FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Tracking Related Multiple Targets in Videos

DISSERTATION

zur Erlangung des akademischen Grades

Doktor/in der technischen Wissenschaften

eingereicht von

Nicole M. Artner
Matrikelnummer 0727746

an der
Fakultät für Informatik der Technischen Universität Wien
Betreuung: O.Univ.Prof. Dipl.Ing. Dr.techn. Walter G. Kropatsch

Diese Dissertation haben begutachtet:

(O.Univ.Prof. Dipl.Ing. Dr.techn. Walter G. Kropatsch)

(Prof. Em. Dr. Horst Bunke)

Wien, 10.10.2013

(Nicole M. Artner)

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Tracking Related Multiple Targets in Videos

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor/in der technischen Wissenschaften

by

Nicole M. Artner
Registration Number 0727746

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: O.Univ.Prof. Dipl.Ing. Dr.techn. Walter G. Kropatsch

The dissertation has been reviewed by:

(O.Univ.Prof. Dipl.Ing. Dr.techn. Walter G. Kropatsch)	(Prof. Em. Dr. Horst Bunke)

Wien, 10.10.2013

(Nicole M. Artner)

Erklärung zur Verfassung der Arbeit

Nicole M. Artner
Bahngasse 1, 7312 Unterpetersdorf

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasserin)

Acknowledgments

I would like to thank all former and current people working at PRIP for their support. Thanks for being test subjects for my experiments, for giving feedback and for cheering me up. Special thanks goes to my mentor and *Doktorvater* Walter G. Kropatsch for his constant support, inspiring ideas and patience. Thank you for the countless hours we spent discussing the research presented in this thesis. I would also like to thank Adrian Ion for his invaluable scientific and emotional support. Thanks for the endless brainstorming sessions, late night paper submission sessions and philosophical discussions about the meaning of a PhD and life.

Deep appreciation goes to my partner Martin. Thank you for loving and supporting me all this years. Without you this thesis would not have seen the light of day. Last but not least, I would like to thank my parents Maria and Josef as well as my brother Armin for their love and care.

Abstract

This cumulative thesis presents research in the field of tracking. Tracking is one of the most thoroughly researched problems in computer vision. The aim of tracking is to follow an object of interest (target) in a video. In this thesis, I focus on a special problem: tracking related multiple targets. Two important questions in tracking are: *What is the target?* and *Where is the target?* The core contributions of this thesis answer these two questions with the help of graph-based representations and methods.

The first core contribution is a fully automatic initialization for target models (*What?*), based on the principal that *things which move together belong together*. The input of the approach is a video showing the targets in motion. In this video a set of salient points is tracked to extract the necessary motion information in the form of trajectories. A triangulated graph is built based on the initial positions of the tracked points. Then, the triangulated graph is deformed based on the motion encoded in the trajectories. This deformation of the triangulation over time is the input of a hierarchical grouping process, which is realized by an irregular dual graph pyramid. In the top level of the resulting pyramid the rigid entities (e.g. body parts of a human body) are identified. Finally, the motion of these rigid entities is analyzed to find possible points of articulation connecting them (e.g. upper and lower arm of a human).

The second core contribution is a novel approach for finding temporal correspondences of multiple related targets (*Where?*). This thesis proposes to represent the targets by a graph model, where each target is represented by a vertex and their relationships are encoded by edges. The traditional solution to find the temporal correspondences of a graph model is graph matching. In contrast to that, this thesis proposes a novel approach, which finds the correspondence of each vertex (target) by combining the appearance cue of a simple tracker with the structural cue deduced from a graph model. These two cues are combined in an iterative process inspired by the well-known Mean Shift algorithm. The outcome are correspondences for all vertices and edges in the graph, which locally maximize the similarity in appearance and locally minimize the deviation from the structure encoded in the model.

Finally, the main goal of this thesis is to show the potential of graph-based representations and methods in tracking. This goal has been achieved through these two core contributions.

Kurzfassung

Diese Dissertation präsentiert Forschung auf dem Gebiet des *Trackings* (Verfolgung). Tracking ist eines der am gründlichsten erforschten Themen im computerunterstützten Sehen (Computer Vision). Das Ziel beim Tracking ist es ein gewähltes Objekt (Ziel) in einem Video zu verfolgen. Diese Dissertation konzentriert sich auf ein spezielles Problem bei dem mehrere Ziele verfolgt werden sollen die in Beziehung zueinander stehen. Zwei wichtige Fragen beim Tracking sind: *Was ist das Ziel?* und *Wo ist das Ziel?* Die zwei wichtigsten wissenschaftlichen Beiträge dieser Dissertation beantworten diese Fragen mit Hilfe von Graphen.

Der erste Beitrag der Dissertation ist eine vollautomatische Initialisierung für Zielmodelle (*Was?*) basierend auf dem Prinzip: *Dinge die sich gemeinsam bewegen gehören zusammen.* Als Eingabe dient ein Video der sich bewegenden Ziele. In diesem Video werden interessante Punkte verfolgt und die Bewegungsinformation in Form von Trajektorien gespeichert. Basierend auf den Positionen der verfolgten Punkte im ersten Bild des Videos wird ein triangulierter Graph erstellt. Auf Grund der Bewegungsinformationen in den Trajektorien wird der Graph verformt. Die Verformung des Graphen wird zur Eingabe der folgenden, hierarchischen Gruppierung verwendet. Die Gruppierung wird durch eine unregelmäßige, duale Graphenpyramide umgesetzt. An der Spitze der Pyramide findet man die starren Komponenten des Videos (z.B. die Körperteile eines Menschen). Im letzten Schritt kann man durch Analyse der Bewegung feststellen, ob sich Komponenten durch Artikulationspunkte verknüpfen lassen (z.B. Ober- und Unterarm eines Menschen).

Der zweite Beitrag ist ein innovativer Ansatz, um zeitliche Übereinstimmungen für mehrere voneinander abhängige Ziele zu finden (*Wo?*). In dieser Dissertation wird vorgeschlagen das Ziel als Graph zu repräsentieren, wobei jedes Ziel als Knoten und ihre räumlichen Zusammenhänge als Kanten im Graphen gespeichert werden. Um eine zeitliche Übereinstimmung für einen Graphen zwischen zwei Bildern eines Videos herzustellen, wird üblicherweise nach dem ähnlichsten Graphen im zweiten Bild gesucht. Im Gegensatz dazu wird in dieser Dissertation ein innovativer Ansatz vorgestellt, der die Übereinstimmung für jeden Knoten (jedes Ziel) einzeln sucht. Dabei werden Informationen eines einfachen Trackingverfahrens, die vom Aussehen des Ziels abhängen, mit strukturellen Informationen aus dem Graphen kombiniert. In einem iterativen Prozess, der dem bekannten *Mean Shift* Algorithmus ähnlich ist, werden diese zwei Arten

von Information kombiniert. Das Ergebnis sind Übereinstimmungen für alle Knoten und Kanten im Graphen die lokal optimal bezüglich ihres Aussehens und ihrer Struktur sind.

Das Ziel dieser Arbeit war das Potential von Graphen im Tracking aufzuzeigen. Durch die zwei Beiträge dieser Dissertation konnte dieses Ziel erreicht werden.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Tracking aims to find the correspondence (e.g. location and pose) of an object of interest in every frame of a video sequence. It is one of the most thoroughly researched problems in computer vision. Even books [70] and reviews [100] are only able to cover a fraction of the field of tracking. The research in tracking is mostly application-driven and the developed approaches are tailored for them. Tracking is often an important processing step in frameworks and is used in many different applications. These applications can be grouped into six main areas [70]:

1. Media production [40, 103] and augmented reality [50, 18]
2. Medical applications [92, 97, 53] and biological research [82, 94]
3. Surveillance [106, 105] and business intelligence [83, 104]
4. Robotics [84, 77] and unmanned vehicles [80, 44]
5. Tele-collaboration [87, 41] and interactive gaming [38, 23]
6. Art installations and performances [74, 79, 42]

Tracking is an active field of research. Nevertheless, there are still challenging and open problems. Imagine a tracking approach as a detective trying to follow a suspicious person (object of interest). The detective needs to be careful not to mistake the suspect with a similar person (distractors). Furthermore, the suspect he is tracking might hide (occlusion), change clothes (changes in appearance) or even run away (fast, unexpected motion). These challenges and more have to be tackled by a successful tracking approach (see Section 2.3 for difficulties in tracking).

