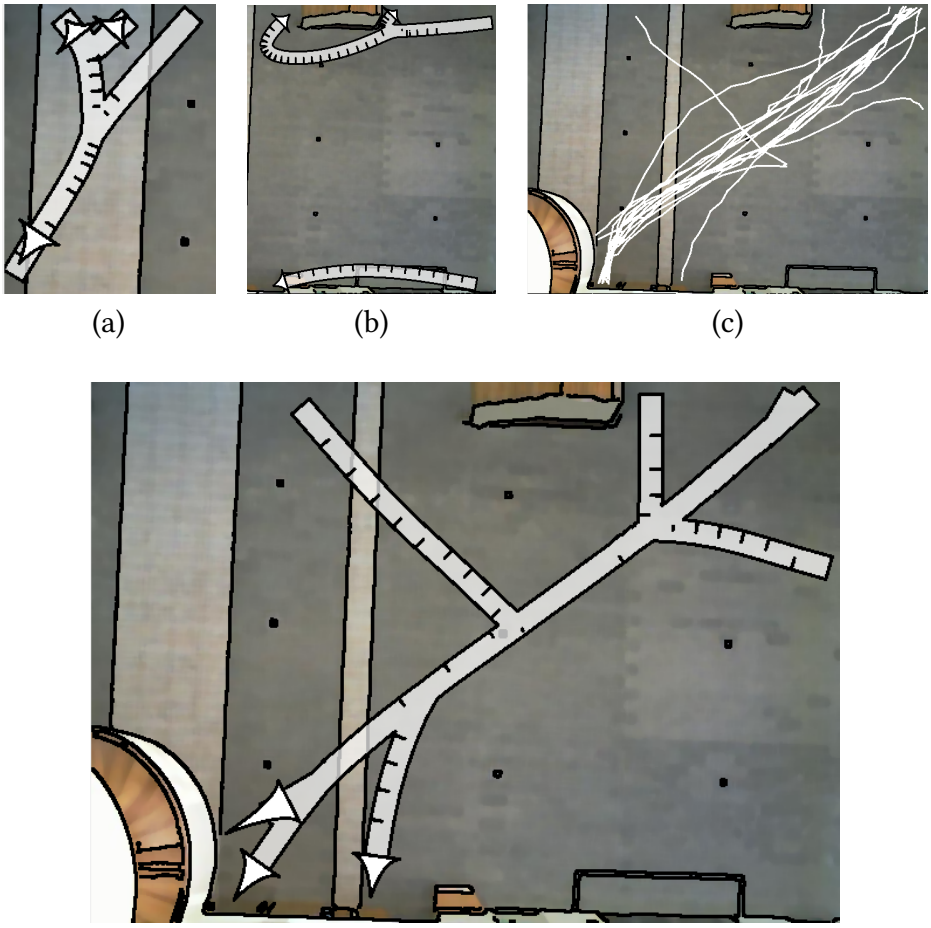


Figure 6.25 — Cluster representation by schematic arrow rendering. (a) If locations act as start and end points, the tail of an arrow is combined with an arrowhead. (b) Two unconnected visual cluster representatives. (d) Visual cluster representation of (c). Different start points merge into a main path and split again at the end.

context information in terms of a background image calculated from several frames of the video. The interpretation of abstract positions as well as paths change if an image of the background (e.g., a keyframe of the video) is provided. Often, the background image includes many details that are not required to understand the scene, and thus, becomes cluttered. Such detail may negatively affect the perception of foreground objects or might draw the attention of users. This might confuse users, especially if foreground objects are drawn semitransparent or overlap with each other. Background images with low contrast pose additional difficulties: important regions may become irrecognizable by semitransparent overdrawings. To account for these issues, the method creates an

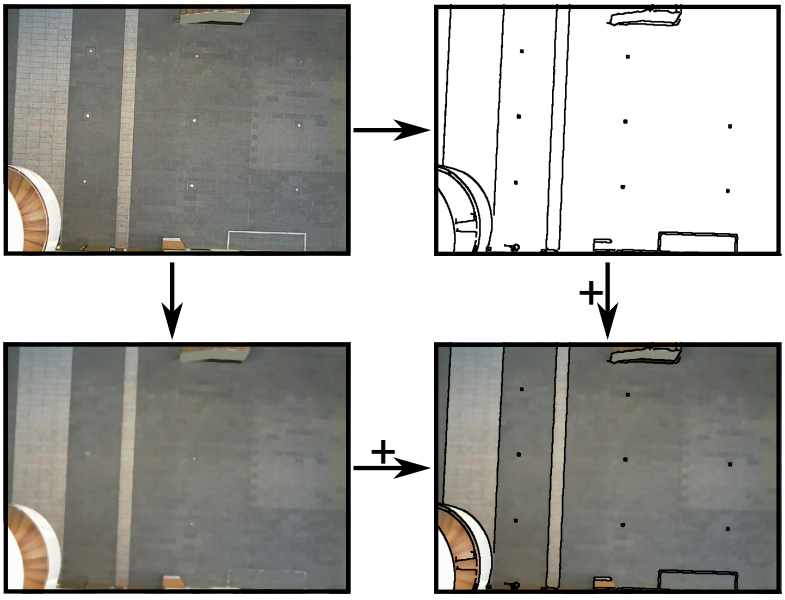


Figure 6.26 — Schematic background generation.

abstract background image and reduces the visual load of areas with high frequency, but highlights distinctive areas for better orientation.

In detail, the schematic background is made up of two images that both are derived from the background image of the background subtraction model. The first image is created by smoothing the background image with a Gaussian kernel (9×9). The second image contributes distinctive edges to the result. For this purpose, the approach uses a smoothed version of the background image, too. Then, a Canny edge detector is applied and these edges are emphasized by morphological dilation. Finally, both images are combined to the schematic background, see Figure 6.26.

Temporal Context View

The objective of the temporal context view (see Figure 6.27) is to complement the other views with temporal information. The spatial context view described above, as well as the facet showcase view do not communicate temporal information. Therefore, the timelines are attached as a linked view to complement the ISS. Each timeline shows the temporal coverage of all trajectories belonging to a particular cluster. For correspondence visualization, the timeline is also drawn in the cluster's particular color, which is used to draw the arrows in the spatial context view. Besides the period of all trajectories, the time slots of the single trajectories are illustrated. To further support users' orientation and navigation, the approach displays the number of trajectories within a cluster as well as the divergence of the cluster. In the same way as the spatial

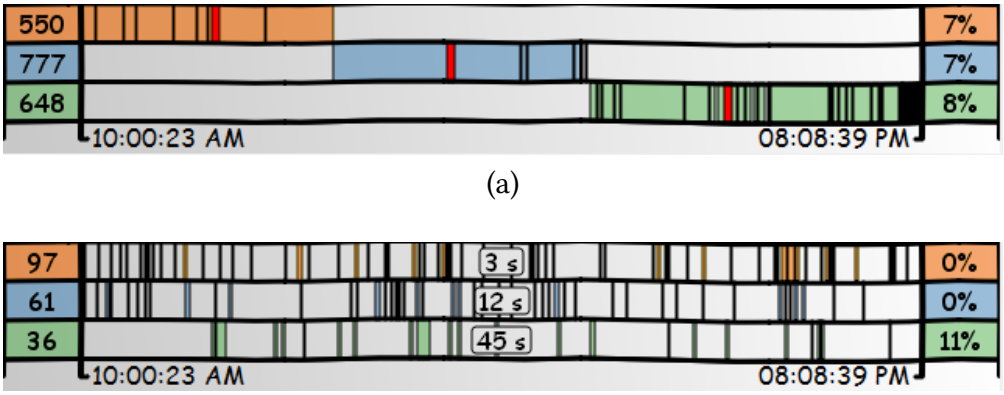


Figure 6.27 — Temporal context view. For each cluster, information about the number of trajectories (left), their temporal coverage (middle), and the cluster's diversity (right) is provided. In this example, the trajectories are clustered according to the time facet using the distance between means (a) and the distance between standard deviations measure (b). (a) The bright red bars depict the temporal occurrence of the medoid trajectories. (b) Labels show the duration of the medoid trajectory of each cluster.

context view, the temporal context view is always displayed, regardless of the selected facets.

If users choose the time facet for scattering, the temporal context view will be augmented with the corresponding facet information. An example of the temporal context view that shows trajectories clustered according to their temporal occurrence (i.e., time[mean]; shown as red bars in the timelines) is illustrated in Figure 6.27 (a). Figure 6.27 (b) shows another example with trajectories clustered with respect to their lifetimes (i.e., time[standard deviation]; shown as label).

Facet Showcase View

The third component of the schematic summaries is the facet showcase view. It communicates important information on facet characteristics of each cluster, with respect to the facets and distance measures chosen for scattering. While some facet characteristics are visualized best in the corresponding spatial or temporal context views (and are therefore only displayed there), visual clutter and overdrawing that would originate from additional facet information in these views demand for a separated showcase view.

To account for visual clutter, the method displays additional facet information of the medoid of each cluster in this separate showcase view. Only those facets (with respect

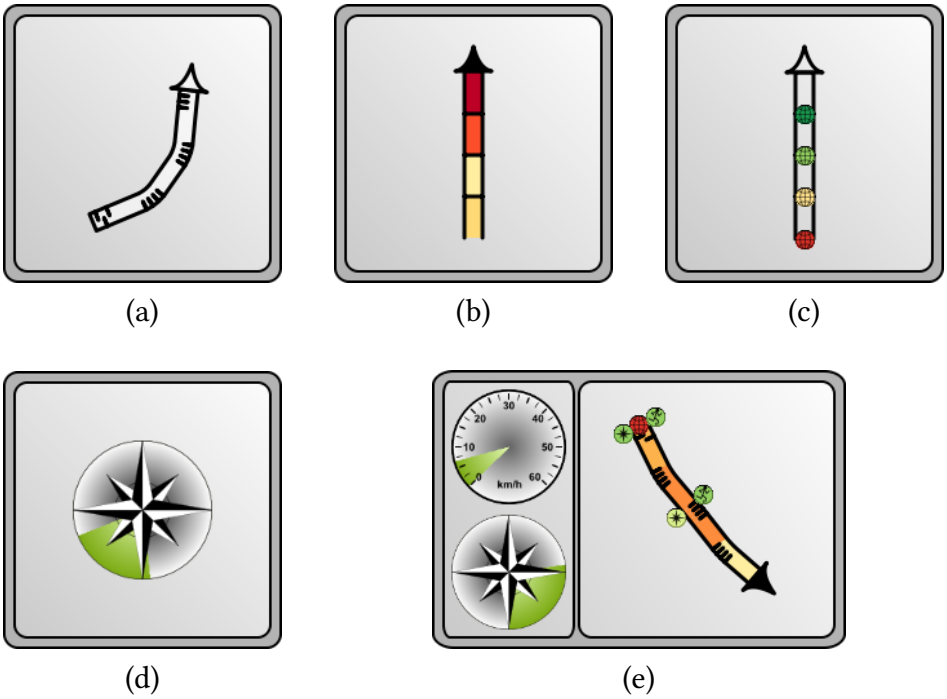


Figure 6.28 — Facet visualization: (a) azimuth[mean] with 4 segments; (b) velocity[mean] with 4 segments; (c) position[standard deviation] with 4 segments; (d) azimuth[coverage]; (e) combination of velocity[coverage], azimuth[coverage], velocity[mean] (3 segments), azimuth[mean] (4 segments), position[standard deviation] (1 segment), azimuth[standard deviation] (2 segments), and velocity[standard deviation] (2 segments).

to their similarity measures) of the medoid's trajectory that were used for scattering are shown. This provides appropriate feedback of the trajectories and their facet characteristics of each cluster.

If only those facets are selected that are not visualized in the showcase view, the whole view is hidden. Depicting additional facets and/or additional similarity measures of facets that were not chosen by the users would mislead them: they may assume that other trajectories inside this cluster share those features, too.

Figure 6.28 illustrates a couple of examples of showcases representations of different facets and their similarity measures. The showcase representation of azimuth[mean] is a showcase-aligned arrow that is bent according to the mean direction(s) of the cluster's medoid (see Figure 6.28 (a) and (e)). Figure 6.28 (b) shows velocity[mean] by color-coded inking of the segments of a trajectory. Here, azimuth[mean] is not depicted and the trajectory is simply drawn from bottom to top. The similarity measure distance between standard deviations is depicted by color-coded glyphs attached to the

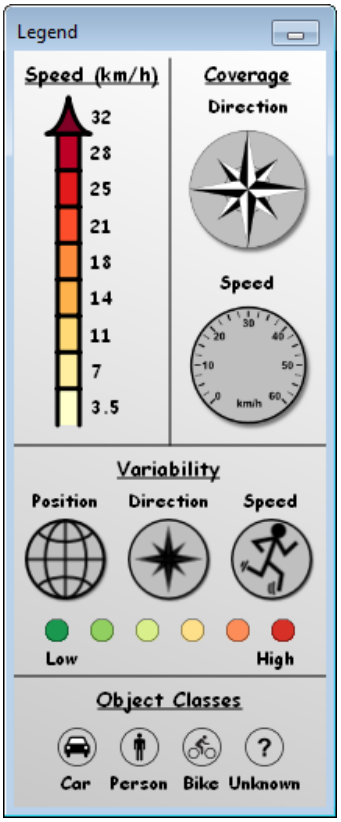


Figure 6.29 — Legend for the facet showcase view. In the top left, the color mapping of the velocity facet used for the distance between means measure is depicted. Templates, which will be filled according to the coverage of the azimuth and velocity facets respectively, can be seen in the top right. Glyphs and color mappings used for the distance between standard deviation measures (variability) of the three facets position, azimuth, and velocity are illustrated in the middle. The lower row shows the icons of the object classes that are present and extracted for the current video data.

trajectory visualization. Figure 6.28 (c) shows the augmented trajectory with glyphs for position[standard deviation] at the particular segments. Position[coverage] and azimuth[coverage] are visualized by inking covered ranges on a speedometer and a compass rose, respectively (see Figure 6.28 (d) and (e)). The facet object class is communicated by icons superimposed in the showcase. A comprehensive legend of the mappings can be found in Figure 6.29. The visualization of the facets is flexible and capable of displaying all different facets and similarity measures at once, without suffering visual clutter. An example that illustrates this flexibility with seven different $\mathcal{F}[\mathcal{S}]$ combinations is demonstrated in Figure 6.28 (e).

6.3.4 Initial User Feedback

To receive initial feedback on the advantages and disadvantages of the approach, a qualitative user study by means of expert interviews was conducted. In total, five visualization experts participated in the user study that took one hour in average. After an introduction to the system was given (20 min), the participants were asked to *think aloud* while exploring a dataset of the *Edinburgh Informatics Forum Pedestrian Database* [205] (see Chapter 6.3.1; duration: 30 min). During the browsing session,

the interactions of participants with the prototype were logged, such as the selection of facets and similarity measures and the number of gathered clusters. Finally, they were asked to fill out a questionnaire (10 min), which consisted of 10 questions. For each question, a 10-point Likert scale was provided and the participants were asked to comment their ratings. Please note that the provided means of the questions only give a first indication of user satisfaction without statistical significance.

The browsing approach as well as the visualization received very positive feedback. According to the ratings and comments of the participants, the tool is helpful to find common behavior, such as main paths and prominent directions (mean in Likert scale: 7.6). In contrast, unusual paths and abnormal behavior are more difficult to identify (mean: 6.0). The summary of answers of four questions about particular views and their linking exhibited that the visualization supports understanding of trajectory cluster characteristics (mean: 8.2). According to the questionnaire, the visualization communicates relevant information well (mean: 8.6) while it hides unnecessary details from the users (mean: 7.8). The participants agreed that the visualization successfully handles the issue of visual clutter (mean: 7.6). However, the sketchy nature of the visualization shows only little benefit for the exploration task (mean: 6.3). The different facets as well as the similarity measures were also considered very useful. However, the participants mentioned the system to be an expert tool with versatile opportunities that requires more time to learn which facets in combination with which similarity measures lead to particular behaviors. In this context, some of the participants also proposed to provide the possibility to save and load predefined facet/similarity combinations for standard queries (e.g., search for loitering people). Moreover, providing a history was suggested for support in order to keep track of previous browsing steps.

All of the facets (except object class since only persons exist in the dataset) were frequently used: the position facet was applied to 31.6 % of the re-scatter steps, azimuth to 27.7 %, speed to 30.2 %, and time to 24.9 % (>100 % due to multi-selection). Two participants suggested adding a relationship facet that allows clustering group behavior (i.e., considering tuples of moving objects with respect to a predefined temporal and spatial distance). The participants preferred the use of distance between means measure (59.8 %) over distance between standard deviations (25.2 %) and coverage (21.0 %). On average, the number of clusters was changed every 6.25^{th} step, 3.66 clusters were used, and every 2.57^{th} step the facet or similarity measure was changed.

6.3.5 Conclusion

In this section, a novel method to explore video data hierarchically was introduced. The proposed technique adapts scatter/gather browsing to trajectories of moving objects. Further control over this interactive browsing technique is provided to the users by the selection of arbitrary facets used to steer the scattering process. The effectiveness of

the interactive schematic summaries was illustrated in two examples, which pointed out that a good overview of video sequences can be obtained within a few browsing iterations. A fundamental part of the presented video browsing approach is the schematic visualization that consists of three views (spatial context view, temporal context view, and facet showcase view). This visualization facilitates exploration by a cluster representation that is easy to interpret and robust to visual clutter. It communicates the major characteristics of a video by highlighting the main paths of moving objects, hot spots, normal behavior, and prominent characteristics of facets. Clarity of visual representation is achieved by cartoon-style rendering of video context and by simplifying trajectory bundles. For this purpose, a novel trajectory bundling technique was introduced to reduce visual clutter and overdrawing. Finally, fusion of schematic summaries with interactive faceted scatter/gather browsing results in a powerful tool for the exploration of surveillance video, which was confirmed by a qualitative user study conducted to receive initial user feedback.

The user feedback also unveiled some directions for future work: the facets could be extended to provide the exploration of group behavior, and the possibility to save and load facet/similarity combinations is desired. Additionally, a browsing history was pointed out to be important, which led—chronologically afterward—to treat the ISS browsing steps as white list filters (see Chapter 4) and the depiction of them in the history-aware filter graph (see Figure 4.1).

6.4 Video Visualization for Tracked Moving Objects

After the strongly summarizing video visualization and browsing method of the last section, this section introduces a visualization for videos with tracked moving objects that is based on the VPG. The VPG was originally developed by Botchen et al. [35] and is enhanced for the purposes of the proposed video visual analytics system.

The modified version of the VPG aims at displaying relevant parts of video (defined in the *relevance measure* stage), entities extracted in the *feature extraction* stage, such as trajectories, and the consequences of filter definitions. Additionally, the video visualization is advanced to be uncertainty-aware: the objective is to communicate trajectory relevances (from the *relevance measure* stage, for example, confidences from the filter definitions) and uncertainties originating from the *feature extraction* stage.

The VPG visualizes continuous video streams similar to the illustration of seismographs and electrocardiograms. Features, such as trajectories of moving objects, are displayed together with a couple of keyframes that convey the context of the original video data. The VPG depicts a video volume composed of the time axis and two spatial axes of the video frames. This representation supports the navigation and orientation in the spatio-temporal video space. Hence, users are enabled to keep track of spatial and

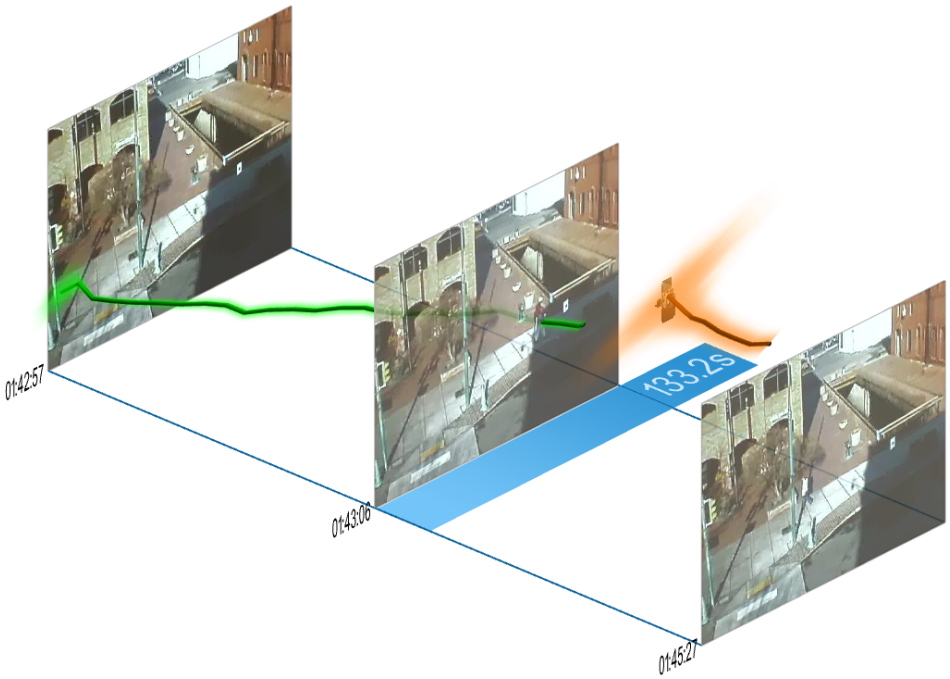


Figure 6.30 — Video visualized by the *VideoPerpetuoGram* (VPG) containing the two spatial axes and the temporal axis of a video. Three keyframes with their time stamps are depicted to convey context information. Although there are just few keyframes, the video volume is dense and shows detailed trajectories (in the space-time volume) with their relevance that is defined as the *degree of membership* (DOM) related to filters and is mapped to color (green: high; red: low). Their positional uncertainty is illustrated by semi-transparent blur. The green trajectory, featuring high DOM, shows a person walking from left across the footway. The orange trajectory, featuring moderate to low DOM, represents a car appearing for a short period in the upper part of the video. Trajectories with a DOM below a user-defined threshold are hidden, and periods without trajectories are skipped. Between the second and third keyframe, no trajectory is left. Therefore, this period can be skipped and is cut off the (afore dense) video volume. The blue bar indicates the gap in the video volume and the amount of time skipped.

temporal relationships between several events. This abstract visualization of the video helps overview a long video sequence at a glance and allows interactive exploration of trajectories.

Figure 6.30 shows the enhanced VPG visualizing a part of the video stream from the IEEE VAST Challenge 2009 dataset [1].