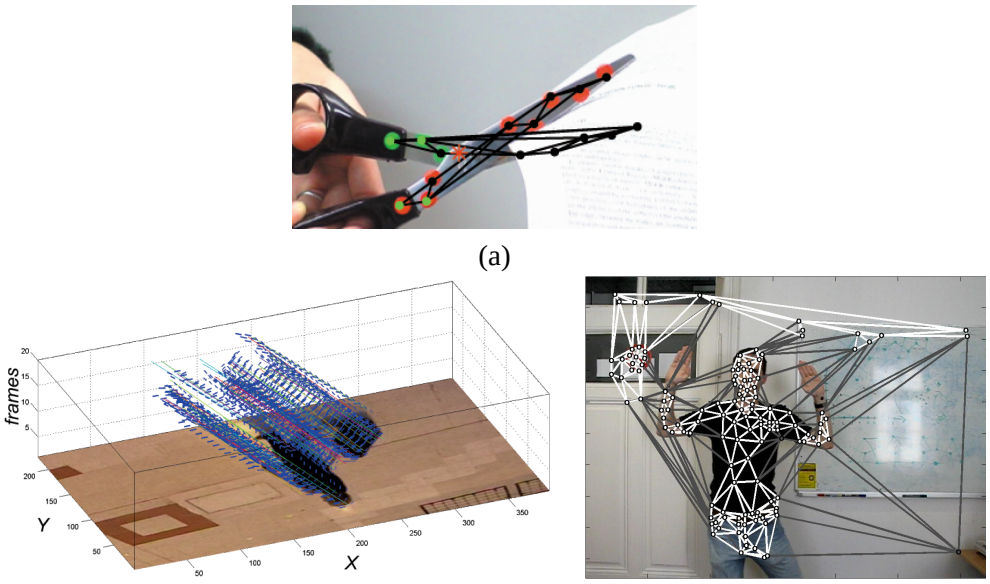
The success of the chase of the detective depends on his tracking skills (method for finding temporal correspondences) and his knowledge and understanding of the person of interest (target model). In cases where his suspect hides (occlusion), the detective is able to estimate the location based on his previous observations and his knowledge (target model). He might also be able to tell, where the suspect will reappear due to his understanding of the suspect (target model). If the suspect disguises itself, the detective may be able to recognize it by other information he knows (target model), e.g. body composition (target structure) and style of walking. Hence, this thesis focuses on how to describe the object of interest (initialization of target models) and how to follow it over time (finding temporal correspondences).

The research presented in this thesis is based on the work of the Pattern Recognition and Image Processing (PRIP) group at the Vienna University of Technology, especially on their work in the TWIST project (see Section 1.2). Their expertise are graph-based representations and methods. Graph-based representations (for details see Section2.4.1.3) are an elegant way to model an object of interest (target model) based on different kinds of information (e.g. appearance, geometry and structure). Coming back to the example with the detective, graph-based representations allow to describe the suspect based on its clothing, behavior, style of walking and composition of body by storing these information and their inter-dependencies. Graph-based methods employ such descriptions to solve tasks like tracking. Thus, the main goal of this thesis is to study the potential of graph-based representations and methods in tracking related multiple targets.

1.2 TWIST-CV Project

This thesis is based on the research done within the TWIST-CV project [1]. “Tracking with Structure in Computer Vision” (TWIST-CV) was funded by the Austrian Science Fund – Fond zur Foerderung der wissenschaftlichen Forschung (FWF) – under the grant FWF-P18716-N13 and was carried out between March 2006 and December 2009. The main goal of this project was to solve open problems in computer vision with the help of graph-based methods. There were three sub-goals:

1. Finding object correspondences in image sequences.
2. Finding object correspondences in images from different view points.



(a) (b) (c)

Figure 1.1: (a) Tracking through partial occlusion with a graph-based representation [10]. (b) Compositional representation in space and time [16]. (c) Initialization of graph-based representation based on the observation of motion [9].

3. Finding object correspondences in image sequences from different view points.

My contributions to this project are focused on the first subgoal. The main findings were a graph-based representation [15, 7, 10], a compositional representation [16, 6] and an approach to automatically initialize graph-based representations [9, 72]. Figure 1.1 shows images from the related publications.

1.3 Outline

This thesis is a cumulative doctoral thesis and is divided into three parts (in addition to this brief introductory chapter).

1.3.1 Outline Part I: Basic Methodologies and Concepts in Tracking

The aim of this part is to provide the necessary knowledge to understand the selected publications in Part II. Chapter 2 covers an introduction to tracking, which is relevant for all publications in Part II. Chapter 3 presents basic concepts and methodologies of the initialization of target models for Paper A and B. Chapter 4 introduces methodologies and concepts relevant for the proposed

Table 1.1: Publications associated with the two main topics of this thesis and ordered by their date of publication (decreasing, starting with most recent). Read from bottom to top.

Initialization of target models		**Finding temporal correspondences**	
Selected publications (see Part II)			
Paper B [13]	Determination of points of articulation for scenes with motion in the image plane.	Paper D [14]	Introduction of a novel structural cue. Simplification of the proposed tracking algorithm in [12]. Comparison of the novel structural cue against the previously used one.
Paper A [11]	Extension of [9] to motion out of the image plane.	Paper C [12]	Refinement of approach presented in [8]. Additional experiments.
Previous publications			
[9]	Refinement of the ideas proposed in [72]. This approach improves results for articulated targets and accuracy in general.	[8]	Adding a coarse-to-fine methodology to the tracking algorithm of [10].
[72]	First concepts for the initialization of target models of rigid and articulated targets based on their motion in the image plane.	[10]	Extension of [15] to articulated targets. Additional experiments.
		[7]	Extension of [15] to articulated targets.
		[15]	Introduction of a novel tracking algorithm combining appearance and structural information for rigid targets.
		[71]	Motivation and first ideas for finding temporal correspondences using structural information.

tracking approach in Paper C and D. This part concludes by listing the original contributions of this thesis and possible future work (see Chapter 5).

1.3.2 Outline Part II: Selected Publications

Part II is a selection of four papers from my publications (see Appendix A). These four papers were selected as they cover the most important or recent research on the two main topics of this thesis: **initialization of target models** (see Paper A and B) and **finding temporal correspondences** (see Paper C and D). All four papers passed through international peer-reviewing and are published as articles in international journals (Paper A and C) or as chapters in books (Paper B

and D). Table 1.1 lists all publications related to the two main topics and shortly explains their dependencies.

1.3.2.1 Contributions of the co-authors

This section describes the contributions of Walter G. Kropatsch, Adrian Ion and myself to the selected papers of this cumulative thesis.

My contributions: As the first author, I was the main source of the ideas and the developed methods presented in all selected papers. For Papers A and B, I adapted the existing implementation of the PRIP group of the irregular dual graph pyramid for image segmentation to motion segmentation. The novel concepts (observing and measuring similarity in motion, and identification of points of articulation) were implemented from scratch. The implementation of the methods proposed in Paper C and D, was solely done by myself (except for the explicitly cited third party code in Paper C). I created the synthetic and real life videos (except for the explicitly cited third party videos), gathered ground truth data and conducted the experimental evaluations and their analysis for all papers. The text and figures of the four Papers were mainly written and created by me, except for short paragraphs resulting from the feedback of Walter G. Kropatsch and Adrian Ion.

Contributions of Walter G. Kropatsch: As my mentor (Doktorvater) and head of the Pattern Recognition and Image Processing Group (PRIP), he was involved in all my research activities, including the selected papers of this thesis. Irregular graph pyramids are the long term research topic of Walter G. Kropatsch. Therefore, he supported me with his in-depth knowledge during the development of the ideas leading to these publications and also in solving difficult problems. Furthermore, he provided feedback for the written text of all papers and gave ideas for possible future work.

Contributions of Adrian Ion: Adrian Ion worked at PRIP as a project assistant and his research partly overlaped with the topics of this thesis. Together with Walter G. Kropatsch and Yll Haxhimusa, he did research on cognitive vision [55, 54] and image segmentation [48, 62], which is related to the research presented in this thesis. For Paper A and B, he provided the implementation of the grouping framework based on irregular dual graph pyramids for image segmentation and he supported me during the adaptation of the code to the problem of motion segmentation. Furthermore, he assisted in recording the videos for the experiments. For Paper A, B and C, Adrian Ion contributed to the text by giving feedback, especially for the paragraphs about the theoretical background of irregular dual graph pyramids.

1.3.3 Outline Part III: Appendix

This appendix includes my curriculum vitae and a complete list of my publications.

Part I

Basic Methodologies and Concepts in Tracking

CHAPTER 2

About Tracking

The aim of this chapter is to provide an introduction to tracking and the relevant terminology for the following chapters and the selected publications in Part II.

2.1 What is Tracking?

In the book of Maggio et al. tracking is defined as follows [70]:

> "One fundamental feature essential for machines to see, understand and react to the environment is their capability to detect and track objects of interest. The process of estimating over time the location of one or more objects using a camera is referred to as video tracking."

Maggio et al. use the term *video tracking* in their book to emphasize that the input is a video. A video is an ordered sequence of images, which are often called frames. In this thesis, I simply use the term *tracking*. Figure 2.1 visualizes the concept of a simple tracker over three frames.

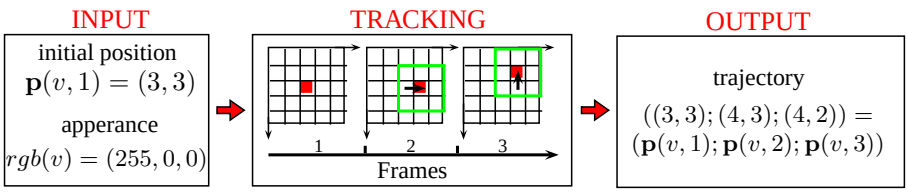


Figure 2.1: Example of a simple tracker over three frames. Input: position $\mathbf{p}$ of target v in the first frame and appearance described by a color-vector (red). Tracking: starting from the position in frame 1 the tracking algorithm searches for the new position within a search window (green rectangle). Output: 2D trajectory consisting of a sequence of 2D positions.

Table 2.1: Targets in tracking.

Term	Dimensionality	Homogeneity
target point	0-dimensions	homogenous
target line	1-dimension	homogenous
target patch	2-dimensions	homogenous
target volume	3-dimensions	homogenous
target structure	arbitrary	non-homogenous

2.2 What is a Target?

The object of interest in tracking is often called *target object* or short *target*. This thesis additionally distinguishes between targets depending on their dimensionality and homogeneity (see Table 2.1). A target is homogenous if its elements are of the same kind, e.g. a target patch is made up of pixels. Target structures consist of related multiple non-homogenous elements and can be of arbitrary dimension. They can be composed of several targets. For example, a human can be tracked as a target structure, which consists of target patches (body parts) connected through target points (points of articulation). In addition, target structures can be hierarchies built of several levels of different kinds of targets. Targets should not be confused with target representations, which are used to describe them (e.g. a target patch can be described by a histogram and a target structure by a graph; see Section 2.4.1).

2.3 Levels of Difficulty in Tracking

A *tracking task* is a concrete problem with a certain input and an expected output. The difficulty of a tracking task depends on several factors. Figure 2.2 gives an overview of these factors and their properties. In the following sections the factors are explained in detail.

2.3.1 Input

The *input* of a tracking task is the data on which the tracker operates. This data is not limited to videos, but includes user interaction and knowledge. Therefore, the factor input includes properties of the acquisition of the video and properties independent of the video.

The quality of the sensor, which is used to capture a video, has a direct influence on the *video quality*. Video quality includes resolution of the target, but also grade of image noise, which is introduced into the video during the acquisition. In general, the higher the video quality the higher is the resolution of the target (more data) and the lower the grade of noise. High-quality videos allow accurate tracking and low-quality videos fast tracking (less data to process).

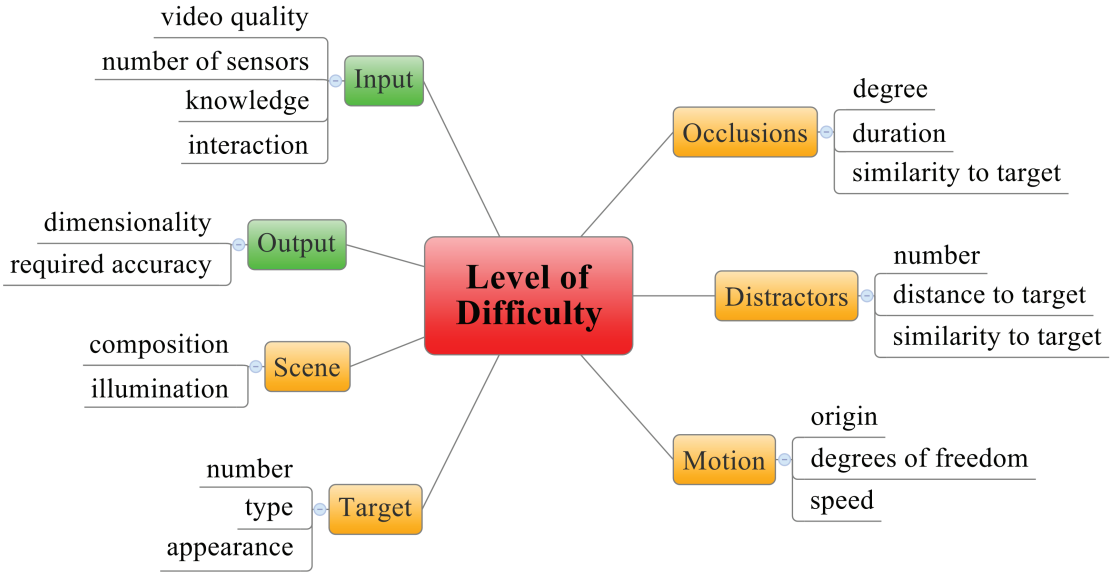


Figure 2.2: The level of difficulty of a tracking task depends on several factors (yellow and green boxes) and their properties. Input and output differ from the other factors as they do not concern the video, but its acquisition and issues independent of the video (e.g. interaction).

The video input of a tracking task can originate from different *numbers of sensors*. Generally, the computational complexity of a tracking approach increases with the number of sensors. The input of multiple sensors needs to be related (registered) against each other as each sensor typically captures the scene from a different viewing angle [47]. Having two or more sensors does not automatically increase the level of difficulty. For example, estimating the 3D position of a target from the input of a single sensor [65] is more challenging than from multiple sensors.

Prior *knowledge* is another form of input, which is available to the tracker and aims to improve its performance. It can be information about the scene (see Section 2.3.3), the target (see Section 2.3.4), the motion (see Section 2.3.5), the distractors (see Section 2.3.6) and the occlusions (see Section 2.3.7). The higher the amount of knowledge the more information is available to the tracker to find correspondences of the target over time. For example, a background-model [30], which allows the tracker to separate the target from the background, can substantially reduce the difficulty of a tracking task. However, if the background-model does not describe the background well (e.g. the model assumes a static background, but a tree is moving in the background) it has a negative influence on the tracking and may lead to worse results than tracking without a background-model.

Tracking can be done manually, interactive or fully automated. In manual tracking the user determines the temporal correspondences of the target at each frame (e.g. by selecting the center

position of the target). Such manual tracking is very time consuming and is only employed in cases where automated procedures fail to deliver the necessary accuracy (e.g. tracking landmarks on faces of facial palsy patients [37]). Interactive tracking comes with different degrees of interaction. Some tracking approaches use a manual initialization of the target by the user in the first frame of the video. Other trackers ask for several manual corrections during tracking (e.g. when the target is lost). Fully automated tracking does not employ any user interaction. In general, the higher the *degree of interaction* the easier the tracking (under the assumption that the input of the user is correct).

2.3.2 Output

Each tracking approach delivers an *output*, where the properties of this output depend on the task.

The higher the *dimensionality* of the expected output the higher the level of difficulty. A simple output is a binary one, where 1 indicates that something moved between two frames and 0 stands for no motion (i.e. two consecutive frames are equal). For most tracking tasks such a binary output is not sufficient and the output consists of at least one or more trajectories (i.e. a sequence of target positions) in 2D or even 3D. Besides trajectories, some tasks require the output of the orientation, area, shape or even pose of a target.

A tracking task typically requires a certain *accuracy* of the output. High accuracy is usually related to a higher level of difficulty and/or computational complexity. Accuracy can be quantified by the deviation from the ground truth. For example, if the expected output is the 3D pose of a rigid target, the deviation from ground truth can be calculated by comparing the values of the six degrees of freedom of the result (rotation around every axis and translation in all three dimensions) against the values in the ground truth data. Depending on the task, it is necessary to deliver results with high accuracy (e.g. medical applications [37]) or low accuracy (e.g. real time tracking for gaming).

2.3.3 Scene

A *scene* can be defined as a fraction of reality captured by the sensor of a camera in the form of a discrete grid of pixels.

The *composition* of a scene is relevant for the tracking task. It divides the scene into foreground (target) and background (everything except the target). Scenes where the target occupies a significant part of the space (e.g. a target of 300 $\times$ 300 pixels in a video with frames of 800 $\times$ 600 pixels) are generally easier to track than scenes where the target is only a marginal part. Concerning the background, the level of difficulty depends on the amount of clutter (low: easy;

high: difficult). A uniformly colored background has no clutter (e.g. white wall), whereas a highly textured background has a high amount of clutter (e.g. library).

The *illumination* in a scene influences tracking as most features (e.g. color and edges) are not invariant to changes in illumination. Constant illumination, which can be found in controlled, indoor environments, reduces the difficulty of a tracking task, whereas non-constant illumination increases the difficulty.

2.3.4 Target

In tracking, a *target* is the object of interest which is tracked over time (see Section2.2).

The difficulty of a tracking task depends on the *number* of targets. Tracking a single target is computationally less expensive than tracking multiple targets. It is difficult to deliver accurate tracking results (close to ground truth) for multiple targets in real time, while keeping the complexity of the algorithm low. Furthermore, it is easier to track an object of interest as a single target patch (e.g. human) [26] in comparison to tracking related multiple target patches (e.g. a target patch for each body part) [12]. Please note, that tracking multiple targets can also be an advantage. It can be easier to track several targets than to track only one target. For example, in a scene with pedestrians it can be difficult to avoid mixing them up, but if all pedestrians are separately tracked this problem becomes easier.

The *type* of a target is a relevant issue. One can distinguish between rigid (e.g. mobile phone), non-rigid (e.g. face) and articulated targets (e.g. robot arm). Articulated targets are built of several parts connected via points of articulation [11]. The most challenging target to track is probably an articulated target consisting of non-rigid parts (e.g. human). Furthermore, the spatial distribution of a target is of importance. The pixels of compact targets are more or less evenly distributed in space (2D: x- and y-axis; 3D: x-, y- and z-axis). Examples for compact targets are face, ball and car. In non-compact targets, the pixels are unevenly distributed in space (e.g. airplane, nail and train).

2.3.5 Motion

The *motion* in a video sequence can *originate* from the motion of the camera or from the motion of entities in the scene (e.g. humans, cars, animals and trees moving in the wind). Videos where the camera and entities in the scene move, are more difficult to track than videos with only one source of motion.

The difficulty of motion can be quantified based on the *degrees of freedom* (DOF). A train moving along a track has one DOF, where the position of the train is determined by the traveled (moved) distance along the track. Rigid targets moving in 2D space have up to three DOF

(translation along x-axis, translation along y-axis, rotation in the 2D plane). In 3D space, a rigid target may move with up to six DOF (rotation around every axis and translation in all three dimensions). For a non-rigid target, the quantification of the DOF of its motion is more complex. One possibility is to split the non-rigid target into patches and define the DOF of each patch. Another possibility is to approximate the motion of the non-rigid target by the DOF of a rigid target. This solution is suitable for cases, where the focus lies on the global motion of the target rather than small local movements.