There is a trade-off between using abstract visualizations and visualizations similar to

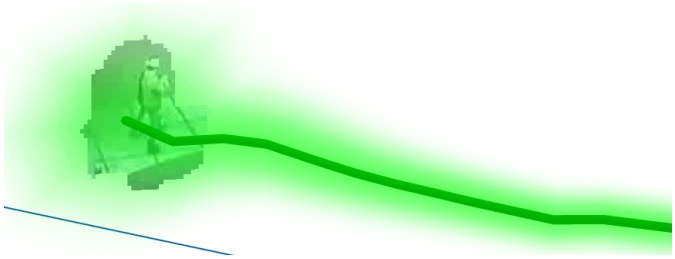


Figure 6.31 — The trajectory of a moving object is depicted by a geometric tube. A blob in the beginning conveys context information, i.e., to which object this trajectory belongs. DOM with respect to filters and positional uncertainty is depicted by color and semi-transparent blur, respectively.

standard video playback. The former provide faster exploration with more details than the latter, but are unfamiliar. Therefore, the proposed video visual analytics system provides both kinds of views. Nevertheless, it was demonstrated by a controlled user study that users are able to learn abstract visual signatures for abstract visualization [55]. Even visualization novices are able to understand the VPG after introducing simple related 2D visualization designs (see Chapter 6.5).

To convey data quality, uncertainty information has to be provided by the VPG. There are different visual mappings of uncertainty discussed in literature: adding glyphs, adding or modifying geometry, modifying attributes (e.g., color and shading), using animation [225], and applying transparency [67]. Botchen et al. [35] use two levels of saturation to indicate the relationship plausibility of trajectories in the VPG. In contrast, the method proposed here uses hue to encode relevance assigned to trajectories in the *relevance measure* stage. This relevance can originate, for instance, from the DOM of the trajectories with respect to the defined filters (see Chapter 4 and 5). In addition, quantitatively expressed trajectory quality as well as further information of the trajectory can be acquired on demand by selection of the trajectories.

Furthermore, trajectories may be filtered out in the *filtering* stage, and a relevance measure on frames can be defined that treats frames without the occurence of any relevant trajectories as irrelevant (see Chapter 5). The proposed VPG utilizes these frame relevances, which results in skipped periods whenever no relevant trajectories are present. The blue bar in Figure 6.30 indicates the amount of time skipped.

Another modification of the original VPG is the visualization of uncertainty information. The VPG is superimposed by semi-transparent blur that shows the positional uncertainty originating from the *feature extraction* stage. This represents the probability density function of the object's location; it is realized by an elliptical tube with an according transparency distribution. The uncertainty visualization enables the users to be aware of the trajectory's quality and facilitates reliable conclusions.

Another extension of the VPG aims to improve context information. The trajectory's start position is therefore augmented with the blob of the moving object (see Figure 6.30 (orange trajectory) and Figure 6.31).

In the next section, another advancement of the VPG is discussed in context of a feasibility study that evaluates whether video visualization can assist snooker skill training.

6.5 Video Visualization for Snooker Skill Training

The visualizations introduced above mostly focus on tasks in context of video surveillance with users that are most likely forensic experts. The application area in this section is snooker skill training and the target group consists of coaches and players. In detail, a feasibility study conducted in conjunction with a snooker club and a sports scientist on using video visualization to aid snooker skill training is presented. By involving the coaches and players in the loop of intelligent reasoning, the approach addresses the difficulties of automated semantic reasoning by visualization.

The visualizations discussed in the previous sections aim to reach scalability for long video sequences. The approach there is either to condense the dimension time in dynamic visualizations (e.g., the fast-forward visualizations or the video visualization for tracked moving objects) or to browse static summaries interactively (e.g., the ISS). In contrast to that, the videos originating from snooker shots are short and a static visualization that summarizes the whole shot without the need of interaction is proposed in this section. In detail, the approach utilizes the principal design of the VPG to convey spatio-temporal information to the viewers through static visualization to remove the burden of repeated video viewing. Therefore, the VPG design is extended to accommodate the need for depicting multiple video streams and respective temporal attribute fields, including silhouette extrusion, spatial attributes, and non-spatial attributes. Nevertheless, the proposed pipeline for video visual analytics (see Chapter 2.4) is sufficiently general to deal with this variability. The extraction of the features required by this approach is done in the *feature extraction* stage and discussed in Chapter 3.3.

6.5.1 Application Background

Cue sports encompass a family of skill-based games where a player uses a wooden cue to strike billiard balls on a table. Snooker is one such sport, which has been popular in many English-speaking countries since the 19th century, and in recent years, its popularity grew rapidly in Asia. The game of snooker has benefited greatly from the arrival of color television in the early 1970s and affordable video technology in the

1980s. Today, videos are used extensively in snooker coaching. In comparison with most other sports, however, snooker is yet to benefit from modern technology-based performance analysis and coaching.

In snooker skill training, all snooker coaches encounter the difficulty of analyzing the progress of a player quantitatively and making an objective comparison between players. Although videos provide an effective means of recording raw data, watching videos is time-consuming and making a comparative judgment by juxtaposing videos is generally ineffective. An average snooker shot takes about 2–3 seconds, while a cue strikes in the blink of an eye. High-speed filming is often necessary, but watching such slow motion videos in everyday training is agonizingly laborious.

The approach addresses the above-mentioned difficulties by applying video visualization to snooker skill training. In particular, it utilizes the principal design of the VPG [35], which provides a focus-and-context visualization of a video stream. It extends the design of the VPG to accommodate the need for depicting multiple attribute fields, including silhouette extrusion of objects, spatial time series (e.g., the center of a snooker ball), and non-spatial time series (e.g., ball size). This work was conducted in conjunction with a snooker club that is led by a former world champion and offers coaching to both professional and amateur players. A few of the professional players trained in the club are ranked among the top 20 players in the Official 2000 World Snooker Rankings. It is a feasibility study to determine whether video visualization can be deployed in a snooker club to aid snooker training, and if so, what further investment in equipment, research, and development is necessary for realizing a technology usable in everyday training.

First, the specific needs of snooker skill training are described, which provide the motivation for this approach, before the objectives of the work are discussed.

Snooker Skill Training

A good snooker player possesses a wide range of skills. While a beginner needs to master basic skills such as maintaining correct stance and grip, forming a bridge with the hand (a V-shaped channel), aligning the cue, and delivering a strike [90], a professional needs to possess necessary mental qualities such as motivation, commitment, concentration, confidence, and decision making under pressure [64, 305]. This work focuses on a set of skills at the intermediate level. Players at such a level have already acquired basic skills as well as a reasonable feeling of the interaction between different entities (i.e., body, sight, cue, ball, table, etc.).

For players at the intermediate level, snooker coaches are interested in the following skills [119]: speed of delivery, application of power, stun, screw, side, cue alignment, spin delivery, and spin avoidance. The feasibility study focuses on spin avoidance: the ability to strike the cue ball without applying unintentional spin.

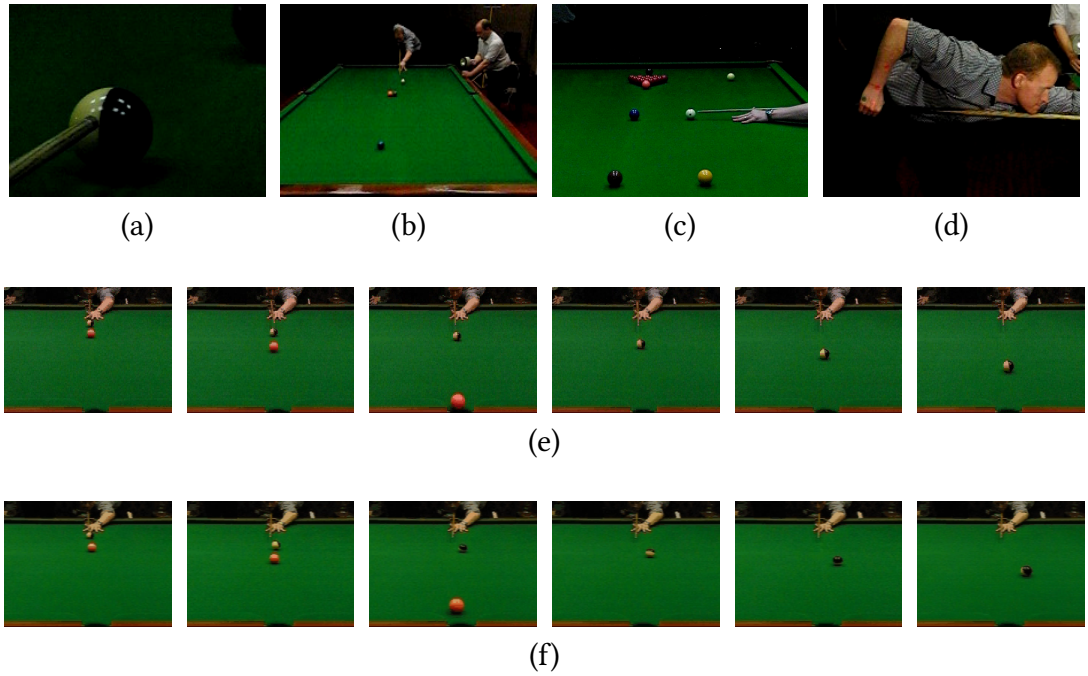


Figure 6.32 — Example frames extracted from six typical snooker skill training videos. (a)–(d) represent different actions that are interesting to snooker coaches: (a) cue and ball interaction, (b) ball trajectory, (c) alignment and delivery, and (d) grip and wrist motion. The two video sequences "pink-pot-b2" (e) and "pink-pot-c2" (f) show different spin avoidance actions and are each represented by six example frames ($t = 0, 62, 124, 186, 248, 310$).

Videos can help maintain a good training record by capturing problems and improvements in aspects that are not easily observable during training. Figure 6.32 shows four different aspects in (a)–(d) that coaches would examine closely. Figure 6.32 also shows keyframes extracted from two video sequences in (e) and (f) capturing different spin avoidance actions. While watching videos is intuitive and effective in many cases, it is time-consuming and objective comparisons are difficult. Snooker coaches are longing for modern technologies that can help coaching and training. In comparison with many sports, such as tennis and soccer, the deployment of modern technology in snooker for skill training and performance analysis is rare. In addition to scientific and technical challenges, the technology developed for snooker training must also address the challenges of keeping the costs and resource requirements low.

Application Stakeholders

Most snooker training clubs in the United Kingdom are organized around one or two coaches. As in most sports, coaches and professional players are highly motivated and have a great urge to learn and use new technology. However, they have very limited prior exposure to advanced visualization. This is very different from many other visualization applications that mainly involve users in the scientific and medical communities. It thereby presents an extra challenge to this work. As mentioned in Chapter 3.3.1, the club considered to invest in video capture and replay equipment, but was uncertain if the investment is worth it. Thus, this work is initiated as a feasibility study to help the club answer the following questions:

1. Can video visualization complement video watching in performance analysis and progress monitoring?
2. Can coaches recognize visual signatures in video visualization and will they be willing to learn such skills?
3. What are the estimated costs and resource requirements to make video visualization a usable technology in a snooker club?
4. Does video visualization have the potential to help increase the use of a costly suite of ceiling-mounted and computer-controlled video capturing and replaying equipment, thus justifying the investment?

The feasibility study focused on four videos (“pink-pot-b1”, “pink-pot-b2”, “pink-pot-c1”, and “pink-pot-c2”) showing the cue action *spin avoidance*. The four videos show two different shots (b and c) that are each captured from two different positions: the side (1) and the front (2). In Figure 6.32 (e) and (f), six keyframes of each shot from the front view are depicted. Since the normal snooker cue ball has no feature that allows human or machine vision to quantify spinning, a training cue ball that has two colored halves, black and white, is used. Some of the proposed visualizations were purposely designed to depict spins using this property. In comparison with other spatial transformation (e.g., translation and scaling), spinning is also more difficult to visualize [55].

Information about video capturing, including details on the used hardware and its configuration, is provided in Chapter 3.3.1.

Approaches for Sports Analysis

Video-based analysis is commonly used in modern sports to aid performance analysis and improvement. In practice, most such analysis is carried out by repeated video

viewing and qualitative discussions in front of a television. Many attempts have been made to use automated computer vision techniques. The current advances in this area are represented by a special issue in *Computer Vision and Image Understanding* [204]. These attempts generally fall into the following categories:

- **Tracking.** Techniques in this category provide the basis for high-level analytical tasks by establishing the motion trajectories of interesting objects or players. For example, Ren et al. [248] reported a technique for tracking a soccer ball from multiple fixed camera views. Kristan et al. [172] presented an algorithm for tracking multiple players in several indoor sports applications. In general, there is a large volume of literature on tracking techniques and their applications in sports [5, 106]. Advances in this aspect include sophisticated algorithms for handling motion-blurred images [49], and graph-based association of tracked objects [314].
- **Indexing and Retrieval.** Techniques in this category allow videos of sport events to be temporally segmented, indexed with known information (e.g., event hierarchy, sensor parameters, landmarks, etc.), and stored in a multimedia database for future contents-based retrieval or further video processing. For example, Pingali et al. [235] reported such a system for tennis games. Assfalg et al. [16] presented a system that segments a sports video into shots of studio interviews, statistical graphics, audience, playing fields, and close-up views of players. They worked with videos of 10 different sports, with varying accuracy, for example, 38 % (javelin), 57 % (diving), 69 % (tennis), 80 % (soccer), and 88 % (track). The technology in this aspect is relatively mature, though human involvement in correcting erroneous classification is necessary in practice.
- **Event Classification.** Techniques in this category are intended to generate semantic description of events in videos using automated reasoning. For example, D'Orazio et al. [80] developed a vision system for classifying doubtful goal scoring situations in soccer matches using cameras located along the goal line. Perše et al. [230] presented a method for detecting three phases, namely offensive, defensive and time-out phases, in a basketball game, achieving 92 % accuracy. In general, the successes in this area are limited by both restricted filming conditions and simplicity of semantic classes.

The term "visualization" in sports commonly refers to mental practice and rehearsal [260], which is out of scope for this thesis. One of the common uses of visualization in sports is to depict the motion trajectories of moving objects and players. Pingali et al. [234] presented a system for tennis with visual designs for viewing trajectory lines, coverage maps, landing points, 2D charts and virtual 3D replay. They demonstrated that visualization can provide insight into performance, style, and strategy in sports.

Such visual designs are now commonly seen in sports broadcasts. Grau et al. [116] presented a system for reconstructing soccer matches from multiple cameras, and allowing replay from arbitrary viewpoints where moving objects are approximated by billboards, visual hulls, or view-dependent geometry. Denman et al. [76] presented a tool for summarizing ball trajectories overlaid on a video frame. They also used 2D graphs to illustrate ball position and speed.

In summary, there are three typical types of sports visualization. (i) Conventional *2D graphs* maximize the key information encoded in abstract representations, and are suitable for domain experts; but do not contain sufficient context information from the video, and are unintuitive to many potential users. (ii) *2D illustrations* (e.g., trajectory and coverage map) overlaid on keyframes are intuitive to use and suitable for both experts and novices; but they contain only simple temporal information, and do not support complex temporal reasoning (e.g., spin). (iii) *3D reconstructed models* are intuitive to use and entertaining to explore; but are difficult to deploy, and usually limited by complex technical requirements (e.g., multi-camera installation). They are also time-consuming to explore, and their support to temporal reasoning is limited by temporal attention and memory. Therefore, it is highly desirable to use advanced video visualization techniques to address the shortcomings of existing sports visualization.

To the best of my knowledge, there has not been any prior application case study on sports video visualization.

6.5.2 Multi-Strand VideoPerpetuoGram

The visualization component is based on the VPG framework originally designed for the integration of spatial and temporal aspects in video visualization [35]. The proposed approach follows the focus-and-context design of the VPG, with a few technical modifications to accommodate the needs of this application. The technical contributions to visualization are the extensions of the VPG to combine different attributes in the same visualization (multi-attributes extension), and the possibility to have several temporally synchronized visualization images displayed simultaneously (multi-strand extension). As shown by Botchen et al. [35], the rendering of the VPG is real-time and the modifications made here maintain interactive frame rates. This section outlines the general considerations for the visual design, then the visualization of individual spatio-temporal attributes extracted in the *feature extraction* stage are considered (see Chapter 3.3), and finally the combination of different attributes in the same visualization, and the multi-strand VPG are discussed.

The targeted audience of this feasibility study includes snooker coaches and players. It is likely that these users are not familiar with advanced visualization systems. Based on this, the following design principles were defined, and later improved according to the feedback from the application partners.

Color Mapping. Color mapping should enable the visualization to match the colors on a snooker table as close as possible. The trajectory of a snooker ball should ideally be depicted using the original ball color. When there is a conflict, such as with the background, a uniquely identifiable color should be used within the visualization.

Providing Context. Following the VPG design, the context (i.e., keyframes of a video) should be present in every visualization. One or just a few keyframes are often adequate as the scene is relatively static.

Minimizing Navigation. Since the primary advantage of video visualization is to save time, time saved for watching video should not be replaced with required time for interaction and navigation. Hence, users are given a few fixed camera positions from which visualizations are generated. Similarly, orthographic projection is employed to avoid misleading perspective foreshortening.

Visualization Literacy. The assumption for this work is that visualization literacy can be improved. Coaches and players should be prepared with simple visualization designs, and gradually introduce designs that are more complex.

Visual Mapping of Spatial Attributes

Attributes are pieces of information extracted from a video during the *feature extraction* stage. As shown in Chapter 3.3, such information is obtained frame by frame. Hence, all attributes are temporal. Some attributes, such as the center of a ball or the line separating a black-white cue ball, are inherently spatial, and can be placed in 3D space with the time as the third dimension. These are termed *spatial attributes*.

Displaying spatial attributes in 3D is intuitive. As shown in Figure 6.33 (a), the center of a ball can be displayed in conjunction with a few keyframes. Similarly, the centers of three colored segments pink, black and white can also be displayed as in Figure 6.33 (b).

These time series of positions are visualized by rendering thick lines implemented as geometric tubes and indicating the trajectories of the balls.

Note that for video “pink-pot-c2”, which did not manage to avoid spin, the patterns in (b) are better observable than in (a). This is because the centers of the black and white segments of the cue ball revolve around each other during the spin. Nevertheless, the tubes in (a) are suitable to show the trajectories of the balls.

The object silhouette can be displayed as a volumetric object passing through the keyframes, as in Figure 6.33 (c). The transfer function used is similar to Ebert et al. [84], where the RGB values in the video are preserved, and alpha is derived from image analysis results. While the object silhouette volumes are inappropriate to show the degree of spin, they can provide the trajectories of balls preserving original radii and colors.

The separating edge of the black-white ball is depicted with a ribbon. A spinning black-white ball results in a twisting or broken ribbon, providing a visually intuitive way to compare cue actions for spin avoidance. For example, as shown in Figure 6.33 (d), a poor shot causes twisted or broken ribbons, while a good shot will not.

Visual Mapping of Non-Spatial Attributes

Certain temporal attributes extracted in the *feature extraction* stage do not have any inherent spatial location, yet still form a time series. These are termed *non-spatial attributes*. For example, the numbers of black and white pixels in the two videos in Figures 6.32 (e) and (f) give a good indication of whether the black-white cue ball is spinning.

In principle, non-spatial attributes can be displayed adequately using a 2D plot. However, this contradicts the principle of providing context (e.g., video frames), whenever possible. Thus, a context and focus visualization for the ratio of white pixels on the black-white cue ball is designed (Figure 6.34). This ratio is defined as: $\frac{\#\,white\ pixels}{\#\,white\ pixels\ +\ \#\,black\ pixels}$. In the visualization, this ratio is mapped to the radius of a tube that emerges from the corresponding ball position in the keyframe.

Note that some attributes and their visual mappings provide insights into similar properties. For example, both trajectories of ball centers (Figure 6.33 (a)) and object silhouette volume (Figure 6.33 (c)) show ball trajectories. The spin of a shot is conveyed by different visual patterns: trajectories of segment centers (Figure 6.33 (b)), the ribbon showing the separation edge on the black-white cue ball (Figure 6.33 (d)), and the non-spatial ratio visualization (Figure 6.34). Therefore, it is not mandatory to show all visual patterns, but the user may benefit from several, independent visualizations in the case of borderline shots.

Multi-Attributes VPG

Multiple attributes can be combined within a single VPG, as demonstrated in Figure 6.35. In addition to the keyframe(s), each visualization shows the object silhouette, and the centers of the black and white segments. A special case of the VPG can be created by looking along the temporal axis toward the initial keyframe; implicitly registering data within the static scene (Figure 6.35 (a)). This special-case layout is similar to many visual representations of trajectories in videos. While this visual design has the advantage of being intuitive, it suffers several drawbacks. It lacks of temporal reference for information, such as speed. It also has limited degrees of freedom for the orientation of tracking geometries. For example, the trajectories of color segments in Figure 6.33 (b) can go up-and-down as well as sideways. It is not easy to depict an up and down motion with the visual design in Figure 6.35 (a). The VPG supports arbitrary, uniform scales of the time axis. The trajectories in Figure 6.35 (b) are space-time trajectories

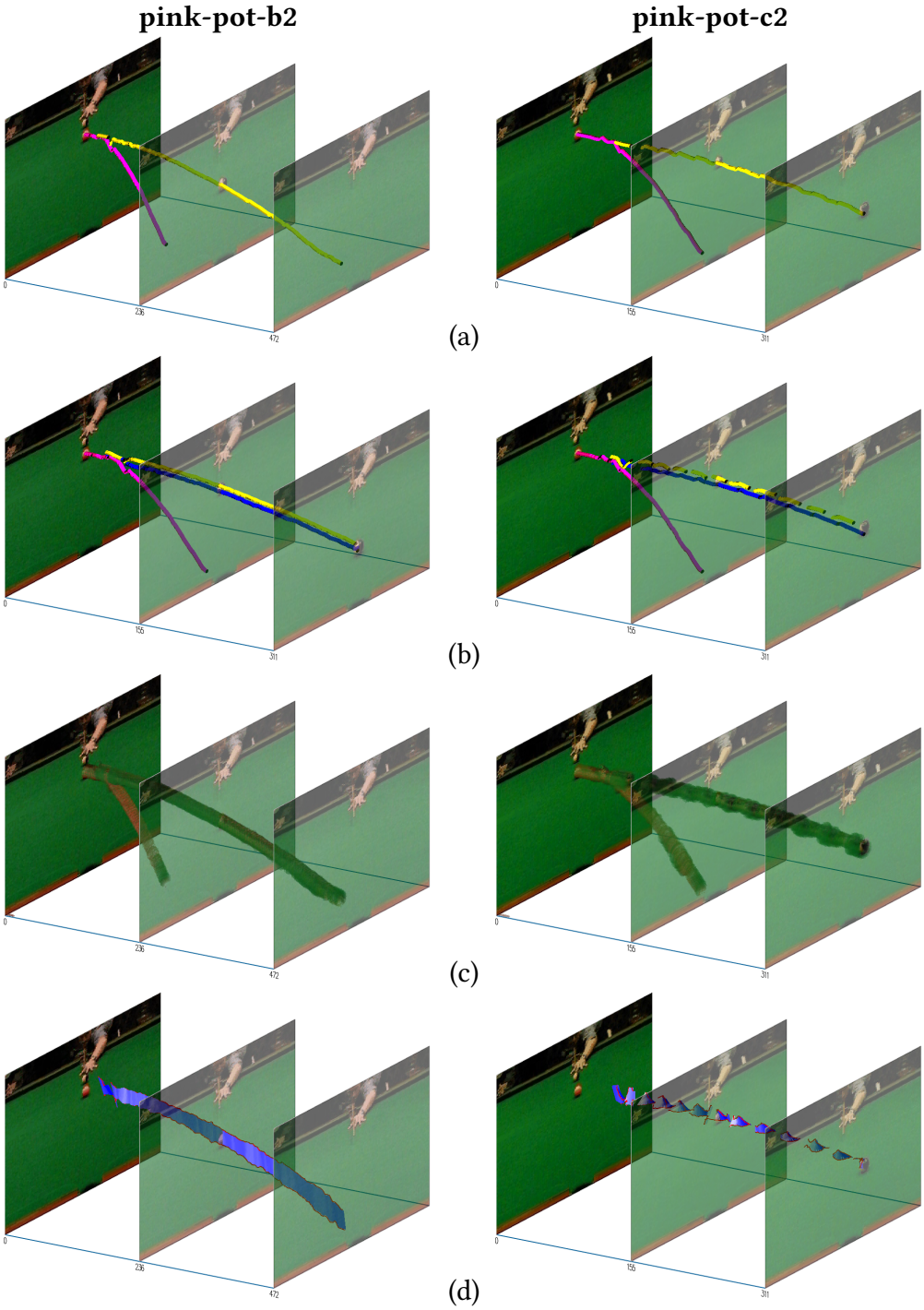
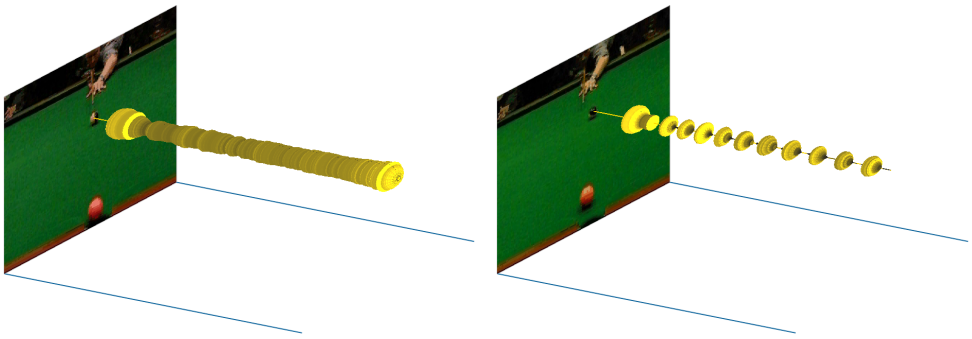


Figure 6.33 — Visualizing four different types of spatial attributes: (a) trajectories of ball centers, (b) trajectories of color segments, (c) object silhouette volume, and (d) the separation edge on the black-white cue ball.



(a) (b)

Figure 6.34 — Visualizing non-spatial attributes on the video sequences pink-pot-b2 (a) and pink-pot-c2 (b). The radius of the yellow tube-like object shows the temporally varying ratio of white pixels on the black-white cue ball.

while the ones in Figure 6.35 (a) are their projection along the time axis. The latter is more familiar to most of novice users. Nevertheless, for the spin avoidance videos, the coaches are not interested in the actual path of the balls, but the juddering or deformation of the tracking geometries. Furthermore, interpretation of space-time trajectories can be learned (as demonstrated in a large controlled user study [55], see also discussion in the next section, i.e., Chapter 6.5.3).

Allowing the simultaneous display of volume data and surface geometry is a further extension to the VPG. The display of surface and volume data together is achieved by first rendering the opaque surface geometry (with depth write and depth test enabled), then rendering the slices through the volume (with blending and depth test enabled, but depth write disabled).

Multi-Strand VPG

A major extension of the VPG is to enable the simultaneous display of several, temporally synchronized visualization images. Multiple time-series visualizations, termed *multi-strand*, are shown side-by-side to visualize a collection of spatial and non-spatial attributes (Figure 6.36). Attributes with similar spatial context may also be separated to prevent visual overload of the user, and occlusion of the VPG. For example, the separating edge should normally be in a different strand from that for the ball centers, as they frequently occlude each other.

Another advantage of the multi-strand visualization is the possibility to visualize a snooker shot, captured by two cameras from different views, at once. A single camera may miss some important features of a shot. Therefore, a two-camera setup capturing the snooker table from a longitudinal and a transverse side is used. Due to the mirror

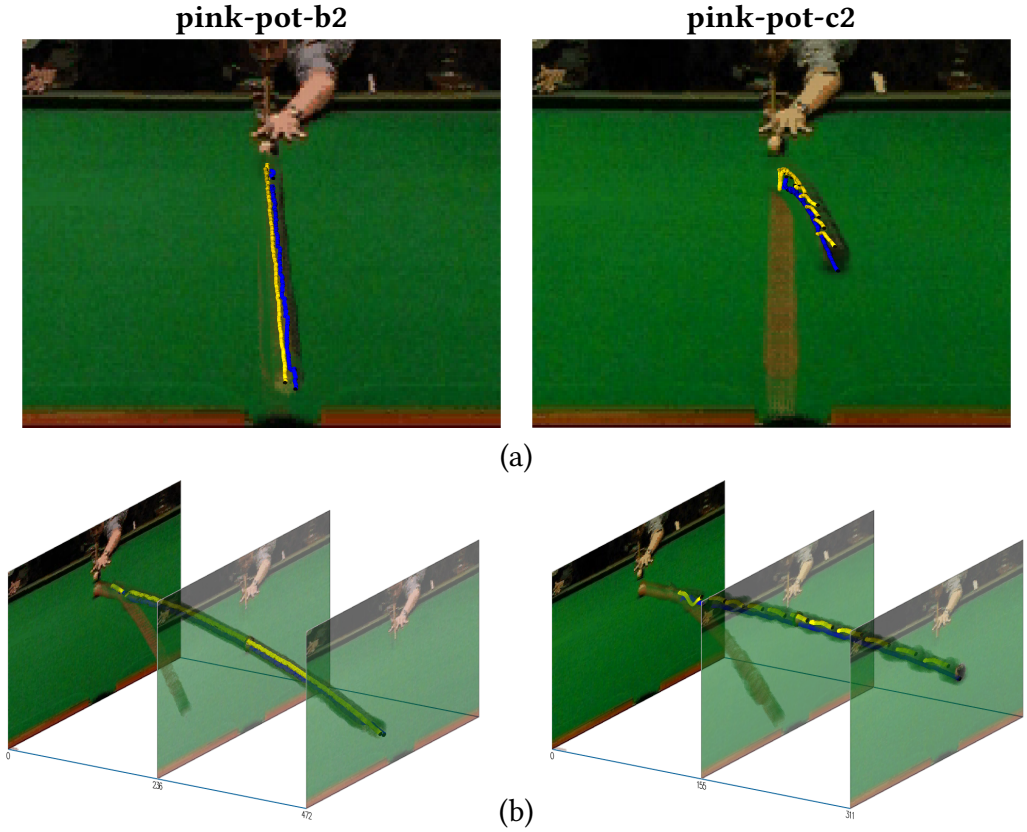


Figure 6.35 — A VPG that combines object silhouette volume and trajectories of color segments: (a) front view, (b) side view.

symmetry of the black-white cue ball, this setup can assure that the desired features are always captured by at least one camera.

The multi-strand VPG is motivated by the navigation cost analysis by Ware [296]. He discusses various methods for navigating through information spaces and their time costs. In the ideal visualization, all information is available on a single high-resolution screen. Hence, a single saccadic eye movement is sufficient to focus from one location to another, which will take about 150 ms. Meanwhile, the cognitive effort for alternative navigation methods, such as a hypertext link, takes at least 2 s. Therefore, a side-by-side visualization is chosen with several attributes of up to two shots.

The multi-strand visualization enables snooker coaches to evaluate shots at a glance even if they were not present. This is typically due to training of several players simultaneously. Additionally, coaches can explain the shot, the mistakes, and the further corrections in front of their trainee using the visualization without having to watch slow motion videos repeatedly.

One of the most important aspects in snooker skill training is repetition. Players have to repeat the same exercises many times to gain accurate control of speed, power, cuing, and other skills. The multi-strand visualization can support such repetitive processes cost-effectively. For example, a coach can give an example shot to illustrate what is required. The players can repeat the exercise and juxtapose their shots with the example using the multi-strand VPG. It is also possible for the players to train on their own, while the coach can monitor the progress through the visualization.

6.5.3 Results and Evaluation

For this feasibility study, a variety of VPG visualizations were designed and rendered. These included:

- *Spatial attributes*, with keyframes as context (Figures 6.33 (a)–(d)).
- *Non-spatial attributes*, with a keyframe as context (Figure 6.34).
- *Combined object silhouette volume and trajectories of color segments*, with keyframes as context (Figure 6.35).
- *Multi-strand VPG* (Figure 6.36).

It is not generally feasible to organize a user study involving a large number of snooker coaches because there are only about 20–30 registered snooker coaches in the UK, spreading in different parts of the country. A validation meeting is organized with five potential users to evaluate the visual results of the feasibility study. The five participants included two full-time snooker coaches, C_1 and C_2, one of whom is a former world champion, one sports scientist, S, who is also a cricket coach, one intermediate-level amateur player A_1, and one basic level amateur player, A_2, who has a full length snooker table at home.

Coach C_2 manages a snooker club and trains intermediate and basic level players. The participants' knowledge about visualization is largely limited to basic graph plotting, such as in spreadsheets. Hence, an evaluation through open discussions allows engaging the potential users not only to provide the feedback, but also to learn and appreciate merits of visualization.

Six sets of questions were prepared, and the discussions were organized using PowerPoint slides that also showed some example videos and visualization results. In addition, two sets of paper copies of sample visualization results forms were prepared, on which a coach could write further comments for a trainee. None of the participants saw any visualization results prior to the meeting. The discussions, which took about an hour, are summarized in a table in the appendix of the original publication [136].

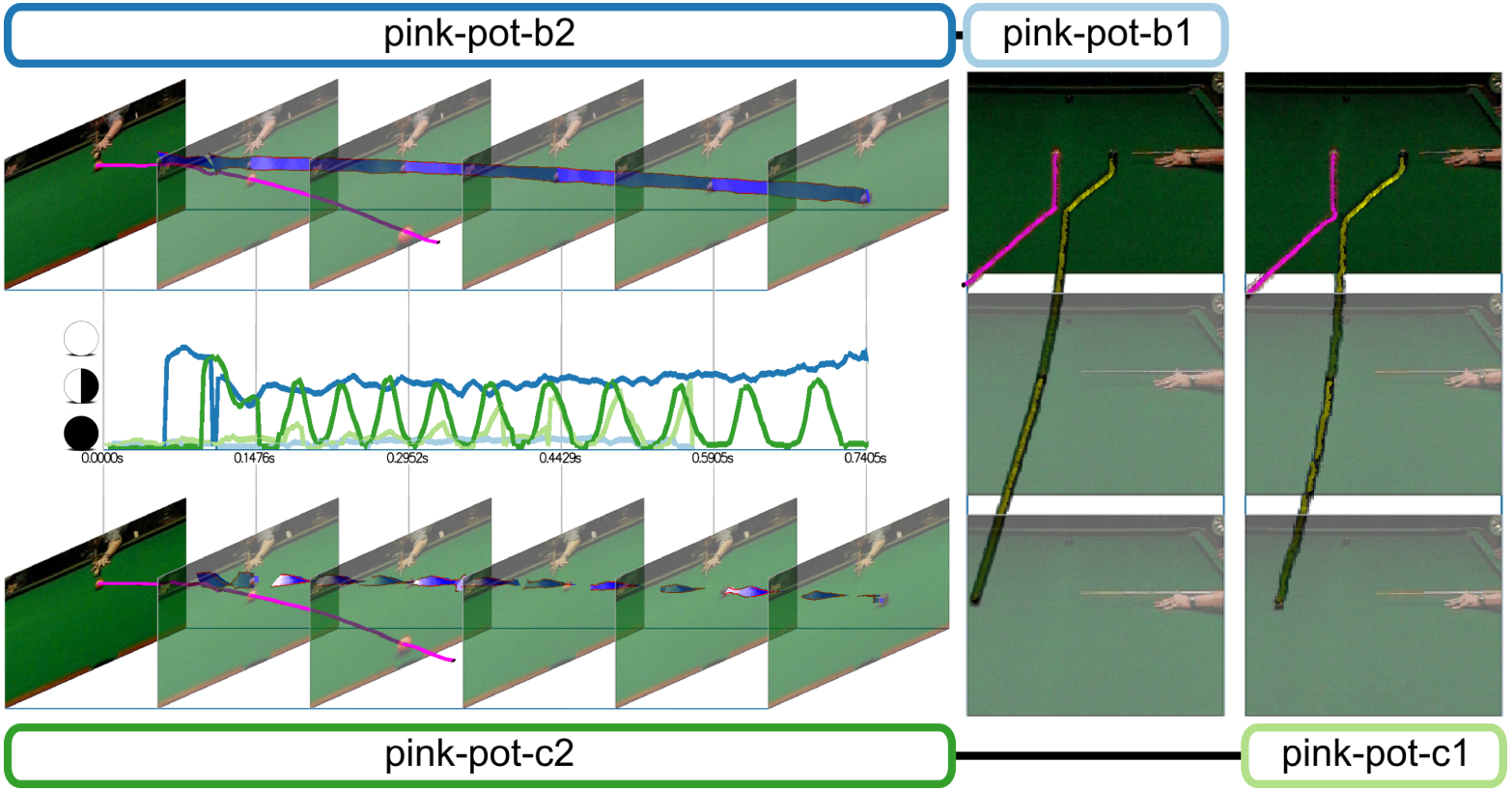


Figure 6.36 — A multi-strand VideoPerpetuoGram comparing the two shots “pink-pot-b” and “pink-pot-c”, each captured from the side (1) and the front (2). The front view video sequences are the same as shown in Figure 6.32. On the upper and lower left side, the front views of the shots are visualized by keyframes, pink ball centers, and the separation lines on the black-white cue balls represented by ribbons. The side views are displayed on the right side. The attributes shown here are keyframes, object silhouettes, and ball centers. The function plots on the left side show the ratio of white pixels of the black-white cue ball. The four curves correspond to the four different video sequences. Their colors match with the colors of the bounding boxes containing the video names. A 2D plot is used to display the non-spatial attributes since the spatial context can be provided by temporal alignment, horizontal spatial alignment, and reference lines to the associated strands of video “pink-pot-b2” and “pink-pot-c2”. A detailed comparison is only required between parallel views, which limits the cognitive effort. Additionally, some features can be visualized in all VPGs, such as the trajectory of the pink ball. This may help the user transfer the information even to orthogonal views.

During the meeting, the video visualizations were introduced gradually by first showing a 2D example, such as a stock market graph, and then showing a video visualization that uses a similar visual metaphor. This was very effective to help the participants understand the results of video visualization. In general, the coaches who took part in the meeting liked the visualizations in Figures 6.33 (a)–(b), 6.34, 6.35 (a), and 6.36. They had some difficulties with translucent volumes as in Figure 6.35 (b). This may be partly because those volumetric effects are inherently more difficult to appreciate and partly because of no effective 2D stimulus was available to introduce this concept. The visualization results in Figure 6.33 (d) were not available to the meeting.

A few visualizations similar to Figure 6.34 were shown in the meeting. Minard's map was used as an introduction, which enthused the participants and stimulated much discussion. Figure 6.34 is an improved version of what was shown in the meeting.

Participants were convinced that video visualization offers an effective means for communication, comparison, and archiving. They also appreciated our observation that automatic video analysis is not as readily usable as video visualization. Some participants were relieved that the technology is not ready to replace the coaches, while some remained hopeful for a fully automatic technology.

In terms of the four questions listed in Chapter 6.5.1 (page 136), the observations based on the discussions in the validation meeting, further engagement with the participants, and the experience gained throughout the feasibility study can be summarized as follows:

1. Snooker coaches are longing for a technology that reduces time spent watching videos, and aid their analysis and monitoring tasks. The coaches who took part in the evaluation were not concerned with the fact that visualization needs human interpretation, and liked the idea that they are not replaced by unreliable machine intelligence. However, for visualization to be usable, the functionality delivered in types of this feasibility study has to be scaled up to some 10–20 cuing actions.

2. Coaches can learn to recognize visual signatures. In particular, it was surprising how quickly they transferred their learning of Minard's map to the visualization of non-spatial attributes (Figure 6.34). Learning stimuli (e.g., simpler examples) and metaphors might play an important role in learning visual signatures.

3. Visualization can be used in many aspects of snooker skill training. In addition, the participants identified a number of potential uses of such visualization, including (i) individual training records in the form of a collection of reports similar to the sample paper copies, (ii) a collection of visualization results from professionals as benchmarks, (iii) introduction of standard skill tests, and (iv) visual records of snooker equipment testing.

4. It is common for ordinary people to overestimate what can be accomplished by machine intelligence. The initial expectation of the snooker coaches was that computer vision techniques are readily available to analyze captured video automatically, for example, by categorizing and recognizing various types of shots, and subtle difference in cuing actions and poses. In this study, it was observed that for automatic video analysis to deliver reliable results, more sophisticated environment (e.g., lighting and ball textures), accurate calibration, and high resolution and high-speed cameras were required. The costs and inconvenience of setting up the environment for each training test will likely outweigh the benefits from time reduction in viewing the videos. This feasibility study made a convincing case for using video visualization as the primary technology for supporting snooker training.

5. The snooker club has since made its investment to install ceiling-mounted cameras. Additionally, the snooker club has searched and found financial sponsors for developing video visualization technology: 1.5 developers were funded for 2 years.

For video visualization, once a visual design is finalized and implemented, it can generally be transferred to other shots. However, this is usually not true for video processing. The area coverage of a shot, the number of balls, lighting, etc. could force some modification to the *feature extraction* stage. If more automatic classification and recognition techniques were used, the semantics of each shot have to be hard-coded into vision algorithms. Hence, for delivering a video visualization system, with 10 times of the functionality delivered in this feasibility study, and taking into account the development cost for this feasibility study, it is estimated that about 10 person-years are necessary for realizing a usable technology in the form of an industrial product. This includes 4 person-years for generic software features, 1 person-year for management, and 10×0.5 person-years for test-specific software features. For developing automatic video analysis to support a similar level of functionality, it is estimated that at least 30 person-years are necessary for realizing a usable system. This includes 3 person-years for generic software features, 2 person-year for management, and 10×2.5 person-years for test-specific software development (e.g., training data capturing, video processing, supervised learning, classification and results presentation).

6.5.4 Conclusion

In this section, a feasibility study on the application of video visualization in snooker skill training was presented. The study has found that the use of automatic video analysis techniques in practice has been hindered by several factors, such as varying accuracy, poor portability from one sport to another, and semantic bottleneck in

supervised machine learning. Existing commercial systems are limited largely to motion tracking and video editing and annotation. Viewing videos is still the main operational mode after a video is processed by such a system.

To make a convincing case for using video visualization to reduce the burden of viewing videos, a prototype system was developed in conjunction with a snooker club. The concept of the VPG was adapted and extended to accommodate various attributes that were calculated at the *feature extraction* stage using reliable and well-known computer vision techniques. This section introduced a multi-attributes extension as well as a multi-strand VPG. Both can be utilized in other applications whenever several attributes should be visualized in one or more VPGs simultaneously. One outcome of the feasibility study is that video visualization is a more "transferable" technology than automated machine vision, so it would cost less to develop it in a short to medium term. Another outcome is that video visualization has to cover a reasonable number of common tasks in snooker training before it becomes cost-effective to deploy. It was a huge reward to find out that novice users can learn to comprehend and appreciate video visualization, and to recognize visual signatures. Although they showed preference for more intuitive visualization as a means to communicate with the players, they enjoyed the collaboration, and determined to continue their investment in new technologies.

This feasibility study opens up many new opportunities for further research. In particular, it would be interesting to study the correlation between different snooker videos. Moreover, new applications of video visualization should be identified, and the video visualization technology should be advanced to serve such applications.

6.6 Layered TimeRadarTrees

In this section of the visualization chapter, a more general visualization, the *layered TimeRadarTrees*, is condensedly presented. The visualization combines *indented tree plots* [44] with *TimeRadarTrees* [43] and shows the temporal evolution of relations in a static view. Hence, it is, besides the video visualization for snooker skill training discussed in Chapter 6.5, an example of static visualization that summarizes a particular time window of a video.

The layered TimeRadarTrees is a technique for visualizing dense time-varying directed and weighted multi-graphs with an additional hierarchical organization of the graph nodes. The visualization can be interpreted in two ways: (i) as an example of video visualization that is solely based on features that are extracted from video and (ii) as an example of a general visualization (here: a graph visualization) applied to the video domain. Regardless of the interpretation, the technique is applicable to many different domains where time-varying relational data is available. When visualizing video, the technique requires extracting and organizing relational features in a time-

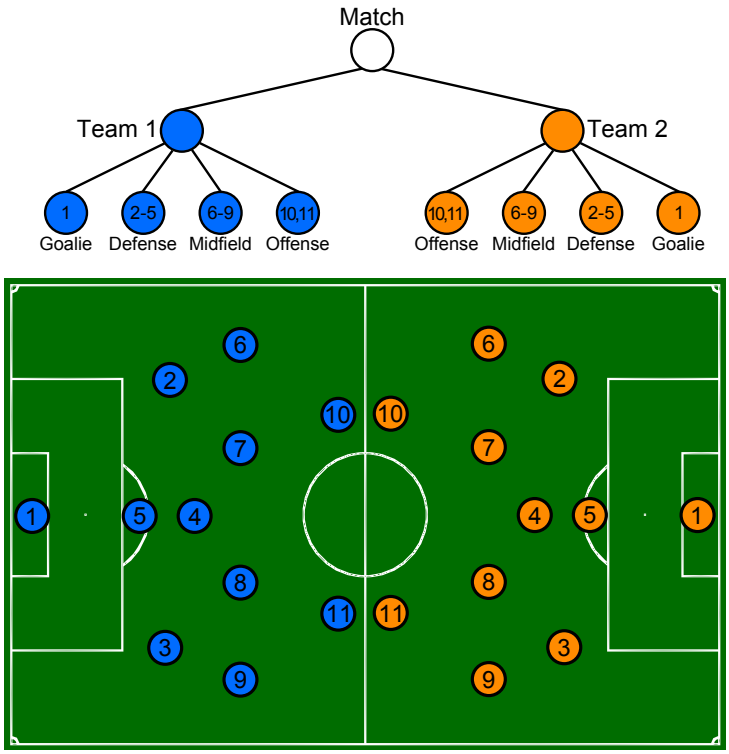


Figure 6.37 — Hierarchy and dynamic relations in a soccer match. The hierarchical organization of a soccer match (top) (leaf nodes, i.e., ball and players, are not shown), along with the approximate spatial mapping of players (numbered nodes) to the soccer ground (bottom).

varying graph. A detailed discussion of this technique is provided in the original publication [46].