The third property of the factor motion is *speed*. In tracking, speed is often measured by how many pixels a target moves between two frames. For a target point it is easy to determine how many pixels it moved, but for a target patch or a target structure it is more complicated. If a target patch rotates around its center the speed of its pixels differs. Pixels closer to the center move slower than pixels farther away. A solution is to pick the maximum pixel distance of a target as representative speed for the whole target. To categorize a certain speed as fast or slow, it is necessary to consider the size (in pixels) and type (shape) of the target. For example, the speed of a target point moving with 20 pixels per frame is considered fast, while the speed of a target patch of 200×200 pixels moving with 20 pixels per frame is considered moderate or even slow. If the target is non-compact (e.g. pen), the speed of motion along its major axis needs to be treated differently than along its minor axis. Fast motion (high speed) is more difficult to track than slow motion (low speed). In addition, it is important to consider if the speed is constant or non-constant. A target with non-constant speed is more difficult to track, because estimating the position of the target cannot be solved by a simple linear motion model.

2.3.6 Distractors

A *distractor* is an entity in the scene of a video, which may distract the tracker from the target leading to inaccuracy (deviation from the ground truth) or even tracking failure (target is lost).

The higher the *number* of distractors and the smaller their *distance* to the target, the higher is the possibility that the tracker is influenced and the higher the level of difficulty. The distance is measured in pixels and depends on the size and shape of the target and the distractor itself. For example, if both, the target and the distractor, are compact and have a diameter of ten pixels, a distance (measured from the centers) of 20 pixels or less is close. Under different circumstances with a target and a distractor of smaller size (e.g. diameter of five pixels), a distance of 20 pixels is less problematic.

Besides the distance, the *similarity* of a distractor to the target influences the level of difficulty. Similarity can be measured based on different criteria (e.g. appearance, geometry or motion). Dealing with distractors becomes more difficult the more similar the distractors are to

the tracked target.

2.3.7 Occlusions

During an *occlusion* the target or parts of the target become invisible. An occluder can be a static entity in the scene (e.g. traffic sign), a moving entity (e.g. car) or a part of the target (e.g. a leg of a human occluding the other leg while walking, a so-called self-occlusion). If a target leaves the scene, this can also be called an occlusion.

An important property of an occlusion is its *degree*. The degree quantifies how much of the target is occluded. This can be measured in percent for target patches or in number of vertices for target structures like graphs. In general, the higher the degree of occlusion the higher the level of difficulty. Full occlusions where the whole target becomes invisible are the most difficult cases. In such situations trackers rely on prior knowledge (e.g. motion during previous frames) to estimate the motion and behavior of the target during the occlusion.

Besides the degree, it is also relevant to consider the *duration* of an occlusion. The longer the occlusion the higher the possibility for tracking errors and the higher the difficulty. Usually a tracker estimates the state of an occluded target based on the state of the target before the occlusion. Due to changes in the motion of the target, tracking errors accumulate over time. Thus, the longer the occlusion the higher the accumulated error.

As for distractors (see Section 2.3.6), occlusions are especially problematic, if they are *similar to the target*.

2.3.8 Rating Levels of Difficulty

Based on the factors described above (see Figure 2.2), the level of difficulty of a tracking task can be rated. To the best of my knowledge, there is no standard-approach for the rating of a tracking task. On the contrary, this is an open issue in the field of tracking. Rating the difficulty of a tracking task is relevant, as it allows to better understand the strengths and weaknesses of a certain approach and it helps to judge the quality of the delivered results. Setting up a rating system is a challenging and complex task, because one has to consider the factors with their properties, their temporal dynamics (e.g. the degree of an occlusion is changing over time due to the motion of the occluder) and their inter-dependencies (e.g. the type of the target limits the DOF of the motion).

In Table 2.2 a simple rating system is proposed, which is used in the discussions in Chapter3 and 4. The *speed* of a target v can be calculated as follows:

$$\text{speed} = \frac{||\mathbf{p}(v,t) - \mathbf{p}(v,t-1)||_2}{\text{size}(v)}, \tag{2.1}$$

Table 2.2: Rating the level of difficulty of tracking tasks. This rating results in a vector of scalars, where each element represents the difficulty ($0 =$ not occurring, $1 =$ low, $2 =$ medium, $3 =$ high) of the related factor. For example $(1, 1, 2, 3, 2, 2, 1, 0)$, where the ordering of the factors is as listed in this table.

Factor	**Low difficulty (1)**	**Medium difficulty (2)**	**High difficulty (3)**
Input	number of targets known and initialization manually	number of targets unknown and initialization manually OR number of targets known and no manual initialization	number of targets unknown and no manual initialization
Output	single trajectory	multiple trajectories	multiple trajectories in a hierarchy
Scene	background not cluttered	background cluttered	background highly cluttered
Target	rigid	articulated with rigid parts	articulated with non-rigid parts
Motion	DOF ≤ 3 and speed < 1.0	$3 <$ DOF < 6 or speed ≥ 1.0	DOF ≥ 6 or speed ≥ 5.0
Distractors	similarity < 0.5	similarity ≥ 0.5	similarity ≥ 0.7
Occlusions	degree < 0.5, duration < 0.5 and similarity < 0.5	degree ≥ 0.5, duration ≥ 0.5 and similarity ≥ 0.5	degree ≥ 0.7, duration ≥ 0.7 and similarity ≥ 0.7

where *size* is the length of the major axis of target v and $\mathbf{p}(v, t)$ is the position of target v in frame t (consequently the size of a target point is 1). As this thesis is about tracking related multiple targets, the speed of the fastest target is chosen as representative for all other targets. The *similarity* of a distractor or an occluder towards a target is determined based on their appearance (i.e. a distance measure is employed to calculated the similarity). If several distractors or occluders appear, the value of the most similar one is selected. Furthermore, I only consider distractors which have an influence on the tracker (which are close to the target(s)). The *degree of occlusion* is measured by how many targets are occluded. For example, if ten target patches are tracked and five of them are occluded, the degree of occlusion is 0.5. The *duration of occlusion* is the percentage of frames of a video, where not all targets are visible (independent of the degree of occlusion).

2.4 Components of a Tracker

A tracker consists of several interdependent components. Maggio et al. identify five components [70]: (i) feature extraction, (ii) target representation, (iii) finding correspondences, (iv) track management and (v) meta-data extraction. Besides these five components, tracking frameworks often include pre- and post-processing steps. Possible pre-processing steps are denoising [3] and segmentation [33, 61]. During post-processing a common step is to smooth trajecto-

ries [88].

This thesis contributes to two of the five components: target representations (see Paper A and B) and finding correspondences (see Paper C and D). Hence, these two components are introduced in more detail in Sections 2.4.1 and 2.4.2.

The aim of *feature extraction* is to exploit discriminative information from the input. Features are used to describe the target and to distinguish the target from the background of the scene. Depending on the input data, certain features are suitable for this purpose. There are three levels of feature extraction, where higher levels generally come with higher computational extraction costs [70]: Low-level (color, gradient, motion), Mid-level (edges, corners, regions), and High-level (background, objects).

Track management deals with the organization of trajectories. This component is especially important for applications with multiple targets, where there is no user interaction (e.g. manual initialization in the first frame). In such cases, targets need to be detected automatically and a new trajectory is initialized through track management – this is called the *birth* of a target. The *death* of a target occurs, if it leaves the scene or if it is occluded. It is particularly difficult to re-identify targets, which have been occluded and become visible again. Mostly, these targets are identified as new targets and a new trajectory is created instead of continuing the existing trajectory.

The *meta-data extraction* is a component of the post-processing stage of a tracker. After acquiring the temporal correspondences, the task of the meta-data extraction is to exploit additional data from these correspondences. Depending on the application, this meta-data could be 3D information used for 3D reconstruction, navigation commands for a robot or recognized gestures, which are used to interact with a game.

2.4.1 Target Representations

This section gives an structured overview about well-known, basic target representations. In practice, a target representation consists of a combination of these representations. A target representation aims to describe the target in a discriminative way as to distinguish it from the background and other entities in the scene. In general, a suitable representation is chosen based on the application, the target and the expected output. The actual description of a target is called *target model* in this thesis.

Representations can be distinguished based on the encoded information. The following three categories are distinguished in this thesis: (i) *appearance representations*, (ii) *geometric representations*, and (iii) *graph-based representations*. This section does not cover temporal representations (i.e. motion models), because they are not relevant for this thesis.

Every representation described in the following sections has parameters, which can change over time. For example, if the target is described by a 2D point, its parameters are the x- and y-coordinates in the image. These coordinates change, if the target or the camera moves. Updating the parameters of a representation can be a challenging problem, which is why it is an open problem [73, 31].

2.4.1.1 Appearance Representations

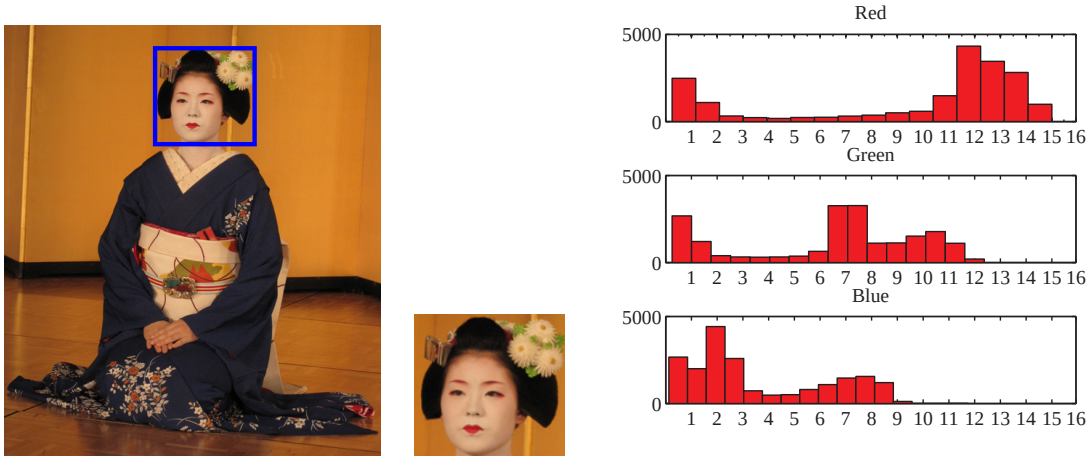
The *appearance* of a target is how it looks from a particular view-point and under certain illumination conditions [35].