In this section, the approach is applied to dynamic data of team sports. In detail, a soccer match of the 2D Soccer Simulation League World Championship 2010 is used. Therefore, the required features, such as the positions of the ball and the players, are known and need not to be extracted by the *feature extraction* stage, which would be the case for video visualization of real soccer matches.

For the soccer match, the following data aspects are considered:

- For hierarchical organization, the root node represents the match; both teams are on the second level. The team parts—goalie, defense, midfield, and offense—build the third level of the hierarchy. The players and the ball are the leaf nodes (see Figure 6.37).

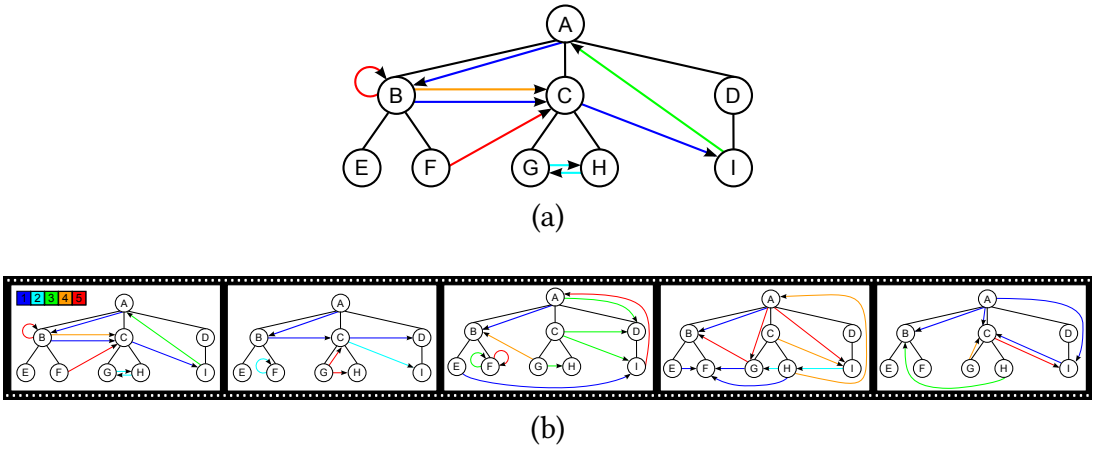


Figure 6.38 — Node-link diagrams: (a) inclusion relations (black lines without arrowheads) and directed adjacency relations (colored lines with black arrowheads) in a node-link diagram; (b) a sequence of graphs as an animated node-link diagram.

- The relationship "Euclidian distance" between the players and the ball on the soccer ground are used as adjacency edges and their weights. For the non-leaf nodes, the arithmetic means of all their elements are computed first, and then the Euclidian distances to the position of any other hierarchical element is used. This leads to a dense graph.
- The dynamics of the data is determined by the sample rate of the Soccer Simulation League (alternatively: the frame rate of the video), which provides the positions in 100 millisecond intervals.

6.6.1 Data Representation

An information hierarchy is modeled as a tree in a graph theoretic sense as $T = (V, E_I)$. The set V contains the vertices of the tree and $E_I \subset V \times V$ denotes the set of edges that express parent-child relationships among $v \in V$, which are termed here *inclusion edges* (see Figure 6.38 (a)). They are separate from the directed *adjacency edges* that form cross relations between any kind of hierarchy vertices—leaf or inner vertices—and hence build a graph $G = (V, E_A)$ (see Figure 6.38 (a)). The adjacency edges are modeled as $E_A \subseteq V \times V$. An ordered list of weights $W_i = w_1, \ldots, w_{m_i}$, $w_j \in \mathbb{R}_0^+$, $1 \leq j \leq m_i$, $m_i \in \mathbb{N}^+$ is attached to each $e_i \in E_A$. If the list of weights contains more than one element, a multi-edge is present, and a single edge in the other case.

Furthermore, a sequence of n multi-graphs with directed and weighted adjacency edges is modeled in a tree T as a sequence G_i, $1 \leq i \leq n$ of single graphs (see Figure 6.38 (b)).

The inclusion edges are required to be constant in this sequence.

6.6.2 The Visualization Technique

Representing dynamic relational data in information hierarchies in a static diagram leads to the question of how different data dimensions can be visualized simultaneously. Effective encoding of graph vertices and edges, as well as their directions and weights is required. Furthermore, multi-edges may occur that all might have different weights. Vertices are hierarchically organized, which leads to another type of structured relations. The representation of the time dimension among the graph edges is another challenge when using static diagrams.

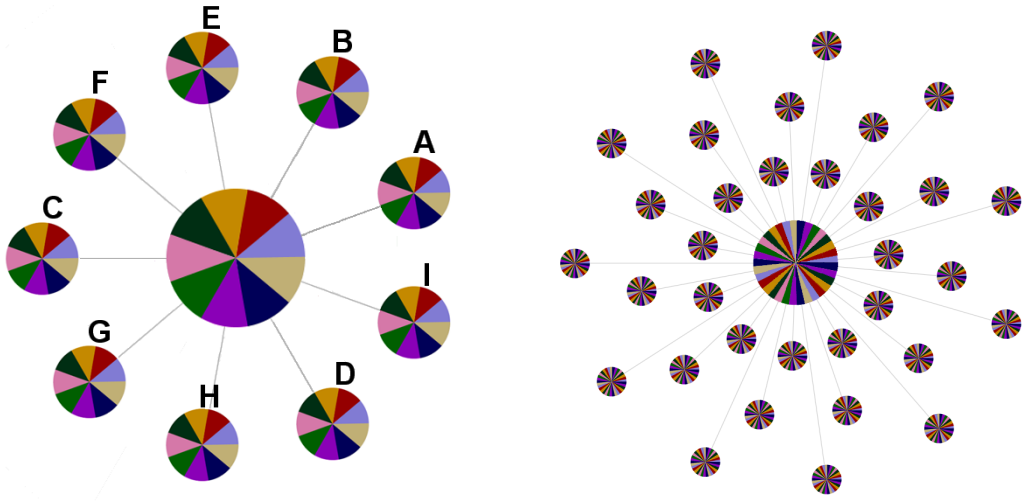
The following describes how all these different dimensions in the data can be visually encoded in a single static diagram with the goal to reduce visual clutter and to allow the user to detect trends, countertrends, periodic behaviors, temporal shifts, and anomalies in dynamic relational data between different hierarchy levels.

Implicit Link Representation

The layered TimeRadarTrees use indirect encoding of correspondence between vertices by common orientation in a glyph. The glyph is designed as a circular area that is divided into as many sectors as vertices have to be represented (see Figure 6.39 (a)). To achieve visual separation of sectors, color-coding is applied to discriminate the appearance of neighboring sectors. The idea is to make visual objects as distinct as possible on several feature channels by utilizing the perceptual grouping property of color [297]. This glyph, termed *CenterWheel*, is located in the center of the visualization.

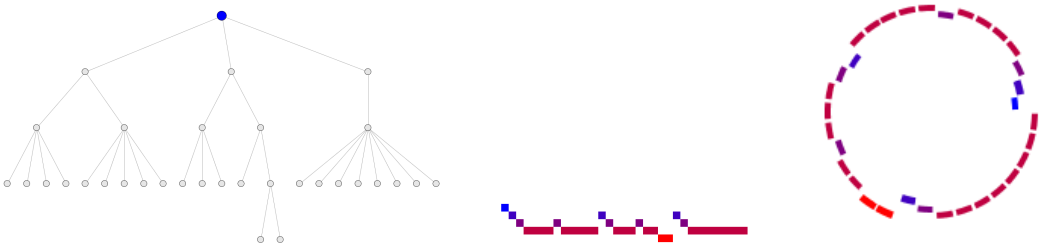
In addition to the CenterWheel, the visualization adopts the concept of small multiples [286, 26]: the implicit link representation contains smaller copies, termed *ThumbWheels*, that surround the CenterWheel. The circle sectors of the CenterWheel and ThumbWheels correspond to each other. Additionally, the ThumbWheel located in the direction of the circle sector (according to the center of the CenterWheel) represents the corresponding vertex. For example, vertex B in Figure 6.39 (a) is represented by the red colored circle sector of the CenterWheel as well as the red colored circle sectors of each ThumbWheel, and the corresponding ThumbWheel is located in the sectors direction.

A large number of ThumbWheels may lead to overlapping and occlusion. Therefore, ThumbWheels can be placed on different radial layers; a guiding line connects each circle sector of the CenterWheel to the corresponding ThumbWheel (see Figure 6.39 (b)).



(a) (b)

Figure 6.39 — *CenterWheel* and *ThumbWheels*: (a) The large circle in the center—the CenterWheel—is divided into nine differently colored sectors. Nine miniature thumbnail copies—the ThumbWheels—generate a contextual view. (b) A contextual view for a larger node set may exploit several layers to display all ThumbWheels.



(a) (b) (c)

Figure 6.40 — An example hierarchy consisting of 33 nodes displayed as (a) rooted node-link diagram, (b) indented tree plot, and (c) radial indented tree plot.

Hierarchy Representation

To represent adjacency relations between hierarchy levels, the radial indented tree plot adopts the linear indented tree plot technique [44]. These plots are well suited for this purpose because they are essentially one-dimensional zigzag curves that can be deformed to fit tightly around the CenterWheel (see Figure 6.40).

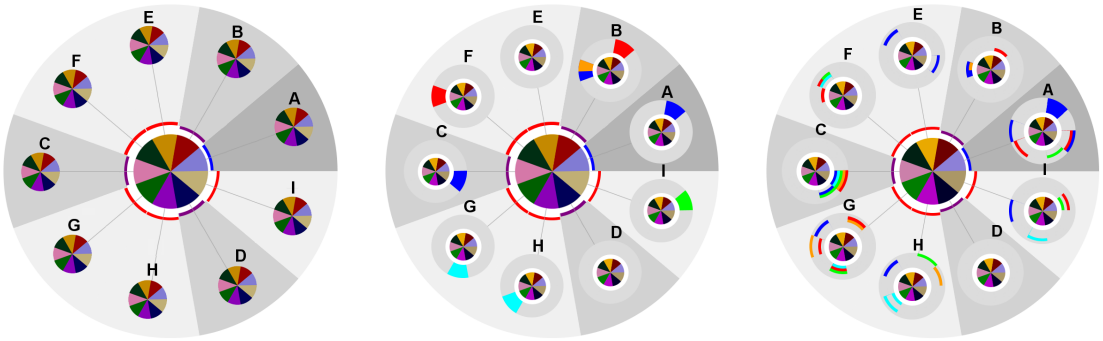


Figure 6.41 — (a) The hierarchical organization of the nodes of Figure 6.38 is shown as a radial indented tree plot with additional color-coding that depends on the depth of each hierarchical element. (b) Directed relations of the static multi-graph of Figure 6.38 are visually encoded as color-coded sectors and put on top of each representative ThumbWheel at the corresponding circle sectors. (c) The dynamic graph from Figure 6.38 is visually encoded as color-coded sectors.

The indented plot uses a blue-to-red color gradient whose value depends on the depth of a hierarchical element. Indentation expresses the node's depth in the tree. Similarly, radial indentation (further away from the circle center) shows that a node is on a deeper level in the hierarchy. The root node of a radial indented plot is placed to the rightmost position of a circle and the following nodes are processed counter-clockwise and in depth-first order. This ordering of nodes determines also the order of sectors in the CenterWheel and the according configuration of the ThumbWheels. In this way, sectors and indented tree elements that point in the same direction also refer to the same node.

The contextual view from Figure 6.39 (a) is extended by this hierarchical information. The radial indented tree plot is displayed between the CenterWheel and the ThumbWheels. Each hierarchical element is oriented according to the direction of the circle sector (see Figure 6.41 (a)). The hierarchical organization of the graph nodes is additionally expressed by elongated grayish sectors, where the gray level depends on the depth of the corresponding hierarchy element.

Static Graph Representation

Each adjacency edge in a multi-graph consists of a start and a target node, and additionally of a list of weights attached to it. When an edge originates at node A and points to node B, this edge is represented as a color-coded *adjacency arc* attached to the

ThumbWheel that represents node A; the adjacency arc is located in the sector of the ThumbWheel that points into direction of node B. To support multi-edges, this sector is subdivided into as many subsectors as edges are available. The weights $w_1, \ldots, w_n$ are represented by color-coding applied to the adjacency arcs.

An example that shows the node-link diagram of Figure 6.38 (a) is depicted in Figure 6.41 (b). The adjacency edge starting at node B and ending at node C is a multi-edge consisting of two differently weighted edges. All edges of this graph are converted to circle sectors and placed on top of the respective ThumbWheels.

Dynamic Graph Representation

The main benefit of the proposed visualization is the capability to display dynamics. Figure 6.38 (b) shows a sequence of five graphs in an animated node-link diagram with color-coded adjacency edges and additional hierarchical information. To map the dynamic graph data, the representation of the static case is extended: displaying a sequence of graphs is implemented by layering edges, where time starts in the circle center and grows radially outwards. The time-dependent relations lead to temporally varying weights that are also shown by color mapping. The increased display space of outer sectors puts focus on newer relations.

Figure 6.41 (c) is the visualization result of the dynamic graph of Figure 6.38 (b). Here, trends in the time series can easily be detected. For instance, a whole stack of adjacency arcs for the ThumbWheel of node A is colored blue. Using the contextual view, we identify that this sector corresponds to node B. Hence, there is a dynamic edge from node A to node B with constant weight throughout the sequence. Another insight may be gained from the ThumbWheel sector from node C to node I: the weight of the edge increases over time. Node D is the only node without any outgoing edge in the sequence. Furthermore, the hierarchical information displayed as a radial indented tree plot helps understanding between which hierarchy levels relations occur. Using animated node-link diagrams, or even a sequence of node-link diagrams placed side-by-side as in Figure 6.38 (b), identifying such insights would be difficult.

Interactive Features

The visualization additionally supports several interactive features: expanding and collapsing hierarchy levels, changing the color-coding of the CenterWheel and the ThumbWheels, applying different color-codings for the adjacency relations, details-on-demand, zooming of ThumbWheels, changing the radial distance of ThumbWheels, selecting time intervals, and switching from outgoing to incoming relations.

6.6.3 Visualization Results

The visualization applied to the soccer match introduced above is depicted in Figure 6.42 and provides an overview of the whole game (6000 time steps). The hierarchy consists of 34 elements. It is represented as radial indented plot with blue-to-red color-coding. The vegetation color mapping (see Figure 6.42 top) is used to indicate the distances between nodes (i.e., players and ball, team segments, etc.). Zero-valued distances are encoded in light gray, which is common for self-relations. The ThumbWheels are organized in two layers due to their amount. Circular text labels are attached to each ThumbWheel.

From this visualization, various insights can be obtained:

- The diagram is dominated by blue color. This expresses that distances between hierarchical elements are mainly small. Four ThumbWheels show primarily green to yellow color, which indicates that the corresponding players are farther away from the others. A closer inspection of this phenomenon reveals that these ThumbWheels belong to the goalies of both teams and the hierarchy container of their corresponding team parts.
- Offense players have mostly minimal distances to their defending counterparts.
- The midfield players of the Helios 2010 team are mainly in blue: they are closer to other players and further apart from the two goalies, whose corresponding ThumbWheel sectors are green. The midfield players of the WrightEagle team show a different behavior. Player number eight has a larger distance to the defending players of the Helios 2010 team, which indicates that he is defensively-minded. The other two players of the WrightEagle's midfield do not share this behavior. These two players, number six and seven, feature a larger distance to their own defense. Hence, those two midfield players are more offensively-minded than the midfield of their opponent.
- The ThumbWheels show periodic behavior in the distances. This originates from different actions during a soccer match: offense and defense. The ThumbWheel of the ball exhibits that the midfield and the offense of the Helios 2010 team are most frequently close to the ball. Thus, we may infer that these players are more often in ball possession.

Applying interaction to receive details-on-demand unveils further insights. Figure 6.43 (a) displays a scaled ThumbWheel of the ball, showing an interval of 100 seconds. The yellow and red color encodings unveil that the ball is often more distant from the goalies than from other players. The upper right circle quarter in blue color indicates that the ball is close to the goalkeeper and the defending players of Helios 2010, which is also the case for the midfield players of WrightEagle. We may conjecture that the WrightEagle team is in ball possession and attacks the opposite team.

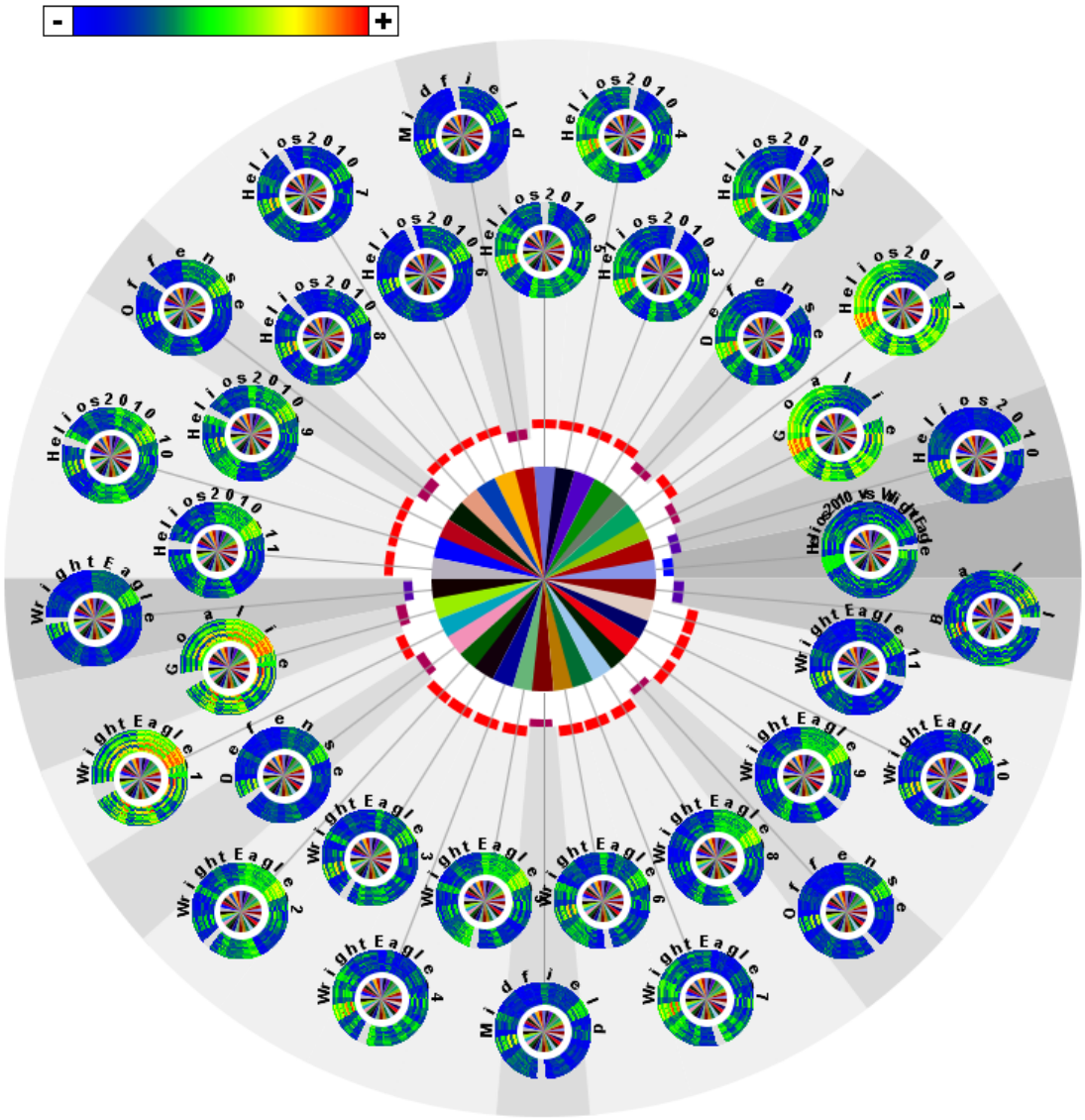
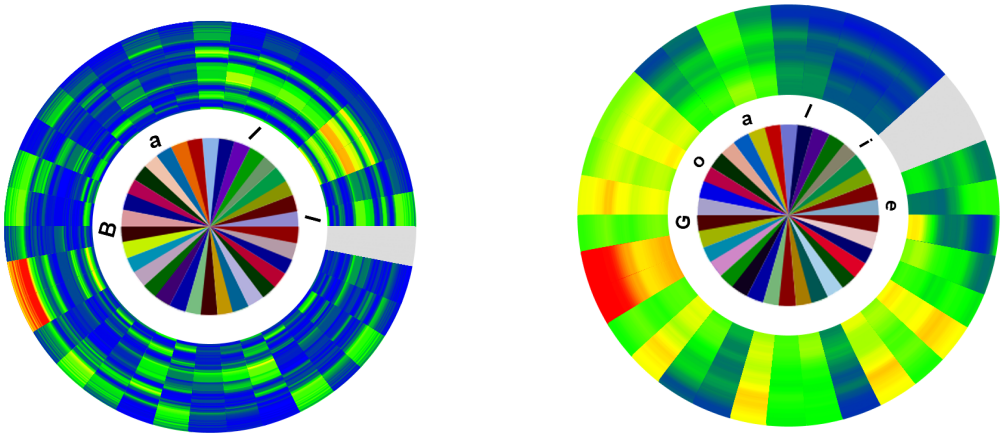


Figure 6.42 — Visualization of the 2D Soccer Simulation League World Championship 2010 final: Helios2010 vs. WrightEagle. Adjacency arcs show the Euclidian distances between players, team segments, and the ball during the soccer match.

The enlarged ThumbWheel of the goalie of the Helios 2010 team in the same time interval is illustrated in Figure 6.43 (b). Compared to the ThumbWheel of the ball, the color transitions are smoother. The ball frequently changes its position abruptly since it can be kicked with high speed. Two sectors change from yellow to red, which indicates that the distance from the Helios 2010 goalie to the opposite goalie (and its corresponding hierarchy element) is increasing.



(a) (b)

Figure 6.43 — The enlarged ThumbWheel view for two graph nodes: the ball (a), and the goalkeeper of Helios 2010 (b).

6.6.4 Conclusion

This section introduced a video visualization that is solely based on extracted information, organized in a dense time-varying directed and weighted multi-graph with additional hierarchical organization. This visualization can also be interpreted as an example of a visualization from another domain—dynamic graphs—applied to video data.

In detail, a novel technique that combines indented tree plots [44] with TimeRadarTrees [43], termed layered TimeRadarTrees was presented. The layered TimeRadarTrees display time-varying directed multi-graphs in information hierarchies. The approach uses an implicit link representation to encode graphs visually. Relations are shown by adjacency arcs stacked on top of corresponding ThumbWheel sectors. The information hierarchy is displayed as radial indented tree plot that allows visualizing relations between hierarchical elements. The cognitive effort is kept low for exploration of dynamics by representing the dynamic data in a static diagram. Additionally, the mental map is preserved by displaying the graph nodes at a fixed position. In contrast to node-link diagrams of dense graphs, visual clutter is marginal since edges cannot cross. Moreover, the approach does not need time-consuming layout algorithms. The graphs can be created on-the-fly without changing the graph layout.

The technique was applied to a dataset of the 2D Soccer Simulation League, which consists of 34 nodes including a hierarchical organization, and is a time-series with 6000 sample points. The weights of edges are calculated by Euclidian distances for each

pair of players, team parts, and the ball. Trends, countertrends, periodic behaviors, temporal shifts, and anomalies are easily observable in the diagram.

In future work, the strengths of implicit link representation as in the proposed approach should be evaluated and compared to explicit link representation as in node-link diagrams.

In general, visualization of video data can benefit from visualizations of other domains, as shown in this example. Similar to the integration of the features (that can be extracted from video) into the proposed graph visualization, the integration of the features into other domains are possible. Nevertheless, the selection of features as well their integration plays a key role for applicability.

6.7 Sonification

A key element for efficient video surveillance is situational awareness as outlined in the discussion in Chapter 2.2. Besides the characteristics of human perception, such as change blindness, inattentional blindness, and the short period of attention (see Chapter 2.2), which complicate online monitoring tasks for CCTV operators, their attention may also be distracted by further responsibilities they have. According to Gill et al. [110] and Keval [162], these duties include the logging of incidents, communication with individuals inside and outside the control room, tape management, preparation of working copies for further investigation or evidence to the court, and controlling the entry/exit of the control room itself. Further, human needs, such as toilet breaks or the break for a smoke may interrupt continuous surveillance. Gill et al. [110] observed control room operators being away from their screens in approximately 20 % of their shift time.

Situational awareness can benefit from sonification—the auditory pendant to visualization—of surveillance video data that commonly lacks audio tracks. This is due to the omnidirectional and ubiquitous nature of human sound perception. In contrast to vision, hearing does not require a particular direction of the listener's head-body configuration to perceive a desired signal. This allows the user to move freely while listening.

However, humans are not just aware of a situation because an acoustic signal reaches their ears. The question rather is, to which extent humans can handle multiple tasks and "split" their attention and processing resources among these. This question is addressed by the multiple-resource theory. Dual-task experiments indicated that structural dichotomies (e.g., such as visual and auditory processing) behave like separate resources [302]. This finding led to the 4-dimensional multiple resources model that claims increasing interference between two tasks to the extent that they share processing stages (perception/cognition, response), sensory modalities (auditory, visual),

codes (verbal, spatial), and channels of visual information [303]. Hence, dual-task performance can benefit from the use of separate resources [30].

Hence, to alleviate the problem of dual-task interference, displays of different modalities should be used. The goal is to support situational awareness and reduce the workload of CCTV operators by complementing the video display by an auditory display.

In the following, two different sonification approaches are briefly outlined. The first approach is a parameter mapping sonification based on low-level features that can be reliably extracted by low-level computer vision algorithms (published in [128]). In the second approach, trajectories of moving objects extracted from surveillance video are sonified by auditory icons. They are mapped on a spatial auditory display to communicate location, direction, and velocity according to a *virtual listener* (published in [126]). Both approaches are built upon the assumption that only changes in video data are relevant for surveillance.

To get an impression of the different sonification approaches, example video clips for use with headsets are provided on the project's website[8].

6.7.1 Video Sonification by Parameter Mapping

The basic idea of the proposed parameter mapping sonification is depicted in Figure 6.44. The approach uses as feature the dense optical flow field of two subsequent video frames. Therefore, the *feature extraction* stage (see Chapter 3) utilizes the method of Horn and Schunck [144]. Additional to the motion vectors, the *feature extraction* stage calculates a running average background model for foreground segmentation of the video data. This step is necessary, since optical flow calculation is prone to errors in the presence of noise and coding artifacts. Motion vectors in background regions are neglected for further processing to reduce background noise and thus decrease obtrusiveness of the auditory display.

The video screen and the corresponding optical flow field is split into non-overlapping segments aligned in a regular grid as illustrated in Figure 6.44.

For each segment, the average length of the contained motion vectors is calculated. This value represents the extent of activity for each segment. Please note that both the number of moving pixels and the length of the motion vectors (i.e., the velocity) influence the activity value. Hence, there are three properties for each segment to be mapped to auditory parameters: the segment's horizontal coordinate, its vertical coordinate, and its activity.

There are many possible design choices for mapping the segment properties to sound parameters. However, preliminary own experiments considering the users' expectations suggest the use of:

[8] www.vis.uni-stuttgart.de/index.php?id=vva

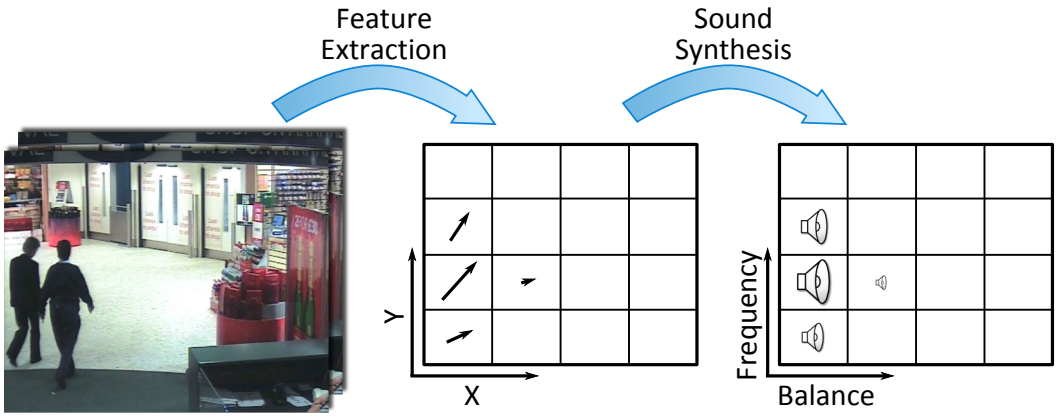


Figure 6.44 — Segment-based feature processing and mapping to auditory parameters.

- Balance to represent position's horizontal component
- Frequency to represent position's vertical component (rising frequency with increasing position)
- Amplitude to represent activity (low activity – soft sound)

The balance and frequency dimensions are quantized, whereas amplitude is a continuous parameter. The directional information of motion in the segments is neglected. However, the direction of object movement is indirectly encoded in the temporal transition of the amplitude level between neighboring segments. From another point of view, each segment can be regarded to play its own instrument that is defined by balance and frequency. If a segment shows no activity, the according instrument is muted. The complete orchestra of instruments represents the auditory display.

A key requirement of the auditory display is to convey the relevant information in an interpretable fashion. Additionally, the sonification has to be aesthetically pleasing to be non-obtrusive and broadly accepted [295]. A formative user study emphasized the importance of psychoacoustic aspects [128]. Therefore, psychoacoustic aspects are taken into account for the mapping and transfer function (for details see Höferlin et al. [128]).

Pure tones are perceived to be unnatural, thus complex tones are used to increase natural sound sensation. For sound synthesis, each segment is represented by a complex sine wave signal with the according fundamental frequency and seven overtones. Hence, the number of harmonic components considered in the sonification is $N_H = 8$. Only overtones that are whole multiples of the fundamental frequency are added to maintain pitch perception of complex sounds. Users can adjust the numbers of harmonics, if desired. However, although natural sounds generally have an arbitrary number of harmonics, their amplitude drops fast with higher harmonics. Thus, only

few are audible and necessary for an almost natural sound sensation. By adjusting the number of segments in each direction (horizontal and vertical), the users can trade the resolution and precision of the sonified information for the complexity of the produced soundscape.

The temporal sampling rate of the continuous sonification is set to the temporal resolution of the video data, and phases of the sine waves are adapted according to this rate to produce the impression of a continuous signal. Typically, the temporal resolution for auditory change detection is beyond 20 ms, even for low frequencies [217]. Hence, the temporal resolution of the human auditory system is capable of detecting sound changes between two successive frames to at least up to 50 fps, which is higher than the typical temporal sampling of online surveillance footage.

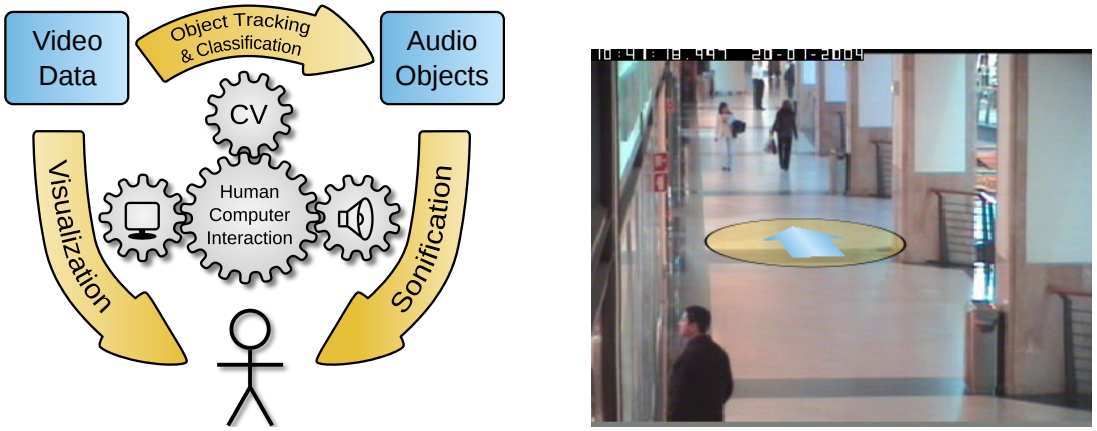
Summarizing, the approach produces a continuous sonic pattern or soundscape of the change in video data to leverage the change detection capabilities of the human auditory system. Recurrent changes in video generate an auditory texture that fades from attentional monitoring after some time of familiarization. In this state of background monitoring, sufficiently large changes of the auditory texture with respect to the familiar acoustic reference pattern reallocate attention, again. This is supported by research of the central auditory processing system that proved that *mismatch negativity* (MMN) is only elicited after a few repetitions of a standard stimulus and only if the deviation exceeds a particular threshold [219].

A user study was conducted and showed that participants are capable of identifying abnormal events by recognizing relevant deviations of the presented soundscape of this sonification. These results are a requisite to support surveillance operators and indicate that the proposed sonification can be used as component to support situational awareness. The evaluation also exhibited limitations of the approach, such as constraints on detection of multiple trajectories or accuracy limits for the estimation of fine movement. A consequence of these results is that such sonification may be used as supportive display enriching video screens.

More details on the approach, including a comprehensive discussion of the mappings and the results of the user study, are provided in the original publication [128].

6.7.2 Video Sonification by Auditory Icons

The sonification approach proposed in this section maps trajectories of moving objects to a spatial auditory display. It relies on trajectories from video footage that are extracted and classified into object categories in the *feature extraction* stage. The approach combines the spatial oriented auditory icons with the parameter mapping approach by adapting the sonic properties of an icon according to the properties of its trajectory. The structure of the proposed approach is depicted in Figure 6.45 (a).



(a) (b)

Figure 6.45 — (a) Structure of video sonification by auditory icons. Trajectories of moving objects extracted from video are sonified as auditory icons by a spatial auditory display. The users gain situational awareness by monitoring audio and video signals that complement each other. Users interact with the system by a graphical user interface to adapt parameters according to their task. (b) The interface of the auditory display is superimposed on a surveillance video from the CAVIAR dataset[9]. Moving objects in video are sonified by auditory icons according to their object category (e.g., persons sound like steps). The blue arrow represents the virtual listener's position and direction in the spatial auditory display aligned to the floor in the video. The yellow circle (projected to the ground-plane) indicates the maximum distance of auditory icons that sound with maximum volume. Auditory icons outside the circle follow a logarithmic sound attenuation.

In detail, each trajectory is represented by an auditory icon according to its object category. For example, trajectories that belong to the category "people" can be acoustically represented by footsteps, and "cars" may sound like an engine. The advantage of natural sounds over earcons or artificial sounds is that users are familiar with the sounds and know how to interpret them, without the need of training. Hence, category information is conveyed in a natural way in which the sounds build a familiar sonic ecology.

Information about the position, direction, and velocity of object trajectories are fundamental to gain situational awareness in surveillance scenarios. The relative position of auditory objects is presented to the users by a 2D spatial auditory display. Only two spatial dimensions are used due to a ground-plane assumption (i.e., all tracked objects are located on the ground-plane). This additionally avoids difficult elevation judgments

[9] EC Funded CAVIAR project/IST 2001 37540

of a sound source.

The auditory icons are displayed with respect to the *virtual listener*, which can be placed anywhere in the scene (see Figure 6.45 (b)). Moreover, users can explore the monitored area by moving around the virtual listener to find a sweet spot that supports situational awareness (e.g., the center of a road junction to monitor car-turning activity).

Further, parameter mapping is applied to the auditory icons to encode additional information of the trajectories, such as movement direction, and velocity. The properties of trajectories are mapped to the sound parameters in a way the users are familiar with. Therefore, a physical model is applied to the virtual 2D audio space that corresponds to real-world acoustics.

The spatial orientation of an auditory icon with respect to the properties of the moving object it represents can be regarded as a special form of parameter mapping. By this means, the location of a moving object is mapped to interaural level difference, interaural time difference, pinna reflections, etc. In the same way, properties such as the distance between object and listener are encoded by volume, sound roll-off, or early reverberation.

The impression of the movement direction and the velocity of an auditory object is created by frame-wise update of its position, according to the extracted trajectory. Additionally, movement direction and velocity of a trajectory are mapped to the Doppler effect. To amplify velocity perception further, users can choose to map object velocity to playback speed of the auditory icon, which for example approximates the well-known effect of an accelerating engine. The benefit of parameter mapping being analogous to real-world perception is that object recognition and situational awareness are facilitated.

Even when surround/binaural recordings of the surveillance scene are available, which is untypical as mentioned above, the proposed sonification has advantages over the recordings. An obvious benefit of the proposed method is that the virtual listener (steered by the CCTV operator) is able to move to any position in the 2D auditory space; the listener is not fixed to the position of the recording device. A second advantage of video sonification over playback of natural audio recordings is the abstraction of the audio content. Inconvenient background noise, such as the blowing of the wind, is avoided in sonification. Further, the proposed auditory display is rather schematic, which allows highlighting relevant parts in auditory perception. For example, auditory icons of a person and a bus might be represented at the same volume and could both be recognized by the operators. In natural environments, the bus would drown out the person, without the operators being aware of the person. In the same way, auditory icons of trajectories could be accentuated following the schematic illustration of cartoons. Figures in cartoons are reduced to their relevant elements and main objects are disproportionately highlighted. The cartoon metaphor can be used to emphasize

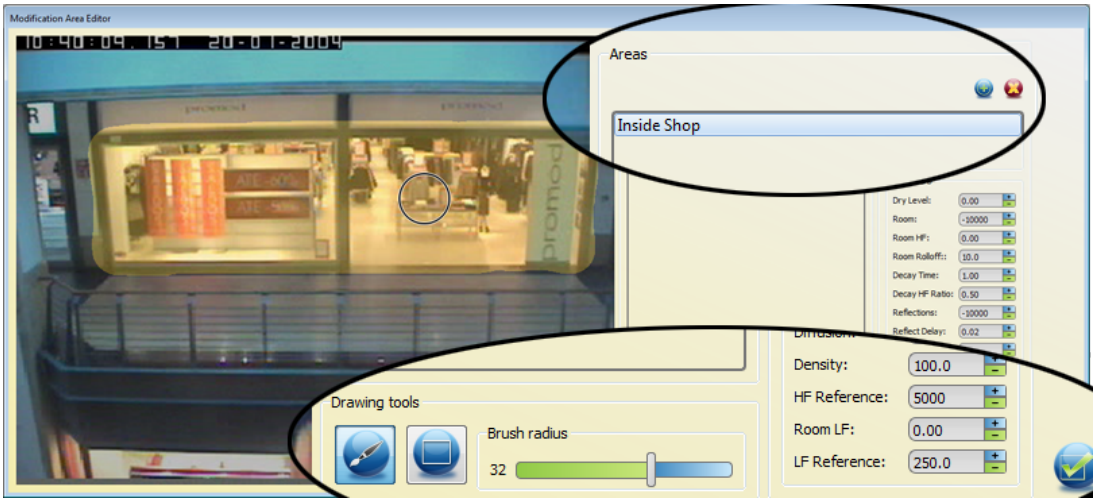


Figure 6.46 — Graphical user interface for the definition of a modification area. Left: specification of the region of influence by Photoshop-like brushing (the circle shows the brush). Right: specification of modification filters and their parameters.

important properties of an auditory object and to neglect irrelevant details, too.

Another concept introduced with this sonification approach is the *modification area*. Modification areas are user-defined regions in the video context that affect the sound properties of auditory icons located in these regions in 2D auditory space. The concept of modification areas is derived from the observations of real-world effects. Their counterparts in real-world could be green areas aside the pavement or space enclosed by walls. For example, if pedestrians walk from the pavement across the grass and back to the pavement, the sound of their steps will change, even though the steps remain the same. For instance, the sound turns muted and high frequencies are cut off, while walking over grass. We observe an analogous alteration of sound between an area enclosed by walls and space without walls. In the first case, we expect more reverberation than in the latter. However, the usage of modification areas is not restricted to natural effects. In fact, they can be used in an abstract and exaggerated way, too. For example, users can define restricted areas or virtual trip-wires to get an auditory alarm, if people move into a sterile zone or a dangerous area. The user interface for the definition of modification areas is depicted in Figure 6.46.

A user study that was conducted with 16 participants showed that this sonification approach by auditory icons leads to lower workload and, thus, to less stressful and more comfortable surveillance.

More details about this sonification approach, including the evaluation by a user study and discussion about benefits and shortcomings of the proposed sonification in the

light of psychology, cognitive science, and neuroscience, are provided in the original publication [126].

Chapter

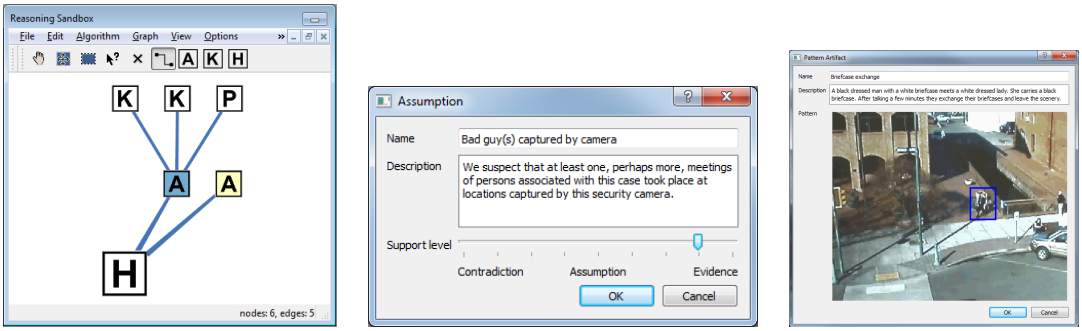
7 Reasoning Sandbox[1]

The *reasoning sandbox* (see Figure 7.1 (a)) supports the *human analysts* in their analytic discourse and sense-making. Primarily, it manages the reasoning artifacts generated during the foraging and sense-making process by the *human analysts*. Therefore, this is the only stage of the video visual analytics pipeline that is uncoupled from the linear stream processing mechanism. By providing overview of the different hypotheses, their support by evidence, alternative reasoning paths, and conflicting ideas, the *reasoning sandbox* helps maintain and assess situational awareness.

According to research agenda of visual analytics, the "analytic discourse is the technology-mediated dialogue between an analyst and his or her information to produce a judgment about an issue" [282]. The authors further describe the discourse to be an iterative and evolutionary process and the three classes of involved information of the analysts: (1) the issue, (2) corresponding information that the analysts gathered, and (3) the analysts' evolving knowledge.

The *reasoning sandbox* supports this iterative discourse and is therefore tightly coupled with the foraging loop. It visually organizes reasoning artifacts in all three levels of abstraction (closely following the reasoning artifacts defined in the research agenda of visual analytics [282]). The first level (see top of Figure 7.1 (a)) includes *relevant information*, *pattern artifacts*, and *higher-order knowledge constructs*. The second level consists of *assumptions* that can either be supported or refused by the top-level reasoning artifacts and, thus, may provide evidence or contradiction to the *hypotheses* in the third level. The degree of confidence of an assumption is modeled by its *level of support* (see Figure 7.1 (b)) that is judged by the analysts with respect to the related top

[1] Based on Höferlin et al. [132].



(a) (b) (c)

Figure 7.1 — The *reasoning sandbox* (a) shows the links between a couple of elementary reasoning artifacts, such as higher-order knowledge constructs (K), pattern artifacts (P), assumptions with their support level mapped to color (A), and hypotheses (H). (b) Dialog to set the level of support of an assumption. (c) Relevant information and pattern artifacts can be directly imported from other views or from the selection manager.

level artifacts. Highly supported and, thus, proved assumptions are called evidence following the reasoning terminology of the research agenda of visual analytics [282]. Assumptions that are not supported or refused by any reasoning artifact of the first level are considered as knowledge gaps. It is especially important for the analysts to keep these gaps in mind during hypothesis generation. That is why the need for separately handling assumptions and evidences are emphasized in the research agenda of visual analytics.

Within the sandbox, the reasoning process is represented as a graph of artifacts (nodes) connected with each other by supporting or refusing arguments, provided by the human analysts. To account for the separation of assumptions not proved and evidence, the visual appearance of an assumption changes its color with its level of support (see Figure 7.1 (a)) from yellow (level of support: not approved) to blue (approved (evidence)) or to red (disproved). Assumptions (as well as hypotheses) can further be inter-connected to highlight conflicting or competing assumptions (or hypotheses).

Within the foraging loop, data elements that seem to be of relevance for the current case are selected by the analysts and directly imported into the *reasoning sandbox* as relevant information artifacts (see Figure 7.1 (c)). Insight of patterns or structures in the data that is retrieved by visualization and data mining can be added to the *reasoning sandbox* as pattern artifacts. Together with the higher-order knowledge constructs that can be directly formulated by the analysts, these three types of reasoning artifacts form the *corresponding information gathered* by the analysts (2).

The analysts' evolving knowledge (3) is represented by the elements of the second and third levels as well as by the connections between them. These elements of the

second level are arguments of different levels of support. The third level consists of hypotheses, which are more complex reasoning constructs, such as complete scenarios.

After a sound and well-supported hypothesis was found that properly answers the issue (1), and competing hypotheses could be rejected, the final outcome of the analysis is produced (see Figure 2.3). This outcome is typically presented in the form of a dissemination product, i.e., a report of the analysis result and its reasoning path.