Templates are a common appearance representation and date back to 1981 [67]. They describe the target by its pixel information consisting of color or gray values and the corresponding coordinates. A template can be initialized by the user through a manual selection of a region of interest including the target or by automatic detection. Furthermore, a template can be built from one or several images in a training set. Tracking algorithms employing a template representations often assume that the appearance of the target remains more or less the same throughout the video. However, this assumption is only valid for a limited field of applications.

Histograms are another frequently used appearance representation. In comparison to templates, histograms generally do not encode position information and the appearance information is usually quantized into a certain number of bins. Histograms represent a target by the statistical distribution of certain appearance information within a region of interest and are invariant to 2D transformations up to a certain degree. Color histograms are built from the color values of the pixels representing the target and are for example employed by the popular Mean Shift tracking algorithm [26, 63]. Maggio et al. [69] propose a target representation based on multiple color histograms, which are computed from a region of interest in a semi-overlapping manner to additionally integrate spatial information. Dalal et al. introduce *histograms of oriented gradients* in [28], where the idea is to describe a target by local histograms of image gradient orientations. This is realized by dividing the region of interest containing the target into “cells” (small spatial patches) and by extracting a local 1D histogram of gradient directions or edge orientations from each cell. The combined histogram entries form the representation. Figure 2.3 shows examples for a template and a histogram representation.

2.4.1.2 Geometric Representations

The simplest representation for a target is through a single *point*, which describes a certain 2D or 3D position. This kind of representation has been thoroughly researched by the radar



(a) (b) (c)

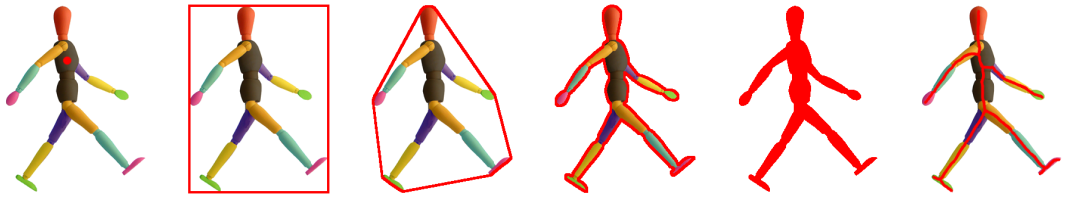
Figure 2.3: (a) Image with selected target patch (region inside blue rectangle: face); (b) Template of target patch; (c) RGB-histograms of target patch with 16 bins for each color channel.

community [70]. A set of points is commonly used to represent and track corner points [89, 57] for applications like structure from motion [5] and video object segmentation [36, 11].

Primitive shapes describe a target with a generalizing, primitive, 2D shape (e.g. bounding rectangle or ellipse) or a 3D volume (e.g. bounding cylinder or sphere). In comparison to the point representation, primitive shapes provide additional information about the size of a target. Both representations are not able to give detailed information about the spatial decomposition of a target. Hence, they are suitable for compact, rigid targets (e.g. man-made objects like cars). A popular application of this representation is tracking with Mean shift [26, 63]. Bradski [22] additionally extracts the orientation of the target.

Representations giving more information about the spatial decomposition of the target are *convex hull*, *silhouette* and *contour*. An intuitive explanation for the convex hull is to imagine an elastic band stretched upon a given target to encompass it [102]. Convex hulls are used for human action recognition from videos [68, 24]. Silhouette and contour are related: a contour is a sequence of coherent pixels separating a target from its background – a boundary, whereas a silhouette is the region inside a contour. They are both suitable for tracking complex non-rigid shapes [99] or for action recognition [96, 43]. A strength of contours and silhouettes is that they are robust to changing illumination conditions. Nevertheless, initializing and updating the representation is challenging with regard to the problem of separating the target from the background.

A representation especially popular in applications with articulated targets (e.g. humans) are



(a) (b) (c) (d) (e) (f)

Figure 2.4: (a) Point; (b) Primitive shape (rectangle); (c) Convex hull; (d) Contour; (e) Silhouette; (f) Skeleton; The lines in these drawings are exaggerated for better visibility (e.g. the point representation is not a point but a small circle).

skeletons. They can be extracted by using the medial axis transformation [19] or the distance transformation [56]. An unsolved problem with skeletons lies in their extraction as it is not robust against noise. Applications for skeletons are for example motion capture [40, 103] and human activity detection [81]. Figure 2.4 shows the presented geometric representations.

2.4.1.3 Graph-based Representations

Graph-based representations are used in various fields (mathematics, computer science, etc.) and have a long history of research [46, 64]. They can be employed to represent different kinds of information (e.g. appearance, shape and spatial, hierarchical and temporal relationships).

The baseline representation is a *graph* $\mathbf{G}$ consisting of a set of vertices $\mathbf{V}$, which are connected via a set of edges $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$.

The simplest way to represent a digital 2D image is by a *4- or 8-neighborhood graph*, where each pixel is represented by a vertex and the edge structure describes the local neighborhood of each vertex. Such a representation consists of the same number of vertices as there are pixels in the image. In general, representations used for tracking try to limit the stored information to relevant, discriminative information, which reduces processing time and increases the performance of a tracker.

A graph may be *weighted*, where weights w are stored with the edges and/or vertices. These weights are normally real numbers. Weighted graphs are *attributed graphs* (AG). In this thesis, the term attributed graph is used for graphs, where the vertices store features (e.g. discriminative features extracted from the region covering a target patch) and not only real numbers. The edges of an attributed graph encode the spatial relationships of the features (attributes). Typically, attributes are stored in the vertices, but there are attributed graphs which store attributes with other entities as well (e.g. edges and faces) [14]. The edges in a graph can be *directed* resulting in a *directed graph*. In a directed graph, the weight $w_{ij} \neq w_{ji}$, where v_i and v_j are the vertices, and e_{ij} and e_{ji} are the two possible, oriented edges [64]. A graph is called directed graph as soon

as one edge is directed. In contrast, an *undirected graph* with weights requires that $w_{ij} = w_{ji}$ for all edges. Please note, that there are undirected graphs without weights.

A *region adjacency graph* (RAG) [93] represents adjacency between image regions. Typically, first an image segmentation algorithm is applied to group the pixels of an image into regions based on some homogeneity criterion (e.g. color). Then, the RAG represents each region with a vertex and connects vertices with edges, where the corresponding regions are adjacent to each other.

Graph-based representations are also used to describe hierarchies. A simple hierarchy is a *regular pyramid*, which consists of at least two levels. Each level can be represented by a graph, but it is more common to represent them by arrays (more efficient). Regular pyramids represent parent-child-relationships in-between the successive levels with edges. In a regular pyramid, the input image is the base level. The successive levels reduce the data by a constant reduction factor $\lambda > 1.0$ and with reduction windows of constant size. Regular pyramids are an efficient representation as their vertical structure is fixed, but they are not invariant to translation, rotation and scaling and do not preserve connectivity [61]. *Irregular pyramids* try to overcome the drawbacks of regular pyramids. They are shift-invariant and adapt to the image data. Hence, the structure of irregular pyramids is not fixed and their building process depends on the data [61]. The levels of an irregular pyramid are frequently represented by graphs and they can be built using dual graph contraction [85, 60] or graph decimation [75].

2.4.2 Finding Correspondences

The objective of this component is to find temporal correspondences of a target based on the inputs (video, knowledge and interaction) and the target model. A tracking approach finds the correspondence between the status of a target in frame t and $t + 1$, where the correspondence is a relative, n-dimensional vector. Its dimensionality depends on the elements of the target model. The components of the correspondence vector are divided into similarity components and change components. Similarity components are used by the tracking approach to establish the correspondence.

For example, the model of a target point consists of its 2D position (x, y) and its color in RGB (r, g, b). This results in a 5D correspondence vector (x, y, r, g, b), where the RGB values are the similarity components. Imagine the position of the target point at frame t is $(10, 10)$ and its color is $(1, 0, 0)$. At frame $t+1$, the employed tracking approach finds the correspondence of the target with the help of the similarity component, because at position $(5, 15)$ the RGB values are equal to the model (i.e. similarity is 100 %). The resulting correspondence is the relative vector $(-5, +5, 0, 0, 0)$.

2.4.2.1 Single-hypothesis versus Multi-hypothesis Methods

One can distinguish methods for finding temporal correspondences based on the number of possible candidates [70]. *Single-hypothesis methods* find at each frame one candidate for the temporal correspondences of a target. Well-known trackers employing a single-hypothesis methodology are the Kanade-Lucas-Tomasi [89, 67] and Mean Shift [26]. In contrast to that, *multi-hypothesis methods* work with multiple candidates. A popular multi-hypothesis method is the particle filter [17]. Single-hypothesis methods are computationally less expensive than multi-hypothesis methods. However, single-hypothesis methods are in general more sensitive to occlusions and have problems in dealing with distractors.