The implementation of the reasoning sandbox utilizes the information visualization framework *Tulip*[2] for graph rendering and interaction. The reasoning sandbox is fully integrated into the video visual analytics system. For example, when the users identify a pattern in a particular view, they can directly add it as pattern artifact to the reasoning sandbox (with an included screenshot of this view) and add further information, such as a name and a description of the pattern and its consequences. Additionally, the selection manager allows to import all selectable objects (e.g., object blobs or trajectories) as relevant information to the reasoning sandbox. The selection is bidirectional, which means that a selection of the object in the reasoning sandbox is also linked to the other views. Other reasoning artifacts, such as higher-order knowledge constructs, arguments, assumptions/evidence, and hypotheses are modeled in the reasoning sandbox.

[2] http://tulip.labri.fr/TulipDrupal/

Chapter

8 Conclusion and Future Directions[1]

8.1 Conclusion

In this thesis, the challenges of the relatively new area of video visual analytics were explored and addressed. These challenges mainly originate from

- the vast amount of video data that has to be analyzed,
- the complexity of the video data (i.e., the degrees of freedom in the environment, such as projection, illumination, and noise),
- and the quality of the search target definition (ill-defined vs. well-defined search targets).

To achieve reliable analysis in presence of these challenges where fully automatic or manual analysis fail, a visual analytics pipeline was proposed that takes advantage of the strength of both parts in an iterative loop. The iterative video visual analytics loop consists of a sense-making loop that is responsible for reasoning and deduction of reasoning artifacts collected in an integrated foraging loop (see research questions **R1** and **R2** in Chapter 1.1).

To obtain *data scalability*, the video visual analytics pipeline utilizes methods of *serialization*, *parallelization*, and *data reduction* for both human and machine (**R2**). Moreover, the pipeline was designed with the variety of tasks important to video analysis in mind, which can be divided into *status determination*, *event detection*, and *model generation*

[1] Chapter 8.2.2 is based on Höferlin et al. [132].

(**R1**). To demonstrate this *task scalability*, the video visual analytics pipeline was applied to different application areas (e.g., video surveillance, sports analysis), online (e.g., approaches for sonification) and offline analysis (e.g., fast-forward methods), as well as to exploratory (e.g., ISS) and known item search (e.g., filters). Another goal of the proposed pipeline and the presented methods for the particular stages was *situational awareness* (**R1**).

The foraging loop of the presented video visual analytics pipeline consists of six stages in which video data is stream processed before the results reach the *human analysts*. The stages are the selection of *data streams*, *manipulation*, *feature extraction*, *filtering*, *relevance measure*, and *visualization*. The *human analysts* can interact and configure each of these stages. Additionally, they can collect reasoning artifacts and organize them in the *reasoning sandbox*, which supports the sense-making loop (**R2**).

To answer the question, how these stages should be designed, and which techniques and methods suit their requirements well (**R3**), they were discussed in detail and various new methods were developed therefore.

In detail, the first two stages of the foraging loop allow the analysts to select video data and to apply *manipulators* to enhance its quality. The *feature extraction* stage calculates various features from raw video material that can be utilized in later stages. In this thesis, mainly mature computer vision algorithms were applied for trajectory extraction and feature extraction for specific application areas (e.g., snooker skill training). Additionally, state-of-the-art methods for automated low-level and high-level video analysis were surveyed.

The next stage in the pipeline, *filtering*, is mainly responsible for data reduction and, thus, important for data scalability. Therefore, a filter manager was proposed that enables users to arrange different filters in arbitrary combinations in a filter graph. One contribution of the thesis was the support of users in their filter definition process by applying four crucial interaction guidelines: easy-to-use filter definition, confidence-incorporated filter definition, decision-guided filter definition, and filter feedback. To ease up usage, users can choose to define filters *by properties*, *by sketch*, or *by example*. The filters show uncertainty information and allow incorporating the users' filter definition confidence. Context information and property statistics are provided wherever possible to guide the users' decisions. Another contribution of the thesis in this stage was the formulation of a configurable similarity metric for trajectories: three different similarity measures (*coverage*, *distance between means*, and *distance between standard deviations*) can be combined with a variety of facets of trajectories. Finally, a novel method for filtering by learning ad-hoc classifiers via an inter-active learning scheme was briefly outlined. The supported variety of filter definition possibilities is also important for task scalability.

In the *relevance measure* stage, the importance of data elements, i.e., the raw video data as well as derived features, is assessed. Besides the direct application of filters' *degree of membership* (DOM) as relevance measure, three methods were introduced to rate relevance frame-wise. The first method rates the importance according to the presence of trajectories (note, this stage is subsequent to the *filtering* stage). Additionally, an information-based relevance measure and a relevance measure based on a learned visual attention model from eye-tracking data were introduced.

The *visualization* stage, which was a focus in this thesis, prepares and communicates information to the *human analysts* by multiple coordinated views. Many views apply data mining and aggregation techniques tightly coupled with interaction techniques, and others render a static image for summarization. After a comprehensive video visualization review was provided, different views were proposed that support the users with different perspectives on the video data.

First, video visualization techniques for fast-forward playback of video were developed and evaluated in a controlled laboratory user study. The performance of *frame-skipping*, *temporal blending*, *object trail visualization*, and *predictive trajectory visualization* with respect to object identification and motion perception was studied. The results suggest using frame-skipping, which performed best in object identification, or predictive trajectory visualization, which was ranked best for motion perception, depending on the task and video. In crowded scenes, frame-skipping is the method of choice. Additionally, the capability to communicate playback speed feedback for adaptive fast-forward was determined for the three methods *speedometer*, *color frame*, and *analog VCR fast-forward*. All three methods performed well and no significant differences could be found.

Next, the *interactive schematic summaries* (ISS) for hierarchical video exploration were proposed. The method integrates visual data exploration with data mining techniques, and is based on scatter/gather browsing of trajectories. Here, trajectories are scattered into a particular set of clusters according to similarity criteria. In the gather stage, users select one or more clusters to be re-scattered. By iteratively applying these scatter/gather stages, users can explore video sequences and gain insight into structure, characteristics, and trends of object movements. For faceted exploration, the similarity criteria for clustering can be adjusted by the users. Additionally, a schematic visualization was introduced that summarizes the clusters according to the particular facets and similarity measures selected for clustering. The visualization avoids visual clutter by introducing and applying *trajectory bundling* and other simplifying cluster representation techniques.

Further, a couple of enhancements for the *VideoPerpetuoGram* (VPG) in different application areas were proposed in this thesis. For dynamic video visualization of tracked moving objects, an uncertainty mapping was introduced that shows the positional

uncertainties of trajectories as well as their relevances according to trajectory-based *relevance measures*. Additionally, the visualization of context information was improved by conveying a blob of the object additionally to the keyframes. Further, time can be skipped according to a frame-based *relevance measure*. In context of snooker skill training, a feasibility study was conducted in collaboration with a snooker club and a sports scientist. The question of the feasibility study was whether video visualization is suitable to aid snooker skill training. Therefore, the VPG was extended to a multi-attribute and multi-strand VPG that enables to display several features of multiple temporally synchronized videos side-by-side. The resulting static visualization focused on a typical cue action (spin avoidance). The study showed that snooker coaches can learn to interpret video visualization. Moreover, video visualization can be used in many aspects of snooker skill training. The subsequent investment of the snooker club in a ceiling-mounted camera system, and the successful request for financial sponsors showed the applicability of the approach.

With the *layered TimeRadarTrees*, a graph-based video visualization technique was presented that only depends on extracted features. Since the visualization displays time-varying directed and weighted multi-graphs in information hierarchies, the features were first organized appropriately. This example also showed how general data visualization techniques can be applied to visualize video.

To improve situational awareness in online monitoring tasks, two sonification approaches were discussed. The *video sonification by parameter mapping* approach sonifies motion in uniform grid segments. The amount of activity is mapped to the amplitude, where the horizontal and vertical position is mapped to balance and frequency, respectively. For *video sonification by auditory icons*, a *virtual listener* can be placed anywhere in the scene. Moving objects are sonified in a spatial auditory display by auditory icons relative to their position to the virtual listener. The auditory icons can be chosen with respect to the object category, such as steps for persons, or an engine sound for cars. The auditory icons are further adapted to the trajectories' speed. The introduction of *modification areas* allowed simulating real-world effects as well as to trigger auditory alarms when restricted areas are violated.

Finally, the *reasoning sandbox* was discussed that supports the human analysts in their analytic discourse and sense-making. With this tool, users can organize reasoning artifacts collected in the foraging loop by direct import from visualization views. By providing an overview of the various reasoning artifacts and their corresponding links, the support of hypotheses by evidence, contradictions, and alternative reasoning paths can be maintained and assessed to support analysis and to improve situational awareness.

8.2 Open Questions and Future Directions

In this section, remaining open questions and future research directions are discussed that go beyond the outlooks of the particular approaches that were presented in the corresponding sections.

8.2.1 Initial Skill Adaptation

Visual analytics systems often tackle complex tasks by applying various highly integrated visualizations and data mining approaches. Although these systems are largely used by domain experts, initial skill adaption is often slow and time-consuming. This observation was even made for browsing techniques that are part of the complete system, and thus smaller and less complex than the whole system. Consider, for example, the qualitative user study for initial feedback of the ISS: after introducing the technique for about 20 min, almost all participants mentioned that they would require by far more time to comprehend all options of the ISS completely. Based on the comments and own experiences, it can be estimated that at least a whole day would be required to learn how to select the right facets for different exploration scenarios.

However, this is only one jigsaw piece in a large compilation of different techniques provided for visual analytics of video data. Furthermore, complexity increases with the number of combinations of different methods utilized. The question that arises is: how can time spent for learning new visual analytics approaches be reduced? A simple yet effective answer might be standards. Thanks to GUI guidelines and style guides, a windows user may transfer the basic knowledge of using a GUI application to any other. In the same way, introduction of basic guidelines for visual analytics applications can provide a good foundation to transfer knowledge from one visual analytics system to another, even though particular visualization and data mining methods differ.

Especially visual analytics systems that address analysis support for non-expert casual users have to consider the ease of learning of their method, for example by assessing the learning effort in the evaluation of a visual analytics system.

8.2.2 Evaluation

An open question is how complex visual analytics systems can be evaluated at all. This emerging question is also subject of the bi-annual BELIV workshop[2], which has been established since 2006 and indicates its importance. Basically, there are two problems when evaluating complex system.

First, complex systems need expert users that are familiar with the system. Training of non-experts to achieve comparable results requires a substantial amount of time.

[2] http://www.beliv.org

Moreover, the participants of the user studies should not only be experts with the system, but even professionals in their application domain, which further reduces their availability. This leads to studies with only few participants and thus insufficient statistical significance.

Second, objective measurements of complex systems are impractical. The performance of a complex system depends on a combination of a vast amount of properties that cannot be assessed by a practical amount of participants. For example, an objective performance measurement conducted to evaluate the fast-forward visualizations included 24 participants. However, even for these simple visualizations, which are far less complex than highly interactive visual analytics systems with multiple coordinated views, significant differences between two pairs of visualizations could not be shown.

For this reason, the proposed video visual analytics system was validated following Ellis and Dix [85]. They claim that validation should consider two parts: justification and evaluation. While the justification of the approach can be found in the particular sections, the strategy pursued for the evaluation of the video visual analytics system can be summarized as:

- separation of smaller parts and evaluation by quantitative (objective and subjective) user studies (e.g., fast-forward visualizations, sonification approaches);
- evaluation of approaches with medium complexity by qualitative (subjective) user evaluation such as expert interviews or think aloud (e.g., ISS, information-based relevance measure);
- evaluation of complex systems by participation in challenges (e.g., VAST Challenge 2009).

Nevertheless, the overall performance of the system as a whole is not thoroughly evaluated, and further evaluation should be considered in future work. In this context, the application of longitudinal studies would be also interesting, especially with regard to reduce training time for foreign users to become experts for each evaluation.

8.2.3 Generalization: Visual Analytics of Streaming Data

The comprehensive analysis of a real-world issue is rarely restricted to video data. In general, complex problems require multiple heterogeneous information sources to draw reliable conclusions. Synchronization, registration, and fusion of data of different characteristics play a central role for successful analysis of realistic scenarios. For example, sports analysis may include ECG recordings from heart rate monitors in addition to video footage. Another example is given by the Grand Challenge of the IEEE VAST Challenge 2009 that demonstrated the close involvement of different data sources and data types in realistic analysis scenarios. Besides video data, badge data

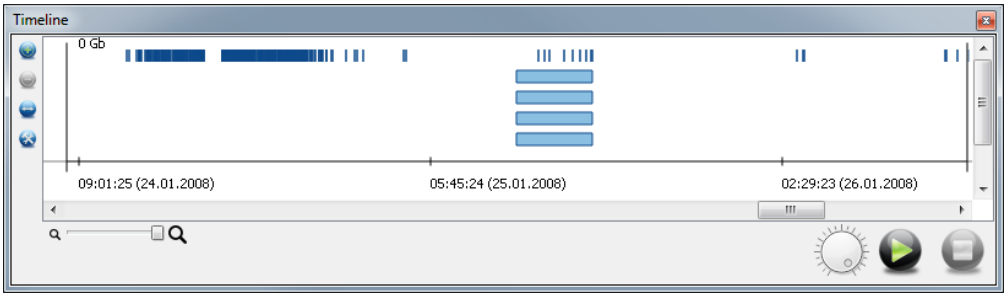


Figure 8.1 — Integration of multivariate data sources. The top row shows a badge data stream of several days. The four rows below show four streams of video data, which originated from a single surveillance camera that panned between four locations and was split accordingly. The duration of the video footage is about 8 hours and the different data sources are automatically synchronized by time.

(i.e., entrance and exit times of persons), computer network traffic, and social network data was provided.

For this purpose, the video visual analytics approach can be extended and generalized for visual analytics of arbitrary time-dependent streaming data, while the video visual analytics pipeline remains the same. The shared dimension—time—can be utilized to synchronize the data streams. In fact, the video visual analytics approach already has been extended and applied to three different applications with multivariate time-dependent data.

- The badge data from the IEEE VAST Challenge 2009 was integrated into the system and is automatically temporally synchronized with the video footage (see Figure 8.1).
- In the context of Nokia's Mobile Data Challenge 2012, the visual analytics approach was applied for the analysis of mobile data (published in [129]). Various events recorded by the mobile phone, such as call logs, system messages, Bluetooth connections, and camera usage, were considered besides trajectories that were extracted from GPS samples.
- Another application was dynamic evacuation planning. Analysis of a building's design and effective evacuation planning in emergency situations can benefit from visual analytics approaches. Therefore, the information of different sensors (e.g., heat detectors, smoke detectors, motion detectors, and surveillance cameras) or their simulation data must be integrated (published in [245]).

In the context of generalization, a trade-off between a generic visual analytics system and a focused and specialized system was experienced. A generic system that supports

any type of data may be applied to solve any type of task. However, a high degree of freedom in analysis methods to choose from and data sources to combine comes at the cost of complexity, both for human and machine. Simply put, the more general the processing model gets, the more complex it is—and thus slower. Specialized and fast processing algorithms often require a constrained environment and processing model. The presented video visual analytics system was designed in consideration of this trade-off. To support all types of video analysis tasks (see Chapter 2) the system pays the price of stream processing that introduces many obstacles for fast processing. Usage of the generic stream processing model inhibits the acceleration of data processing by exploiting background knowledge about a specific type of data. The open question arises if this trade-off can be alleviated, particularly with regard to more complex tasks that demand a higher level of abstraction and generality.

Bibliography

[1] IEEE VAST Challenge 2009, IEEE Symposium on Visual Analytics Science and Technologies (VAST) 2009. Last access: 08.03.2013. [Online]. Available: http://hcil.cs.umd.edu/localphp/hcil/vast/index.php [pages 35 and 131]

[2] Cisco visual networking index: Forecast and methodology 2011-2016. White paper, Cisco, 2012. [page 1]

[3] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58, 2006. [page 33]

[4] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building Rome in a day. In *Proceedings of IEEE International Conference on Computer Vision*, pages 72–79, 2009. [page 31]

[5] J. K. Aggarwal and Q. Cai. Human motion analysis: a review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999. [page 137]

[6] T. Ahonen, A. Hadid, and M. Pietikainen. Face recognition with local binary patterns. In *Proceedings of European Conference on Computer Vision*, pages 469–481, 2004. [page 29]

[7] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011. [pages 76 and 80]

[8] G. Andrienko, N. Andrienko, and S. Wrobel. Visual analytics tools for analysis of movement data. *ACM SIGKDD Explorations Newsletter*, 9(2):38–46, 2007. [page 19]

[9] G. Andrienko, N. Andrienko, I. Kopanakis, A. Ligtenberg, and S. Wrobel. Visual analytics methods for movement data. In D. P. Fosca Giannotti, editor, *Mobility, Data Mining and Privacy*, Geographic Knowledge Discovery, chapter 13, pages 375–408. Springer, 2008. [page 18]

[10] G. Andrienko, N. Andrienko, U. Demsar, D. Dransch, J. Dykes, S. Fabrikant, M. Jern, M. Kraak, H. Schumann, and C. Tominski. Space, time and visual analytics. *International Journal of Geographical Information Science*, 24(10):1577–1600, 2010. [page 17]

[11] N. Andrienko, G. Andrienko, and P. Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages and Computing*, 14(6):503–541, 2003. [page 17]

[12] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. [page 31]

[13] M. Aravecchia, S. Calderara, S. Chiossi, and R. Cucchiara. A videosurveillance data browsing software architecture for forensics: from trajectories similarities to video fragments. In *Proceedings of the ACM Workshop on Multimedia in Forensics, Security and Intelligence*, pages 37–42, 2010. [page 110]

[14] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. [page 31]

[15] J. Assa, Y. Caspi, and D. Cohen-Or. Action synopsis: pose selection and illustration. *ACM Transactions on Graphics*, 24(3):667–676, 2005. [page 85]

[16] J. Assfalg, M. Bertini, and C. Colombo. Semantic annotation of sports videos. *IEEE Multimedia*, 9(2):52–60, 2002. [page 137]

[17] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane, and J. Meunier. Left-luggage detection using homographies and simple heuristics. In *Proceedings of IEEE International Workshop on Performance Evaluation in Tracking and Surveillance*, pages 51–58, 2006. [page 17]

[18] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proceedings of ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 1–16, 2002. [pages 43 and 44]

[19] W. Bailer and G. Thallinger. A framework for multimedia content abstraction and its application to rushes exploration. In *Proceedings of ACM International Conference on Image and Video Retrieval*, pages 146–153, 2007. [page 84]

[20] L. Ballan, G. J. Brostow, J. Puwein, and M. Pollefeys. Unstructured video-based rendering: interactive exploration of casually captured videos. *ACM Transaction on Graphics*, 29(4):87:1–87:11, 2010. [page 83]

[21] H. Barlow. Temporal and spatial summation in human vision at different background intensities. *The Journal of Physiology*, 141(2):337–350, 1958. [page 88]

[22] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein. Video tapestries with continuous temporal zoom. *ACM Transaction on Graphics*, 29(4):89:1–89:9, 2010. [page 84]

[23] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994. [pages 26, 27, and 37]

[24] P. Baudisch, E. Cutrell, and G. Robertson. High-density cursor: A visualization technique that helps users keep track of fast-moving mouse cursors. In *Proceedings of Interact*, pages 236–243, 2003. [page 91]

[25] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *Computer Vision - ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer, 2006. [page 28]

[26] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983, (translation from French 1967 edition). [page 151]

[27] E. Bertini and D. Lalanne. Surveying the complementary role of automatic data analysis and visualization in knowledge discovery. In *Proceedings of ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration*, pages 12–20, 2009. [page 59]

[28] E. Bertini and D. Lalanne. Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. *ACM SIGKDD Explorations Newsletter*, 11(2):9–18, 2010. [page 11]

[29] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. [page 30]

[30] D. Boles. *International Encyclopedia of Ergonomics and Human Factors*, chapter Multiple resources, pages 271–275. Taylor and Francis, 2001. [page 159]

[31] R. Borgo, M. Chen, E. Grundy, B. Daubney, G. Heidemann, B. Höferlin, M. Höferlin, H. Jänicke, D. Weiskopf, and X. Xie. A survey on video-based graphics and video visualization. In *Proceedings of Eurographics 2011 – State of the Art Reports*, pages 1–23, 2011. [pages 3, 4, 6, 23, and 75]

[32] R. Borgo, M. Chen, E. Grundy, B. Daubney, G. Heidemann, B. Höferlin, M. Höferlin, H. Jänicke, D. Weiskopf, and X. Xie. State of the art report on video-based graphics and video visualizations. *Computer Graphics Forum*, 31(8):2450–2477, 2012. [pages 3, 4, 6, 23, 75, 79, and 80]

[33] D. Borland and R. Taylor II. Rainbow color map (still) considered harmful. *IEEE Computer Graphics and Applications*, 27(2):14–17, 2007. [page 95]

[34] H. Bosch, J. Heinrich, B. Höferlin, M. Höferlin, S. Koch, C. Müller, G. Reina, and M. Wörner. Innovative filtering techniques and customized analytics tools. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 269–270, 2009. [pages 3, 6, and 20]

[35] R. P. Botchen, S. Bachthaler, F. Schick, M. Chen, G. Mori, D. Weiskopf, and T. Ertl. Action-based multifield video visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):885–899, 2008. [pages 41, 86, 130, 132, 134, and 138]

[36] J. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker: Description of the algorithm. Technical report, Intel Corp., Microprocessor Research Labs, 2000. [page 34]

[37] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006. [page 27]

[38] G. R. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 2008. [page 41]

[39] M. Broilo, N. Piotto, G. Boato, N. Conci, and F. De Natale. Object trajectory analysis in video indexing and retrieval applications. In *Video Search and Mining*, volume 287 of *Studies in Computational Intelligence*, pages 3–32. Springer, 2010. [pages 35 and 43]

[40] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005. [pages 26 and 27]

[41] S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 1937–1944, 2011. [page 27]

[42] P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching TV (using weakly aligned subtitles). In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2961–2968, 2009. [page 30]

[43] M. Burch and S. Diehl. TimeRadarTrees: Visualizing dynamic compound digraphs. *Computer Graphics Forum*, 27(3):823–830, 2008. [pages 148 and 157]

[44] M. Burch, M. Raschke, and D. Weiskopf. Indented pixel tree plots. In *Proceedings of International Symposium on Visual Computing*, pages 338–349, 2010. [pages 148, 152, and 157]

[45] M. Burch, J. Heinrich, N. Konevtsova, M. Höferlin, and D. Weiskopf. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Transactions on Visualization and Computer Graphics*, 17(6):2440–2448, 2011. [page 5]

[46] M. Burch, M. Höferlin, and D. Weiskopf. Layered TimeRadarTrees. In *Proceedings of International Conference on Information Visualisation*, pages 18–25, 2011. [pages 3, 5, 8, 75, and 149]

[47] M. Burch, G. Andrienko, N. Andrienko, M. Höferlin, M. Raschke, and D. Weiskopf. Visual task solution strategies in tree diagrams. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 169–176, 2013. [page 5]

[48] D. Burr and M. Morgan. Motion deblurring in human vision. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 264(1380):431–436, 1997. [page 97]

[49] V. Caglioti and A. Giusti. Recovering ball motion from a single motion-blurred image. *Computer Vision and Image Understanding*, 113(5):590 – 597, 2009. [page 137]

[50] Y. Caspi, A. Axelrod, Y. Matsushita, and A. Gamliel. Dynamic stills and clip trailers. *The Visual Computer*, 22(9):642–652, 2006. [page 90]

[51] S. Chang, L. Chen, Y. Chung, and S. Chen. Automatic license plate recognition. *IEEE Transactions on Intelligent Transportation Systems*, 5(1):42–53, 2004. [page 1]

[52] G.-C. Chao, Y.-P. Tsai, and S.-K. Jeng. Augmented keyframe. *Journal of Visual Communication and Image Representation*, 21(7):682–692, 2010. [page 93]

[53] D. Chekhlov, A. P. Gee, A. Calway, and W. Mayol-Cuevas. Ninja on a plane: automatic discovery of physical planes for augmented reality using visual SLAM. In *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–4, 2007. [page 32]

[54] C. Chen. Top 10 unsolved information visualization problems. *IEEE Computer Graphics and Applications*, 25(4):12–16, 2005. [page 59]

[55] M. Chen, R. Botchen, R. Hashim, D. Weiskopf, T. Ertl, and I. Thornton. Visual signatures in video visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1093–1100, 2006. [pages 39, 86, 132, 136, and 142]

[56] H. Cheng, X. Jiang, Y. Sun, and J. Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259–2281, 2001. [page 27]

[57] K. Cheng, S. Luo, B. Chen, and H. Chu. Smartplayer: User-centric video fast-forwarding. In *Proceedings of ACM International Conference on Human Factors in Computing Systems*, pages 789–798, 2009. [pages 67, 87, and 95]

[58] N. Chinchor, J. J. Thomas, P. C. Wong, M. G. Christel, and W. Ribarsky. Multimedia analysis + visual analytics = multimedia analytics. *IEEE Computer Graphics and Applications*, 30(5):52–60, 2010. [pages 3 and 16]

[59] P. Chiu, A. Girgensohn, and Q. Liu. Stained-glass visualization for highly condensed video summaries. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 2059–2062, 2004. [page 84]

[60] M. G. Christel. Supporting video library exploratory search: When storyboards are not enough. In *Proceedings of the International Conference on Content-based Image and Video Retrieval*, pages 447–456, 2008. [page 16]

[61] T. Cipra and R. Romera. Kalman filter with outliers and missing observations. *Test*, 6(2):379–395, 1997. [page 35]

[62] R. Class, D. Jordan, N. Lumpp, M. Richter, A. Koch, and M. Höferlin. Verfahren und Vorrichtung zum Vereinzeln von Bauteilen aus einem Behältnis, 2008. [Online]. Available: http://www.patent-de.com/20100422/DE102008052436A1.html, last access: 08.03.2013. Daimler AG. DE102008052436A1. Patent. [pages 5 and 39]

[63] R. Class, M. Höferlin, W. Klumpp, A. Koch, M. Richter, and M. Schreiber. Verfahren und Vorrichtung zum Vereinzeln von Bauteilen, 2009. [Online]. Available: http://www.patent-de.com/20100805/DE102009007024A1.html, last access: 08.03.2013. Daimler AG. DE102009007024A1. Patent. [pages 5 and 39]

[64] W. G. Clifford. *Winning Snooker*. Foulsham, 1981. [page 134]

[65] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3D urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision*, 78(2–3):121–141, 2008. [page 32]

[66] C. D. Correa and K.-L. Ma. Dynamic video narratives. *ACM Transactions on Graphics*, 29(3):88:1–88:9, 2010. [page 84]

[67] C. D. Correa, Y. H. Chan, and K.-L. Ma. A framework for uncertainty-aware visual analytics. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 51–58, 2009. [page 132]

[68] J. Coughlan, A. Yuille, C. English, and D. Snow. Efficient deformable template detection and localization without user initialization. *Computer Vision and Image Understanding*, 78(3):303–319, 2000. [page 27]

[69] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007. [page 27]

[70] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings*

of ACM SIGIR Conference on Research and Development in Information Retrieval, pages 318–329, 1992. [pages 109 and 116]

[71] N. Dalal and B. Triggs. Histograms of orientated gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005. [pages 29 and 30]

[72] G. Daniel and M. Chen. Video visualization. In *Proceedings of IEEE Visualization 2003*, pages 409–416, 2003. [pages 41 and 85]

[73] F. W. Davis and D. S. Simonett. GIS and remote sensing. In D. Maquire, editor, *Geographical Information Systems*, chapter 14, pages 191–213. John Wiley & Sons Ltd, 1991. [page 18]

[74] A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6): 1052–1067, 2007. [page 32]

[75] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. Structure from motion without correspondence. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 557–564, 2000. [page 31]

[76] H. Denman, N. Rea, and A. Kokaram. Content-based analysis for video from snooker broadcasts. *Computer Vision and Image Understanding*, 92(2–3):176–195, 2003. [page 138]

[77] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 126–133, 2000. [page 31]

[78] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. H. Hebert. An empirical study of context in object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1271–1278, 2009. [page 30]

[79] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proceedings of IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005. [pages 29 and 30]

[80] T. D'Orazio, M. Leo, P. Spagnolo, M. Nitti, N. Mosca, and A. Distante. A visual system for real time detection of goal events during soccer matches. *Computer Vision and Image Understanding*, 113(5):622–632, 2009. [page 137]

[81] P. Dragicevic, G. Ramos, J. Bibliowitcz, D. Nowrouzezahrai, R. Balakrishnan, and K. Singh. Video browsing by direct manipulation. In *Proceedings of ACM International Conference on Human Factors in Computing Systems*, pages 237–246, 2008. [page 82]

[82] J.-D. Durou, M. Falcone, and M. Sagona. Numerical methods for shape-from-shading: a new survey with benchmarks. *Computer Vision and Image Understanding*, 109(1):22–43, 2008. [page 33]

[83] J. Dykes and D. Mountain. Seeking structure in records of spatio-temporal behaviour: Visualization issues, efforts and applications. *Computational Statistics and Data Analysis*, 43(4):581–603, 2003. [page 19]

[84] D. Ebert, C. Morris, P. Rheingans, and T. Yoo. Designing effective transfer functions for volume rendering from photographic volumes. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):183–197, 2002. [page 139]

[85] G. Ellis and A. Dix. An explorative analysis of user evaluation studies in information visualisation. In *Proceedings of AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization*, pages 15–21, 2006. [page 176]

[86] A. Endert, C. Andrews, G. A. Fink, and C. North. Professional analysts using a large, high-resolution display. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 273–274, 2009. [page 53]

[87] M. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995. [page 15]

[88] M. Endsley. Designing for situation awareness in complex systems. In *Proceedings of International Workshop on Symbiosis of Humans, Artifacts and Environment*, pages 1–13, 2001. [page 15]

[89] J. Engelhardt. Skizzenbasierte Trajektoriensuche in Videos. Diploma thesis no. 3269, Institute for Visualization and Interactive Systems, University of Stuttgart, 2012. [pages 45, 49, 52, and 57]

[90] C. Everton. *Snooker & Billiards: Technique * Tactics * Training*. Crowood Press, 1991. [page 134]

[91] M. D. Fairchild. *Color Appearance Models*. Wiley-IS&T, 2005. [page 89]

[92] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *Proceedings of European Conference on Computer Vision*, pages 379–393, 1998. [page 32]

[93] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37, 1996. [pages 11 and 77]

[94] S. Fels and K. Mase. Interactive video cubism. In *Proceedings of ACM Workshop on New Paradigms in Information Visualization and Manipulation*, pages 78–82, 1999. [page 85]

[95] P. Felzenszwalb. Representation and detection of deformable shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):208–220, 2005. [page 27]

[96] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. [page 30]

[97] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70:41–54, 2006. [page 32]

[98] P. Ferguson, C. Gurrin, H. Lee, S. Sav, A. Smeaton, N. O'Connor, Y.-H. Choi, and H. Park. Enhancing the functionality of interactive TV with content-based multimedia analysis. In *Proceedings of IEEE International Symposium on Multimedia*, pages 495 –500, 2009. [page 16]

[99] R. Fisher. The PETS04 surveillance ground-truth data sets. In *Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–5, 2004. [page 2]

[100] D. Fleet and A. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990. [page 27]

[101] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centers of circular features. In *Proceedings of ISPRS Intercommission Workshop on Fast Processing of Photogrammetric Data*, pages 281–305, 1987. [page 28]

[102] A. Francois, R. Nevatia, J. Hobbs, R. Bolles, and J. Smith. VERL: An ontology framework for representing and annotating video events. *IEEE Multimedia*, 12 (4):76–86, 2005. [page 12]

[103] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *Proceedings of IEEE International Conference on Computer Vision*, pages 80–87, 2009. [page 32]

[104] J. Gall, N. Razavi, and L. V. Gool. On-line adaption of class-specific codebooks for instance tracking. In *Proceedings of British Machine Vision Conference*, pages 55:1–55:12, 2010. [page 31]

[105] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *Proceedings of British Machine Vision Conference*, pages 195–204, 1998. [page 27]

[106] D. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999. [page 137]

[107] W. S. Geisler. Motion streaks provide a spatial code for motion direction. *Nature*, 400(6739):65–68, 1999. [page 89]

[108] D. Gibson, N. Campbell, and B. Thomas. Visual abstraction of wildlife footage using Gaussian mixture models and the minimum description length criterion. In *Proceedings of IEEE International Conference on Pattern Recognition*, pages 814–817, 2002. [page 28]

[109] J. Giesen, K. Mueller, E. Schuberth, L. Wang, and P. Zolliker. Conjoint analysis to measure the perceived quality in volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1664–1671, 2007. [page 100]

[110] M. Gill, A. Spriggs, J. Allen, M. Hemming, P. Jessiman, D. Kara, J. Kilworth, R. Little, and D. Swain, 2005. Control room operation: Findings from control room observations. On-line Research, Development and Statistics publication, Home Office UK, last access: 08.03.2013. [Online]. Available: `http://rds.homeoffice.gov.uk/rds/pdfs05/rdsolr1405.pdf` [pages 12 and 158]

[111] A. Girgensohn, D. Kimber, J. Vaughan, T. Yang, F. Shipman, T. Turner, E. Rieffel, L. Wilcox, F. Chen, and T. Dunnigan. DOTS: Support for effective video surveillance. In *Proceedings of ACM International Conference on Multimedia*, pages 423–432, 2007. [page 82]

[112] D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz. Schematic storyboarding for video visualization and editing. *ACM Transactions on Graphics*, 25(3):862–871, 2006. [pages 85, 93, and 118]

[113] D. B. Goldman, C. Gonterman, B. Curless, D. Salesin, and S. M. Seitz. Video object annotation, navigation, and composition. In *Proceedings of ACM Symposium on User Interface Software and Technology*, pages 3–12, 2008. [page 82]

[114] C. Goodwin and M. H. Goodwin. Seeing as situated activity: Formulating planes. In D. M. Yrjö Engeström, editor, *Cognition and Communication at Work*, pages 61–95. Cambridge University Press, 1996. [page 12]

[115] C. Graham and R. Margaria. Area and the intensity-time relation in the peripheral retina. *American Journal of Physiology*, 113(2):299–305, 1935. [page 88]

[116] O. Grau, A. Hilton, J. Kilner, G. Miller, T. Sargeant, and J. Starck. A free-viewpoint video system for visualisation of sport scenes. *SMPTE Motion Imaging Journal*, 116(5–6):213–219, 2007. [page 138]

[117] D. Green. Regional variations in the visual acuity for interference fringes on the retina. *The Journal of Physiology*, 207(2):351–356, 1970. [page 95]

[118] M. Green, J. Reno, R. Fisher, L. Robinson, A. General, N. Brennan, D. General, J. Travis, R. Downs, and B. Modzeleski. The appropriate and effective use of security technologies in US schools: A guide for schools and law enforcement agencies series: Research report. Technical report, National Institute of Justice, 1999. [page 15]

[119] T. Griffiths. Snooker – basic skills [vhs]. Clear Vision Video Studio., 1996. [page 134]

[120] G. Grinstein, J. Scholtz, M. Whiting, and C. Plaisant. VAST 2009 challenge: An insider threat. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 243–244, 2009. [page 3]

[121] R. Güting and M. Schneider. *Moving Objects Databases*. Morgan Kaufmann, 2005. [page 19]

[122] S. Hannuna, N. Campbell, and D. Gibson. Identifying quadruped gait in wildlife video. In *Proceedings of IEEE International Conference on Image Processing*, pages 713–716, 2005. [page 26]

[123] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of Alvey Vision Conference*, pages 147–151, 1988. [page 28]

[124] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. [page 32]

[125] A. Hauptmann, M. Christel, and R. Yan. Video retrieval based on semantic concepts. *Proceedings of the IEEE*, 96(4):602 –622, 2008. [page 16]

[126] B. Höferlin, M. Höferlin, M. Raschke, G. Heidemann, and D. Weiskopf. Interactive auditory display to support situational awareness in video surveillance. In *Proceedings of International Conference on Auditory Display*, 2011. [pages 3, 5, 8, 75, 78, 159, and 165]

[127] B. Höferlin, M. Höferlin, D. Weiskopf, and G. Heidemann. Information-based adaptive fast-forward for visual surveillance. *Multimedia Tools and Applications*, 55(1):127–150, 2011. [pages 3, 4, 7, 65, 67, 69, 71, 72, 73, 75, 78, 87, 93, and 98]

[128] B. Höferlin, M. Höferlin, B. Goloubets, G. Heidemann, and D. Weiskopf. Auditory support for situation awareness in video surveillance. In *Proceedings of International Conference on Auditory Display*, 2012. [pages 3, 5, 8, 75, 78, 159, 160, and 161]

[129] B. Höferlin, M. Höferlin, and J. Räuchle. Visual analytics of mobile data. In *Proceedings of Nokia Mobile Data Challenge Workshop*, 2012. [pages 5, 77, and 177]

[130] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, and G. Heidemann. Interactive learning of ad-hoc classifiers for video visual analytics. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pages 23–32, 2012. [pages 3, 4, 7, 43, and 63]

[131] B. Höferlin, H. Pflüger, M. Höferlin, G. Heidemann, and D. Weiskopf. Learning a visual attention model for adaptive fast-forward in video surveillance. In *Proceedings of International Conference on Pattern Recognition Applications and Methods*, volume 2, pages 25–32, 2012. [pages 3, 4, 7, 65, 67, 71, 73, and 98]

[132] B. Höferlin, M. Höferlin, D. Weiskopf, and G. Heidemann. Scalable video visual analytics. *Information Visualization Journal*, 12(3):(to appear), 2013. [pages 1, 3, 4, 6, 9, 23, 43, 65, 75, 167, and 171]

[133] M. Höferlin, A. Koch, R. Class, J. Krammer, N. Lumpp, M. Richter, and H.-G. Ziegler. Verfahren und Vorrichtung zum Vereinzeln von Bauteilen, 2008. [Online]. Available: `http://www.patent-de.com/20100422/DE102008052440A1.html`, last access: 08.03.2013. Daimler AG. DE102008052440A1. Patent. [pages 5 and 39]

[134] M. Höferlin, B. Höferlin, and D. Weiskopf. Video visual analytics of tracked moving objects. In *Proceedings of Workshop on Behaviour Monitoring and Interpretation*, pages 59–64, 2009. [pages 3 and 6]

[135] M. Höferlin, A. Koch, and M. Richter. Verfahren und Vorrichtung zum Ermitteln einer Teilfläche eines Bauteils, 2009. [Online]. Available: `http://www.patent-de.com/20100916/DE102009009569A1.html`, last access: 08.03.2013. Daimler AG. DE102009009569A1. Patent. [pages 5 and 39]

[136] M. Höferlin, E. Grundy, R. Borgo, D. Weiskopf, M. Chen, I. W. Griffiths, and W. Griffiths. Video visualization for snooker skill training. *Computer Graphics Forum*, 29(3):1053–1062, 2010. [pages 3, 4, 5, 7, 23, 75, and 144]

[137] M. Höferlin, B. Höferlin, D. Weiskopf, and G. Heidemann. Uncertainty-aware video visual analytics of tracked moving objects. *Journal of Spatial Information Science*, 2:87–117, 2011. [pages 1, 3, 4, 5, 6, 23, 38, 43, 65, and 75]

[138] M. Höferlin, B. Höferlin, D. Weiskopf, and G. Heidemann. Interactive schematic summaries for exploration of surveillance video. In *Proceedings of ACM International Conference on Multimedia Retrieval*, pages 9:1–9:8, 2011. [pages 3, 5, 7, and 75]

[139] M. Höferlin, K. Kurzhals, B. Höferlin, G. Heidemann, and D. Weiskopf. Evaluation of fast-forward video visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2095–2103, 2012. [pages 3, 4, 8, and 75]

[140] M. Höferlin, B. Höferlin, G. Heidemann, and D. Weiskopf. Interactive schematic summaries for faceted exploration of surveillance video. *IEEE Transactions on Multimedia*, 2013, (to appear), doi: 10.1109/TMM.2013.2238521. [pages 3, 4, 5, 7, 43, and 75]

[141] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Transaction on Graphics*, 24(3):577–584, 2005. [page 33]

[142] D. Holten and J. van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983–990, 2009. [pages 119, 120, and 121]

[143] J. Honovich, 2009. Security manager's guide to video surveillance, IPVideoMarket.info, V3. Last access: 08.03.2013. [Online]. Available: `http://ipvideomarket.info/book` [pages 2 and 14]

[144] B. Horn and B. Schunck. Determining optical flow. *Computer Vision*, 17(1–3): 185–203, 1981. [pages 26 and 159]

[145] K. Hornsby and M. Egenhofer. Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence*, 36(1):177–194, 2002. [page 19]

[146] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *Proceedings of IEEE International Conference on Computer Vision*, pages 128–135, 2009. [page 30]

[147] C.-H. Huang, M.-Y. Shih, Y.-T. Wu, and J.-H. Kao. Loitering detection using Bayesian appearance tracker and list of visitors. In *Proceedings of Advances in Multimedia Information Processing*, pages 906–910, 2008. [page 17]

[148] Y. Huang, C. Chen, C. Tsai, C. Shen, and L. Chen. Survey on block matching motion estimation algorithms and architectures with new results. *The Journal of VLSI Signal Processing*, 42(3):297–320, 2006. [page 26]

[149] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proceedings of IEEE International Conference on Computer Vision*, pages 605–611, 1995. [page 84]

[150] M. Jerian, S. Paolino, F. Cervelli, S. Carrato, A. Mattei, and L. Garofano. A forensic image processing environment for investigation of surveillance video. *Forensic Science International*, 167(2-3):207–212, 2007. [page 17]

[151] R. C. Jones, D. DeMenthon, and D. S. Doermann. Building mosaics from video using MPEG motion vectors. In *Proceedings of ACM International Conference on Multimedia*, pages 29–32, 1999. [page 84]

[152] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106 – 1122, 2007. [page 27]

[153] H. Kang, X. Chen, Y. Matsushita, and X. Tang. Space-time video montage. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1331–1338, 2006. [page 83]

[154] U.-P. Käppeler, M. Höferlin, and P. Levi. 3D object localization via stereo vision using an omnidirectional and a perspective camera. In *Proceedings of IEEE Workshop on Omnidirectional Robot Vision*, pages 7–12, 2010. [pages 5 and 39]

[155] T. Karrer, M. Weiss, E. Lee, and J. Borchers. Dragon: A direct manipulation interface for frame-accurate in-scene video navigation. In *Proceedings of ACM International Conference on Human Factors in Computing Systems*, pages 247–250, 2008. [page 82]

[156] W. Ke, C. R. Sugimoto, and J. Mostafa. Dynamicity vs. effectiveness: Studying online clustering for scatter/gather. In *Proceedings of ACM Conference on Research and Development in Information Retrieval*, pages 19–26, 2009. [page 116]

[157] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proceedings of IEEE International Conference on Computer Vision*, volume 1, pages 166–173, 2005. [pages 29 and 30]

[158] D. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in visual data analysis. In *Proceedings of IEEE International Conference on Information Visualization*, pages 9–16, 2006. [pages 9 and 66]

[159] D. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. In *Visual Data Mining*, volume 4404 of *Lecture Notes in Computer Science*, pages 76–90. Springer, 2008. [pages 3 and 10]

[160] D. Keim, F. Mansmann, and J. Thomas. Visual analytics: How much visualization and how much analytics? *ACM SIGKDD Explorations Newsletter*, 11(2):5–8, 2010. [page 14]

[161] M. G. Kendall. *Rank Correlation Methods*. Charles Griffin and Company, 1962. [page 105]

[162] H. Keval. *Effective, Design, Configuration, and Use of Digital CCTV*. PhD thesis, University College London, 2009. [pages 12 and 158]

[163] H. Keval and M. Sasse. Can we ID from CCTV: Image quality in digital CCTV and face identification performance. In *Proceedings of Society of Photo-Optical Instrumentation Engineers*, volume 6982, pages 69820K–69820K–15, 2008. [page 18]

[164] H. Keval and M. A. Sasse. To catch a thief – you need at least 8 frames per second: The impact of frame rates on user performance in a CCTV detection task. In *Proceedings of ACM International Conference on Multimedia*, pages 941–944, 2008. [pages 18, 22, and 86]

[165] D. Kimber, T. Dunnigan, A. Girgensohn, F. Shipman, T. Turner, and T. Yang. Trailblazing: Video playback control by direct object manipulation. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1015–1018, 2007. [page 82]

[166] A. Klein, P. Sloan, A. Finkelstein, and M. Cohen. Stylized video cubes. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 15–22, 2002. [page 85]

[167] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007. [page 32]

[168] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 83–86, 2009. [page 32]

[169] I. Koprinska and S. Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16(5):477–500, 2001. [page 1]

[170] P. Koutsourakis, L. Simon, O. Teboul, G. Tziritas, and N. Paragios. Single view reconstruction using shape grammars for urban environments. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1795–1802, 2009. [page 33]

[171] M. Kraak. The space-time cube revisited from a geovisualization perspective. In *Proceedings of International Cartographic Conference*, pages 1988–1996, 2003. [page 17]

[172] M. Kristan, J. Perš, M. Perše, and S. Kovačič. Closed-world tracking of multiple interacting targets for indoor-sports applications. *Computer Vision and Image Understanding*, 113(5):598–611, 2009. [page 137]