2.4.2.2 Processing Strategies

Traditional tracking approaches find temporal correspondences in a sequential frame-to-frame manner: at each frame, the correspondence of a target is found based on the status of the target in the previous frame and its similarity components. This frame-to-frame processing is not always the best choice as correspondences based on the information of two consecutive frames can be prone to noise and errors. In [59], a forward-backward-processing is proposed which allows to reliably detect tracking failure and select valid correspondences. Such a processing strategy is also helpful if the target is occluded [95]. Furthermore, if the temporal resolution is high (high-speed cameras) or the motion of the target is known, it is not necessary to process every frame to achieve reliable correspondences [52]. There are tracking approaches using space-time processing, where several frames (space-time volumes) are processed at once and the correspondences are estimated by fitting a motion model to the space-time data [16].

2.4.2.3 Local versus Global Optimization

The appearance and shape of a target usually changes over time due to image noise, motion or occlusions. Hence, the correspondence at a certain frame is not equal to the target description in the model, but similar. Therefore, finding temporal correspondences can be seen as an optimization problem. In tracking, one can distinguish between local and global optimization. Local optimization searches for correspondences in a local neighborhood (often called search window) which is mostly around the position of the target in the previous frame [26]. Global optimization looks for the best correspondence in the whole image. Thus, local optimization is in general computationally less expensive. Apart from computational issues, local optimization is mostly inferior to global optimization. Fast or unexpected motion is problematic for local optimization as the search window might be too small. If the target disappears due to occlusions, it is more difficult to search for its reappearance within a local neighborhood instead of the whole image.

Setting the size of the search window itself is challenging, especially if the target undergoes scale changes.

There is another interpretation of local and global optimization in the case of related multiple targets. Lets assume the object of interest is a target structure consisting of vertices and edges. Local optimization finds correspondences by considering local neighborhoods of the target structure (e.g. a vertex, its incident edges and direct neighbors) [12, 14], whereas global optimization finds correspondences by taking into account the whole target structure [27].

This is the end of this chapter about basic knowledge in tracking. The introduced terms, concepts and methods are relevant for the following Chapters 3 and 4, and the selected publications (Paper A, B, C, and D).

CHAPTER 3

Initialization of Target Models

This chapter is about the initialization of target models. Section3.1 is about the motivation and the concept of the proposed approach. Section3.2 is a recall on irregular dual graph pyramids, which are the basis of the presented approach. Section3.3 summarizes the selected publications Paper A and B and discusses the most important results.

3.1 Motivation and Concept

The representation of the target object is an important component of a tracker (see Section2.4). Desirable properties of a target model are:

discriminative: to distinguish the target from the background and similar entities in the scene (distractors);

invariant: to different kinds of transformations due to the motion and pose changes of the target;

efficient: to avoid storing redundant and unnecessary information and to allow an easy extraction of the relevant information;

The quality of a target model depends on its initialization. There are tracking approaches where the user manually initializes the model. The advantage of such an initialization is that humans are in general able to easily initialize the desired target even if the scene is difficult (e.g. distractors). Disadvantages are the effort of doing the initialization manually, the subjectivity of the user (different users may initialize the same target differently) and the required knowledge of the user (some applications require expert-knowledge).

There are approaches which initialize the target model automatically. One possibility is to solve the initialization as a recognition task [34, 32]. The success of such approaches highly depends on the training methodology and the training set. If the target is similar to the training examples, such approaches will perform nicely and deliver a reliable initialization. In cases where the variability of the target is limited, such an initialization is a good choice. Nevertheless, there are applications, where the targets differ or unexpected targets occur. Furthermore, even though such approaches can be fully automatic, training sets are mostly at least partly labeled by users, which again brings in the previously mentioned disadvantages (effort, subjectivity and expert knowledge).

The solution I chose in this thesis is based on segmentation and does not require any prior knowledge or training. In computer vision, segmentation is often applied to single images to group their pixels into similar or even meaningful regions [33, 49, 93, 90, 21, 61]. In the case of image segmentation, these pixels are grouped based on certain criteria, which are often similarity in color and spatial proximity. Papers A and B present an approach based on irregular dual graph pyramids to extract a target model for rigid or articulated targets by applying segmentation in the temporal domain (video). The basic idea is: *"things that move together belong together"*. Hence, the proposed grouping criterion is the observed motion in the input video. This idea stems from the field of cognitive psychology. In 1973, Johansson Gunnar published a well-known work on *biological motion* [58]. He made the observation that humans are able to recognize a human figure based on a few bright spots undergoing motion along the major joints of a human body. The biomotion lab of Prof. Dr. Nikolaus Troje at Queen's University in Kingston, Ontario offers demos about biological motion on their website [2].

3.2 Irregular Dual Graph Pyramids

This section introduces irregular dual graph pyramids, which are the basis of the proposed approach for target model initialization in Papers A and B. Its aim is to shortly explain the most important concepts. For detailed information on irregular dual graph pyramids and the underlying graph theory the reader is referred to [61, 60].

Before explaining the building process of an irregular dual graph pyramid, it is necessary to introduce *planar graphs* and *dual graphs*. A graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is planar if it can be embedded into the plane in $\mathbb{R}^2$, where all vertices $\mathbf{V} \subset \mathbb{R}^2$, every edge $e \in \mathbf{E}$ is an arc between two vertices and no two edges cross each other. A planar graph $\mathbf{G}$ divides the plane into a set of faces $\mathbf{F}$. These faces and their adjacencies can be represented by the *dual graph* $\overline{\mathbf{G}} = (\overline{\mathbf{V}}, \overline{\mathbf{E}})$. Each $\overline{v} \in \overline{\mathbf{V}}$ represents a face $f \in \mathbf{F}$. Every pair of vertices in $\overline{\mathbf{G}}$ which are adjacent, i.e. the corresponding faces $f \in \mathbf{F}$ share a common edge $e \in \mathbf{E}$, are connected by an edge $\overline{e} \in \overline{\mathbf{E}}$, so

that edges e and $\overline{e}$ are crossed. There is a one-to-one correspondence between the vertices $\overline{\mathbf{V}}$ of $\overline{\mathbf{G}}$ and the faces $\mathbf{F}$ of $\mathbf{G}$, and between the edges $\overline{\mathbf{E}}$ of $\overline{\mathbf{G}}$ and the edges $\mathbf{E}$ of $\mathbf{G}$. Furthermore, the dual of $\overline{\mathbf{G}}$ is again $\mathbf{G}$ [61]. In the following, the graph $\mathbf{G}$ is called *primal graph* and $\overline{\mathbf{G}}$ is called *dual graph*.

Irregular dual graph pyramids are built in a bottom-up manner, where level $\mathbf{G_k}$ results from dually contracting the preceding level $\mathbf{G_{k-1}}$ (see Figure 3.1). Hence, such a pyramid is a stack of successively reduced planar graphs $P = \{(\mathbf{G_0}, \overline{\mathbf{G}}_\mathbf{0}), \ldots, (\mathbf{G_n}, \overline{\mathbf{G}}_\mathbf{n})\}$, where $\mathbf{G_0}$ is the bottom level, $\mathbf{G_n}$ is the top level and n is the *height* of the pyramid [61].

The *dual graph contraction* of each level $(\mathbf{G_k}, \overline{\mathbf{G}}_\mathbf{k}), 0 < k \leq n$ consists of two steps [61]:

1. Edge contraction in $\mathbf{G_{k-1}}$ if the corresponding vertices should be merged based on a certain similarity criterion, which is equivalent to edge removal in $\overline{\mathbf{G}}_\mathbf{k-1}$;

2. Edge removal in $\mathbf{G_{k-1}}$ to simplify the structure, which is equivalent to edge contraction in $\overline{\mathbf{G}}_\mathbf{k-1}$;

For the dual graph contraction, it is necessary to define so-called *contraction kernels*. A contraction kernel is a tree of depth one consisting of a subset of non-surviving edges and a subset of vertices, where one vertex is called *surviving vertex*. In Paper A and B, the non-surviving edges for the contraction kernels are selected by the Minimum Spanning Tree algorithm of Borůvka [78]. The vertices of a contraction kernel in level $k-1$ form the *reduction window* $W(v)$ of the respective surviving vertex v in level $k-1$. The *receptive field* $F(v)$ of v is the (connected) set of vertices from level 0 that have been "merged" to v over levels $0 \ldots k$. All surviving vertices of level $k-1$ make up the set of vertices $\mathbf{V_k}$ of graph $\mathbf{G_k}$ after the contraction process.

3.3 Summary of Selected Publications

This section summarizes the selected publications about initializing target models and discusses the most important results.

3.3.1 Paper A

Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Hierarchical spatio-temporal extraction of models for moving rigid parts. *Pattern Recognition Letters*, 32(16):800–810, December 2011[1]

[1] Published by Elsevier: http://www.sciencedirect.com/science/article/pii/S0167865511001401

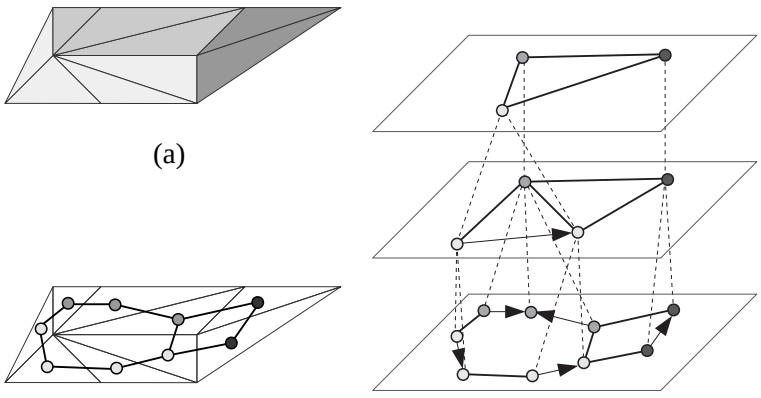


Figure 3.1: Example of an irregular dual graph pyramid. (a) Graph $\mathbf{G_0}$: a triangulation with faces having different gray values. (b) Dual graph $\overline{\mathbf{G}}_\mathbf{0}$ of the triangulation without background vertex for better visibility. (c) Graph pyramid: contracted edges are marked with an arrow. Reprinted from [11] with permission from Elsevier.

3.3.1.1 Summary

This paper proposes a novel approach for the initialization of target models for target structures (related multiple targets) based on an input video. Each rigid entity in the scene of the video (e.g. static background and each body part of a human) is identified and represented by a hierarchical graph model. Besides the video, there is no other input, i.e. no prior knowledge (e.g. number of targets and type of target) and no interaction (e.g. manual initialization by user). Hence, the approach is fully automatic and the built target model is based on the information which can be extracted from the video.

The first step is to extract motion information from the video. This is realized by tracking a set of target points over time. In Paper A and B, a set of target points is tracked with the help of the Kanade-Lucas-Tomasi tracker [20]. The higher the density of the target points the more motion information is collected in the form of 2D trajectories. Please note that the aim of this step is not to tackle difficult cases in tracking (e.g. occlusions), but to extract the motion information for the following processing steps. Hence, any arbitrary tracking approach can be employed. The following steps are based on the extracted trajectories only (the video is not used anymore).

To apply the basic idea: *"things that move together belong together"*, it is necessary to identify if there is some correlation in the motion of the target points (*things*). This is realized by representing the initial spatial configuration of target points (i.e. 2D positions in the first frame) by a triangulated graph. The vertices of the graph represent the target points and its edges their

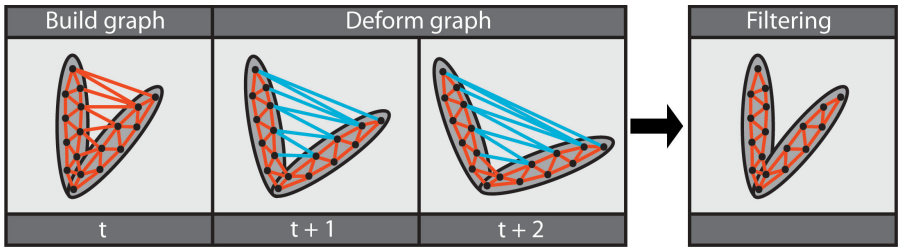


Figure 3.2: From left to right: A triangulated graph is built based on the positions of the target points at time t. While keeping the connectivity of the vertices in the graph it is deformed over time based on the motion encoded in the trajectories. Looking at the deformation over time, it is possible to filter out triangles, which are separating rigid entities and are not of interest for the anticipated aim of the approach (identifying and modeling of rigid entities).

spatial relationships and distances. By keeping the same graph structure (connectivity between vertices), it is possible to observe a deformation of the graph due to the motion encoded in the trajectories. This deformation of the triangulation delivers the necessary information about the correlation in motion.

For each triangle (face) in the graph, the proposed approach answers two interdependent questions: (i) Does it lie on a rigid entity? (ii) To which rigid entity does it belong? The first question is answered in a filtering step, where each triangle is labeled as *relevant* or *separating*. Relevant triangles probably lie on a rigid entity and are processed further. Separating triangles connect rigid entities and are filtered out. Figure 3.2 visualizes the deformation of the triangulation over time and the filtering step.

The answer to the second question (To which rigid entity does it belong?), is found with the help of an irregular dual graph pyramid (see Section 3.2). First the dual graph of the triangulation is built, where each vertex represents a triangle (face) and the edges describe their adjacency. Starting from this dual graph (base level), an irregular dual graph pyramid is built. The grouping criterion used for the building process of the pyramid is the observed motion of the triangles, where their corresponding vertices are merged if their motion is similar. In the ideal case, every vertex in the top level of the pyramid represents one rigid entity of the scene and their receptive field allows to identify the corresponding triangles. Figure 3.3 visualizes the observation of motion and the grouping based on the irregular dual graph pyramid.

3.3.1.2 Discussion

This section discusses the most important results of Paper A. For Paper A, six experiments were conducted. This discussion is about experiment “human 1” and “human dancing 1” (these are the names of the experiments in Paper A). For the convenience of the reader some of the

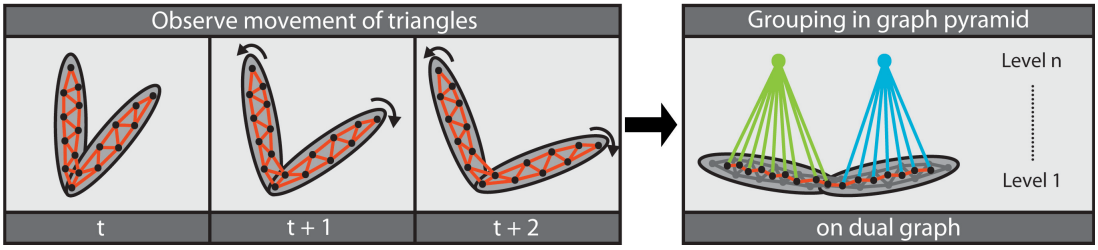


Figure 3.3: Relevant triangles can be grouped based on their similarity in motion with the help of an irregular dual graph pyramid. Each top vertex of the resulting pyramid represents a rigid entity of the scene.

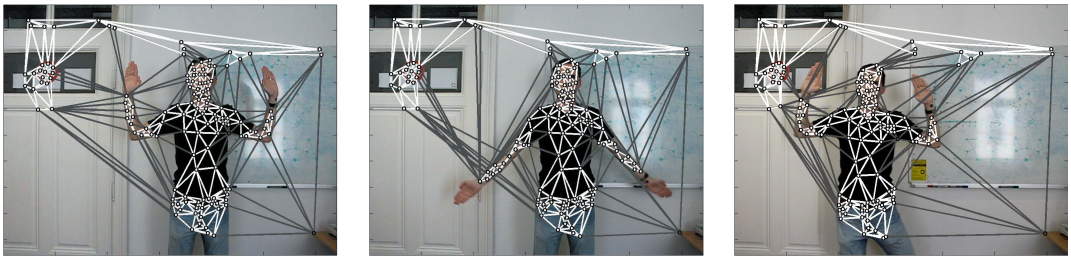


Figure 3.4: Three frames of the video "human 1" with the current state of the graph. White triangles are relevant for the grouping process in the irregular dual graph pyramid. Gray triangles are separating foreground and background and are filtered out. Reprinted from [11] with permission from Elsevier.

corresponding figures of [11] are displayed in this section. For all videos, the ground truth was gathered manually. Furthermore, only complete trajectories are used (trajectories which cover the whole length of the video). Each discussed experiment is rated based on its level of difficulty according to Table 2.2 in Section 2.3.8.

Human 1: This experiment was about identifying the rigid entities of a human based on in-plane motion. The video sequence is self-produced. Information about the video: 640×480 pixels, 860 frames, 134 target points and level of difficulty $= (3, 3, 1, 2.5, 1, 0, 0)$. The grade 2.5 for target results from the partial non-rigid motion, which appears due to the behavior of cloths and skin.

Figure 3.4 shows three frames, where the current state of the graph is drawn. It is noticeable that with the help of the filtering step one can achieve a foreground-background separation for in-plane motion.

Figure 3.5 visualizes the final outcome of the proposed approach. All six rigid entities are identified and all 180 triangles are correctly associated with no errors or outliers. This result is the best case scenario, where each rigid entity of the target is identified and the foreground is correctly separated from the background. Each node in the top level of the irregular pyramid

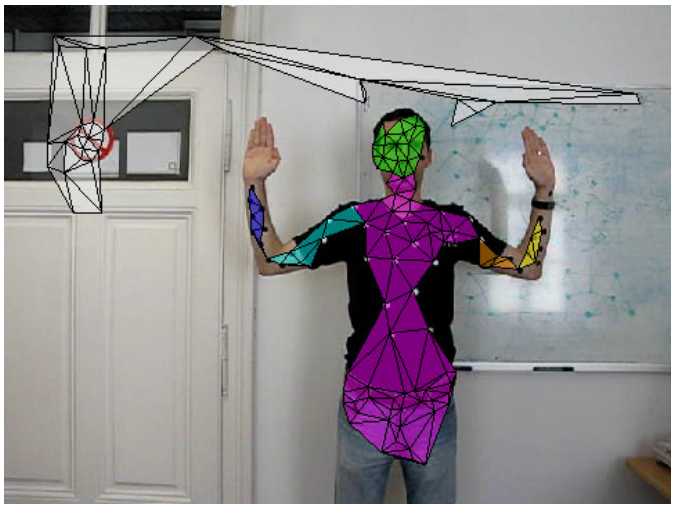


Figure 3.5: Outcome of experiment "human 1". Equally colored triangles belong to the same rigid entity. Reprinted from [11] with permission from Elsevier.

represents one rigid entity. This ideal result could be achieved due to several factors: (i) the video quality was adequate, (ii) the motion of the target was controlled and mostly in-plane, (iii) the non-rigid motion due to clothing was reduced to a minimum.

Human dancing 1: In this sequence, the task is again to identify the rigid entities of a human, but under out-of-plane motion. The source of the video sequence is [98]. Information about the video: 360×240 pixels, 62 frames, 83 target points and level of difficulty $= (3, 3, 1, 3, 1, 0, 0)$. The grade 3 for target results from the non-rigid motion, which appears due to the behavior of cloths and skin.