[173] H. Kruegle. *CCTV Surveillance: Analog and Digital Video Practices and Technology*. Elsevier, Inc., 2nd edition, 2007. [page 18]

[174] A. Kuhn, T. Senst, I. Keller, T. Sikora, and H. Theisel. A Lagrangian framework for video analytics. In *Proceedings of IEEE Workshop on Multimedia Signal Processing*, pages 387–392, 2012. [page 18]

[175] B. Kuijpers and W. Othman. Trajectory databases: Data models, uncertainty and complete query languages. In *Database Theory – ICDT 2007*, volume 4353 of *Lecture Notes in Computer Science*, pages 224–238. Springer, 2006. [page 19]

[176] S. Kullback. *Information Theory and Statistics*. Wiley Publication in Mathematical Statistics, 1959. [page 68]

[177] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Object cut. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 18–25, 2005. [page 27]

[178] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. [page 33]

[179] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536, 2012. [page 87]

[180] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2–3):107–123, 2005. [pages 29 and 30]

[181] I. Laptev and T. Lindeberg. *Local Descriptors for Spatio-temporal Recognition*, volume 3667 of *Lecture Notes in Computer Science*, pages 91–103. Springer, 2006. [page 29]

[182] P. Laube, T. Dennis, P. Forer, and M. Walker. Movement beyond the snapshot–dynamic analysis of geospatial lifelines. *Computers, Environment and Urban Systems*, 31(5):481–501, 2007. [pages 19, 36, and 46]

[183] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994. [page 33]

[184] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3361–3368, 2011. [page 29]

[185] F. Lee and W. Bailer. Video browsing using object trajectories. In *Advances in Multimedia Modeling*, volume 6524 of *Lecture Notes in Computer Science*, pages 219–229. Springer, 2011. [page 110]

[186] H. Lee, A. F. Smeaton, C. Berrut, N. Murphy, S. Marlow, and N. E. O'Connor. Implementation and analysis of several keyframe-based browsing interfaces to digital video. In *Proceedings of European Conference on Research and Advanced Technology for Digital Libraries*, pages 206–218, 2000. [page 84]

[187] M.-C. Lee, W.-G. Chen, C. Lin, C. Gu, T. Markoc, S. Zabinsky, and R. Szeliski. A layered video object coding system using sprite and affine motion model. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):130–145, 1997. [page 84]

[188] M. Leo, T. D Orazio, A. Caroppo, T. Martiriggiano, and P. Spagnolo. Automatic monitoring of forbidden areas to prevent illegal accesses. In *Pattern Recognition and Image Analysis*, volume 3687 of *Lecture Notes in Computer Science*, pages 635–643. Springer, 2005. [page 17]

[189] S. Leyk, R. Boesch, and R. Weibel. A conceptual framework for uncertainty investigation in map-based land cover change modelling. *Transactions in GIS*, 9 (3):291–322, 2005. [page 1]

[190] F. Li, A. Gupta, E. Sanocki, L. He, and Y. Rui. Browsing digital video. In *Proceedings of ACM International Conference on Human Factors in Computing Systems*, pages 169–176, 2000. [page 82]

[191] T. W. Liao. Clustering of time series data–a survey. *Pattern Recognition*, 38(11): 1857–1874, 2005. [page 116]

[192] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998. [page 28]

[193] C.-B. Liu and N. Ahuja. Vision based fire detection. In *Proceedings of IEEE International Conference on Pattern Recognition*, pages 134–137, 2004. [page 17]

[194] Y. Liu, J. Mostafa, and W. Ke. A fast online clustering algorithm for scatter/gather browsing. Technical Report TR-2007-06, Chapel Hill, NC, USA: UNC School of Information and Library Science, 2007. [page 116]

[195] A. Lobay and D. A. Forsyth. Shape from texture without boundaries. *International Journal of Computer Vision*, 67(1):71–91, 2006. [page 33]

[196] H. M. Lomell. Targeting the unwanted: Video surveillance and categorical exclusion in Oslo, Norway. *Surveillance & Society*, 2(2/3):346–360, 2002. [page 12]

[197] P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind. *Geographical Information Systems and Science*. John Wiley & Sons Ltd, 2005. [page 18]

[198] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. [page 28]

[199] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conference on Artificial Intelligence*, volume 2, pages 674–679, 1981. [page 26]

[200] L. Lucchese and S. Mitra. Colour image segmentation: A state-of-the-art survey. *Proceedings of the Indian National Science Academy*, 67(2):207–221, 2001. [page 27]

[201] H. Luo, J. Fan, J. Yang, W. Ribarsky, and S. Satoh. Exploring large-scale video news via interactive visualization. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 75–82, 2006. [page 17]

[202] H. Luo, J. Fan, S. Satoh, J. Yang, and W. Ribarsky. Integrating multi-modal content analysis and hyperbolic visualization for large-scale news video retrieval and exploration. *Signal Processing: Image Communication*, 23(7):538–553, 2008. [page 17]

[203] A. Mack. Inattentional blindness: Looking without seeing. *Current Directions in Psychological Science*, 12(5):180–184, 2003. [page 15]

[204] D. Magee and J. Pers. Computer vision based analysis in sport environments. *Computer Vision and Image Understanding*, 113(5):589–662, 2009. [page 137]

[205] B. Majecka. Statistical models of pedestrian behaviour in the forum. Master's thesis, School of Informatics, University of Edinburgh, 2009. [pages 112, 113, and 128]

[206] F. Mansmann, F. Fischer, and D. A. Keim. Dynamic visual analytics – facing the real-time challenge. In *Expanding the Frontiers of Visual Analytics and Visualization*, pages 69–80. Springer, 2012. [page 19]

[207] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The next frontier for innovation, competition, and productivity. White paper, McKinsey Global Institute, 2011. [page 1]

[208] D. Mark and M. Egenhofer. Geospatial lifelines. In *Integrating Spatial and Temporal Databases. Dagstuhl Seminar Report*, volume 228, 1998. [page 18]

[209] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004. [page 28]

[210] A. McIvor. Background subtraction techniques. In *Proceedings of International Conference on Image and Vision Computing New Zealand*, 2000. [page 27]

[211] T. Mei, B. Yang, S. Yang, and X. Hua. Video collage: Presenting a video sequence using a single image. *The Visual Computer*, 25(1):39–51, 2009. [page 85]

[212] S. Meyer. *Data analysis for scientists and engineers*. John Wiley & Sons, Inc., 1975. [page 38]

[213] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. [page 28]

[214] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1–2):43–72, 2005. [page 28]

[215] S. Mitaim and B. Kosko. What is the best shape for a fuzzy set in function approximation? In *Proceedings of IEEE International Conference on Fuzzy Systems*, pages 1237–1243, 1996. [page 55]

[216] T. Moons, L. J. V. Gool, and M. Vergauwen. 3D reconstruction from multiple images: Part 1 – principles. *Foundations and Trends in Computer Graphics and Vision*, 4(4):287–404, 2009. [page 32]

[217] B. C. J. Moore. Psychoacoustics. In T. D. Rossing, editor, *Springer Handbook of Acoustics*, pages 459–501. Springer Science+Business Media, LLC, 2007. [page 161]

[218] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. *International Journal of Computer Vision*, 73(3):263–284, 2007. [page 29]

[219] R. Näätänen, P. Paavilainen, T. Rinne, and K. Alho. The mismatch negativity (MMN) in basic research of central auditory processing: A review. *Clinical Neurophysiology*, 118(12):2544–2590, 2007. [page 161]

[220] J. Naisbitt. *Megatrends: Ten New Directions Transforming Our Lives*. Warner Books, 1982. [page 2]

[221] F. Navarro, F. J. Serón, and D. Gutierrez. Motion blur rendering: State of the art. *Computer Graphics Forum*, 30(1):3–26, 2011. [page 90]

[222] M. Nienhaus and J. Dollner. Depicting dynamics using principles of visual art and narrations. *IEEE Computer Graphics and Applications*, 25(3):40–51, 2005. [page 85]

[223] F. Nilsson. *Intelligent Network Video: Understanding Modern Video Surveillance Systems*. CRC Press. Taylor & Francis Group, 2009. [page 1]

[224] A. Nusimow. Intelligent video for homeland security applications. In *Proceedings of IEEE Conference on Technologies for Homeland Security*, pages 139–144, 2007. [pages 2 and 14]

[225] A. Pang, C. Wittenbrink, and S. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, 1997. [page 132]

[226] N. V. Patel and I. K. Sethi. Video shot detection and characterization for video databases. *Pattern Recognition*, 30(4):538–592, 1997. [page 24]

[227] K. Peker and A. Divakaran. Adaptive fast playback-based video skimming using a compressed-domain visual complexity measure. In *Proceedings of IEEE International Conference on Multimedia and Expo*, volume 3, pages 2055–2058, 2004. [pages 67 and 68]

[228] K. Peker, A. Divakaran, and H. Sun. Constant pace skimming and temporal sub-sampling of video using motion activity. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 414–417, 2001. [page 67]

[229] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of IEEE International Conference on Computer Vision*, pages 261–268, 2009. [page 31]

[230] M. Perše, M. Kristan, S. Kovačič, G. Vučkovič, and J. Perš. A trajectory-based analysis of coordinated team activity in a basketball game. *Computer Vision and Image Understanding*, 113(5):612–621, 2009. [page 137]

[231] N. Petrovic, N. Jojic, and T. Huang. Adaptive video fast forward. *Multimedia Tools and Applications*, 26(3):327–344, 2005. [page 67]

[232] D. Pfoser and C. Jensen. Capturing the uncertainty of moving-object representations. In *Advances in Spatial Databases*, volume 1651 of *Lecture Notes in Computer Science*, pages 111–131. Springer, 1999. [pages 36 and 37]

[233] M. Piccardi. Background subtraction techniques: A review. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3099–3104, 2004. [pages 27 and 93]

[234] G. Pingali, A. Opalach, Y. Jean, and I. Carlbom. Visualization of sports using motion trajectories: Providing insights into performance, style, and strategy. In *Proceedings of IEEE Visualization*, pages 75–82, 2001. [page 137]

[235] G. S. Pingali, A. Opalach, Y. D. Jean, and I. B. Carlbom. Instantly indexed multimedia databases of real world events. *IEEE Transactions on Multimedia*, 4 (2):269–282, 2002. [page 137]

[236] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis*, 2005. [pages 3, 9, 22, and 66]

[237] C. Poynton. *Digital Video and HDTV: Algorithms and Interfaces*. Morgan Kaufmann Publishers, 2003. [page 89]

[238] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg. Webcam synopsis: Peeking around the world. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2007. [page 83]

[239] Y. Pritch, A. Rav-Acha, and S. Peleg. Nonchronological video synopsis and indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11): 1971–1984, 2008. [pages 83, 84, and 110]

[240] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14(3): 294–307, 2005. [page 27]

[241] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (1):65–81, 2007. [page 31]

[242] G. Ramos and R. Balakrishnan. Fluid interaction techniques for the control and annotation of digital video. In *Proceedings of ACM Symposium on User Interface Software and Technology*, pages 105–114, 2003. [page 82]

[243] S. A. Rathus. *Psychology: Concepts & Connections*. Wadsworth Publishing, 10th edition, 2011. [page 18]

[244] A. Rav-Acha, Y. Pritch, and S. Peleg. Making a long video short: Dynamic video synopsis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 435–441, 2006. [page 83]

[245] R. Reddy, M. Höferlin, M. Dambier, and D. Weiskopf. Visual analytics for dynamic evacuation planning. In *Proceedings of International Workshop on Visual Analytics*, pages 13–17, 2012. [pages 5 and 177]

[246] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Detection of interest points using symmetry. In *Proceedings of IEEE International Conference on Computer Vision*, pages 62–65, 1990. [page 28]

[247] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context free attentional operators: The generalized symmetry transform. *International Journal of Computer Vision*, 14(2):119–130, 1995. [page 41]

[248] J. Ren, J. Orwell, G. A. Jones, and M. Xu. Tracking the soccer ball using multiple fixed cameras. *Computer Vision and Image Understanding*, 113(5):633–642, 2009. [page 137]

[249] R. Rensink, J. O'Regan, and J. Clark. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8(5):368–373, 1997. [page 15]

[250] A. Renyi. On measures of entropy and information. In *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 547–561, 1961. [pages 68 and 69]

[251] J. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proceedings of IEEE International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 61–71, 2007. [page 75]

[252] S. Romdhani and T. Vetter. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proceedings of IEEE*

Conference on Computer Vision and Pattern Recognition, volume 2, pages 986–993, 2005. [page 33]

[253] C. Rother, L. Bordeaux, Y. Hamadi, and A. Blake. Autocollage. *ACM Transaction on Graphics*, 25(3):847–852, 2006. [page 85]

[254] C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3001–3008, 2011. [page 32]

[255] K. Schoeffmann and L. Boeszoermenyi. Video browsing using interactive navigation summaries. In *Proceedings of IEEE International Workshop on Content-Based Multimedia Indexing*, pages 243–248, 2009. [pages 82 and 110]

[256] K. Scott-Brown and P. Cronin. An instinct for detection: Psychological perspectives on CCTV surveillance. *The Police Journal*, 80(4):287–305, 2007. [page 86]

[257] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In *Proceedings of ACM International Conference on Multimedia*, pages 357–360, 2007. [page 29]

[258] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2006. [page 33]

[259] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168, 2004. [page 27]

[260] A. A. Sheikh and E. R. Korn. *Imagery in Sports and Physical Performance*. Baywood Publishing Company Inc, 1994. [page 137]

[261] J. Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 593–600, 1994. [pages 26, 28, and 37]

[262] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages*, pages 336–343, 1996. [pages 66 and 110]

[263] B. Shneiderman. Inventing discovery tools: Combining information visualization with data mining. *Information Visualization*, 1(1):5–12, 2002. [page 59]

[264] W. Siler and J. Buckley. *Fuzzy Expert Systems and Fuzzy Reasoning*. Wiley-Blackwell, 2005. [page 45]

[265] H. Sion. Quadruped gait detection in low quality wildlife video. *PhD Thesis, Univeristy of Bristol*, 2007. [page 28]

[266] C. Snoek, V. Nedovic, M. van Liempt, R. van Balen, F. Yan, M. Tahir, K. Mikolajczyk, J. Kittler, M. de Rijke, J. Geusebroek, T. Gevers, M. Worring, A. Smeulders, and D. Koelma. The mediamill TRECVID 2008 semantic video search engine. In *Proceedings of TREC Video Retrieval Evaluation Workshop*, 2008. [page 17]

[267] J. Starck, A. Maki, S. Nobuhara, A. Hilton, and T. Matsuyama. The multiple-camera 3D production studio. *IEEE Transactions on Circuits and Systems for Video Technololgy*, 19(6):856–869, 2009. [page 33]

[268] F. Stein. Efficient computation of optical flow using the census transform. In *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 79–86. Springer, 2004. [page 27]

[269] E. Stoykova, A. A. Alatan, P. Benzie, N. Grammalidis, S. Malassiotis, J. Ostermann, S. Piekh, V. Sainov, C. Theobalt, T. Thevar, and X. Zabulis. 3D time-varying scene capture technologies – a survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(11):1568–1586, 2007. [page 31]

[270] T. Strothotte and S. Schlechtweg. *Non-Photorealistic Computer Graphics. Modelling, Rendering, and Animation.* Morgan Kaufmann Publishers, 2002. [page 118]

[271] S. Sull, J.-R. Kim, Y. Kim, H. S. Chang, and S. U. Lee. Scalable hierarchical video summary and search. In *Storage and Retrieval for Media Databases*, pages 553–561, 2001. [page 84]

[272] X. Sun, C.-W. Chen, and B. S. Manjunath. Probabilistic motion parameter models for human activity recognition. In *Proceedings of IEEE International Conference on Pattern Recognition*, volume 1, pages 463–446, 2002. [page 30]

[273] M. Svensson, C. Heath, and P. Luff. Monitoring practice event detection and system design. In S. Velastin and P. Remagnino, editors, *Intelligent Distributed Video Surveillance Systems*, volume 5 of *IEEE Professional Applications of Computing Series*, pages 1–30. The Institution of Engineering and Technology, 2006. [page 12]

[274] R. Szeliski. *Computer Vision: Algorithms and Applications.* Springer, 1st edition, 2011. [page 31]

[275] Y. Taniguchi, A. Akutsu, and Y. Tonomura. PanoramaExcerpts: Extracting and packing panoramas for video browsing. In *Proceedings of ACM International Conference on Multimedia*, pages 427–436, 1997. [page 84]

[276] A. Taylor, D. Miller, and B. Wynar. *Wynar's Introduction to Cataloging and Classification.* Libraries Unlimited Inc, 2000. [page 109]

[277] B. Taylor and C. Kuyatt. Guidelines for evaluating and expressing the uncertainty of NIST measurement results. NIST Technical Note 1297, United States Department of Commerce Technology Administration, National Institute of Standards and Technology, 1994. [page 37]

[278] L. Teodosio and W. Bender. Salient video stills: content and context preserved. In *Proceedings of ACM International Conference on Multimedia*, pages 39–46, 1993. [page 91]

[279] L. Teodosio and W. Bender. Salient stills. *ACM Transactions on Multimedia, Computing, Communications and Applications*, 1(1):16–36, 2005. [page 84]

[280] The YouTube Team, 2012. It's YouTube's 7th birthday... and you've outdone yourselves, again, YouTube, LLC, last access: 08.03.2013. [Online]. Available: http://youtube-global.blogspot.de/2012/05/its-youtubes-7th-birthday-and-youve.html [page 1]

[281] D. Thirde, L. Li, and F. Ferryman. Overview of the PETS2006 challenge. In *Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 47–50, 2006. [page 2]

[282] J. Thomas and K. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics.* IEEE Computer Society, 2005. [pages 3, 9, 11, 12, 14, 15, 57, 167, and 168]

[283] L. Torresani and C. Bregler. Space-time tracking. In *Proceedings of European Conference on Computer Vision*, pages 801–812, 2002. [page 32]

[284] L. Torresani, D. Yang, E. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 493–500, 2001. [page 32]

[285] B. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia, Computing, Communications and Applications*, 3(1), 2007. [pages 81, 83, and 84]

[286] E. Tufte. *Envisioning Information.* Graphics Press, 1990. [page 151]

[287] J. Tukey. *Exploratory data analysis.* Addison-Wesley, 1977. [page 11]

[288] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky. Video manga: Generating semantically meaningful video summaries. In *Proceedings of ACM International Conference on Multimedia*, pages 383–392, 1999. [page 84]

[289] R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 238–245, 2006. [page 31]

[290] G. Van Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision*, 47(1):275–285, 2002. [page 32]

[291] I. R. Vasiliev. Mapping time. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 34(2):1–51, 1997. [page 118]

[292] P. Viola and M. J. Jones. Robust real-time object detection. Technical Report CRL 2001/01, Cambridge Research Laboratory, 2001. [pages 29, 30, and 70]

[293] G. Vogiatzis, P. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 391–398, 2005. [page 32]

[294] N. J. Wade and M. T. Swanston. *Visual Perception: An Introduction*. Psychology Press Ltd., 2nd edition, 2001. [page 18]

[295] B. N. Walker and M. A. Nees. Theory of sonification. In T. Hermann, A. Hunt, and J. G. Neuhoff, editors, *The Sonification Handbook*, pages 9–39. Logos Publishing House, 2011. [page 160]

[296] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., 2nd edition, 2004. [pages 15, 87, 90, and 143]

[297] C. Ware. *Visual Thinking for Design*. Morgan Kaufmann Publishers Inc., 2008. [page 151]

[298] M. Ware, E. Frank, G. Holmes, M. Hall, and I. Witten. Interactive machine learning: Letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, 2001. [page 59]

[299] J. Weickert, A. Bruhn, T. Brox, and N. Papenberg. A survey on variational optic flow methods for small displacements. In *Mathematical Models for Registration and Applications to Medical Imaging*, volume 10 of *Mathematics in Industry*, pages 103–136. Springer, 2006. [page 27]

[300] D. Weiskopf and G. Erlebacher. Overview of flow visualization. In C. Hansen and C. Johnson, editors, *The Visualization Handbook*, pages 261–278. Elsevier, 2005. [page 93]

[301] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, 2004. [pages 31 and 34]

[302] C. Wickens. The structure of attentional resources. In R. Nickerson, editor, *Attention and Performance VIII*, pages 239–257. Lawrence Erlbaum Associates, 1980. [page 158]

[303] C. Wickens. Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, 3(2):159–177, 2002. [pages 78 and 159]

[304] B. Wildemuth, G. Marchionini, M. Yang, G. Geisler, T. Wilkens, A. Hughes, and R. Gruss. How fast is too fast?: Evaluating fast forward surrogates for digital video. In *Proceedings of ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 221–230, 2003. [page 83]

[305] K. Williams. *Snooker: Know the Game*. A & C Black, 2002. [page 134]

[306] A. D. Wilson and A. F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21 (9):884–900, 1999. [page 30]

[307] H. Winnemöller, S. C. Olsen, and B. Gooch. Real-time video abstraction. *ACM Transactions on Graphics*, 25(3):1221–1226, 2006. [page 118]

[308] J. Wolfe. Guided search 2.0 a revised model of visual search. *Psychonomic Bulletin & Review*, 1(2):202–238, 1994. [page 69]

[309] O. Wolfson. Moving objects information management: The database challenge (vision paper). In *Proceedings of International Workshop on Next Generation Information Technologies and Systems*, pages 15–26, 2002. [pages 19 and 54]

[310] M. Worring, C. Snoek, O. de Rooij, G. Nguyen, and A. Smeulders. The Mediamill semantic video search engine. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 1213–1216, 2007. [page 17]

[311] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997. [page 34]

[312] R. S. Wurman. *Information anxiety: What to do when information doesn't tell you what you want to know*. Bantam Books, 1989. [page 14]

[313] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005. [page 116]

[314] F. Yan, W. Christmas, and J. Kittler. Layered data association using graph-theoretic formulation with applications to tennis ball tracking in monocular sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10): 1814–1830, 2008. [page 137]

[315] M. Yeung and B.-L. Yeo. Video visualization for compact presentation and fast browsing of pictorial content. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(5):771–785, 1997. [page 84]

[316] L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965. [page 45]

[317] X. Zhao and Y. Liu. Generative tracking of 3D human motion by hierarchical annealed genetic algorithm. *Pattern Recognition*, 41(8):2470–2483, 2008. [page 31]

[318] S. Zhu and K. Ma. A new diamond search algorithm for fast block matching motion estimation. In *Proceedings of International Conference on Information, Communications and Signal Processing*, pages 292–296, 1997. [page 26]

[319] O. Zweigle, U.-P. Käppeler, H. Rajaie, K. Häußermann, R. Lafrenz, A. Tamke, F. Schreiber, M. Höferlin, M. Schanz, and P. Levi. CoPS stuttgart team description 2008. In *Proceedings of RoboCup International Symposium*, pages 1–8, 2008. [pages 5 and 39]