In comparison to the experiment "human 1", this video is remarkably shorter (less than 10% of "human 1"). Furthermore, the non-rigid motion is especially problematic on the shirt of the human subject. This non-rigidity is difficult to handle for the presented approach as it looks for rigid entities. The filtering step is skipped in this experiment as the provided trajectories only describe foreground.

Figure 3.6 shows the triangulated graph and the final result. Four rigid entities were identified, with 366 correctly associated triangles and 25 outliers. For this experiment, it is difficult to decide for the ground truth. The human subject does not properly rotate the upper arms. Therefore, the four rigid entities, where the upper arms are grouped with the torso, are evaluated as correct. This experiment verifies that the proposed approach also works for out-of-plane motion. I expect even better results (with less outliers) for videos with more explicit movements.

(a) (b)

Figure 3.6: (a) Triangulated graph. (b) Result of grouping by irregular pyramid. Reprinted from [11] with permission from Elsevier.

3.3.2 Paper B

Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Spatio-temporal extraction of articulated models in a graph pyramid. In *8th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition*, volume 6658 of *Lecture Notes in Computer Science*, pages 215–224, Münster, Germany, May 2011. Springer[2]

3.3.2.1 Summary

Paper B continues the work in Paper A. The approach presented in Paper A is able to identify rigid entities based on the observed motion in a video sequence. For a target model it is also important to know how the rigid entities are connected. Hence, the approach proposed in this paper finds the points of articulation for articulated targets. In the current state this approach is able to find points of articulation based on motion in the image plane.

A point of articulation connects two or more rigid entities. The presented approach is based on the knowledge that connected entities are able to move independently, but they always keep the same distance to the point of articulation. Figure 3.7 visualizes the two steps of the proposed approach: (i) generation of hypotheses for points of articulation and (ii) verification of these hypotheses.

In comparison to the grouping of the triangles in the irregular pyramid, a point of articulation is not identified among the target points. It is possible, that a point of articulation does not lie on a foreground entity (see Figure 3.7). For each pair of rigid entities a hypothesis for a connecting point of articulation is created. Let's assume that A and B are two rigid entities. For the generation of the hypothesis, it is not necessary to consider all target points of the rigid entities, but only two pairs of reference points $\{v_1, v_2\}$ for A and $\{v_3, v_4\}$ for B. Having the

[2]Published by Springer: http://link.springer.com/chapter/10.1007/978-3-642-20844-7_22

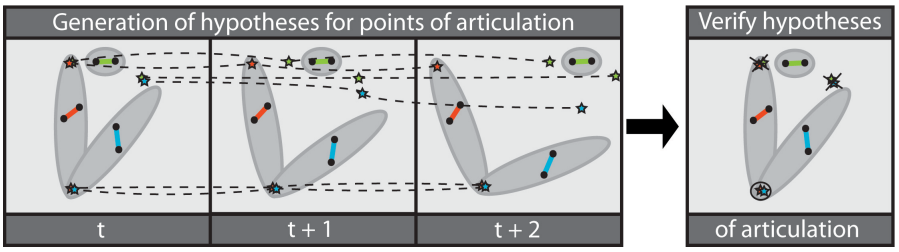


Figure 3.7: Each pair of points (red, green and blue) represents a rigid entity. For each possible permutation of rigid entities a hypotheses for a point of articulation is generated. If a hypothesis is valid, the corresponding rigid entities should always keep the same distance. All hypotheses which do not fulfill this criterion are discarded. The valid hypothesis is marked with a circle and the invalid ones are crossed out.

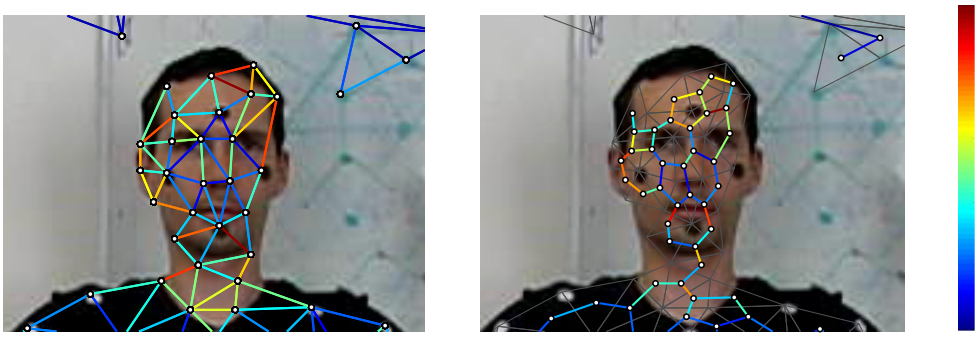
position of these two point pairs at two time instances, the following system of equations can be built:

$$\begin{cases} \mathbf{p}(v_1,t) = (R_A * (\mathbf{p}(v_1,t-1) - \mathbf{p}(a,t-1)) + \mathbf{p}(a,t-1)) + \mathbf{o} \\ \mathbf{p}(v_2,t) = (R_A * (\mathbf{p}(v_2,t-1) - \mathbf{p}(a,t-1)) + \mathbf{p}(a,t-1)) + \mathbf{o} \\ \mathbf{p}(v_3,t) = (R_B * (\mathbf{p}(v_3,t-1) - \mathbf{p}(a,t-1)) + \mathbf{p}(a,t-1)) + \mathbf{o} \\ \mathbf{p}(v_4,t) = (R_B * (\mathbf{p}(v_4,t-1) - \mathbf{p}(a),t-1) + \mathbf{p}(a,t-1)) + \mathbf{o}, \end{cases}$$

where $\mathbf{p}(v,t)$ is the position of reference point v at time t. R_A and R_B are the rotation matrices of the rigid entities and $\mathbf{o}$ is an offset. By solving the system of equations, one can determine the position of the point of articulation $\mathbf{p}(a,t-1)$, the offset $\mathbf{o}$, and the elements of the rotation matrices R_A and R_B, i.e. $\sin(\theta_A), \cos(\theta_A), \sin(\theta_B), \cos(\theta_B)$.

Geometrically, one can imagine that a pair of reference points builds a local coordinate system, where one of the reference points becomes the origin and the second reference point defines the orientation of the two axes. Within this coordinate system, a point of articulation will always have the same position. This is equivalent to the previous statement that rigid entities will keep the same distance to their point of articulation. By using this property, a hypothesis for a point of articulation can be verified. At each frame, the position of the point of articulation is determined using the local coordinate system of each "possibly" connected rigid entity. If the hypothesis is valid, the resulting trajectories of the point of articulation will overlap or at least be similar (see Figure 3.7).

Besides finding the points of articulation, this paper studies the performance of the proposed approach for articulated targets with non-rigid parts, because for such targets the grouping is especially challenging.



(a) (b)

Figure 3.8: Sequence 1: (a) Deformation of the edges of the triangulation over time. (b) Dual graph (in color) of the triangulation (in gray). The edges of the dual graph visualize the dissimilarity of the motion of the corresponding triangles. The color bar describes the used colors, where red is high and blue is low. Reprinted from [13] with kind permission from Springer Science and Business Media.

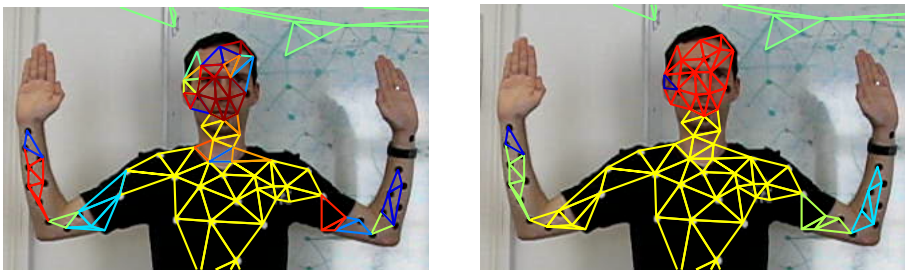
3.3.2.2 Discussion

This section discusses the most important results of PaperB. Three self-produced videos are used for the experiments in this paper. For the convenience of the reader some of the corresponding figures of [13] are displayed in this section. As for Paper C, only complete trajectories are employed (trajectories which cover the whole length of the video). Each discussed experiment is rated based on its level of difficulty according to Table 2.2 in Section 2.3.8.

Sequence 1: This video is also used in Paper B. It shows the in-plane motion of a human. Information about the video: 640×480 pixels, 860 frames, 134 target points and level of difficulty $= (3, 3, 1, 2.5, 1, 0, 0)$. The grade 2.5 for target results from the partial non-rigid motion, which appears due to the behavior of cloths and skin.

Figure 3.8 gives an insight into the motion of the triangles. This figure visualizes the difficulty of this grouping problem. There are cases, where the motion of adjacent triangles is not similar even though they belong to the same rigid entity. For example, tracking target points at the border of a rigid entity to the background is challenging and often results in unexpected drift (e.g. the target points on the head of the human close to the background in Figure 3.8). Furthermore, in local neighborhoods with non-rigid motion, the motion of the target points and the related triangles are ambiguous.

Figure 3.9 shows the outcome of grouping the triangles based on their motion with a global threshold. Due to the difficult nature of the motion (see Figure 3.8), it is not possible to achieve the correct result (six rigid entities) with this simple baseline approach.



(a) (b)

Figure 3.9: Sequence 1: (a) Grouping with global threshold 0.25 results in too many rigid entities. (b) Grouping with global threshold 0.6 results in only one rigid entity for left upper arm and torso instead of two. The different colors visualize the identified rigid entities. Reprinted from [13] with kind permission from Springer Science and Business Media.

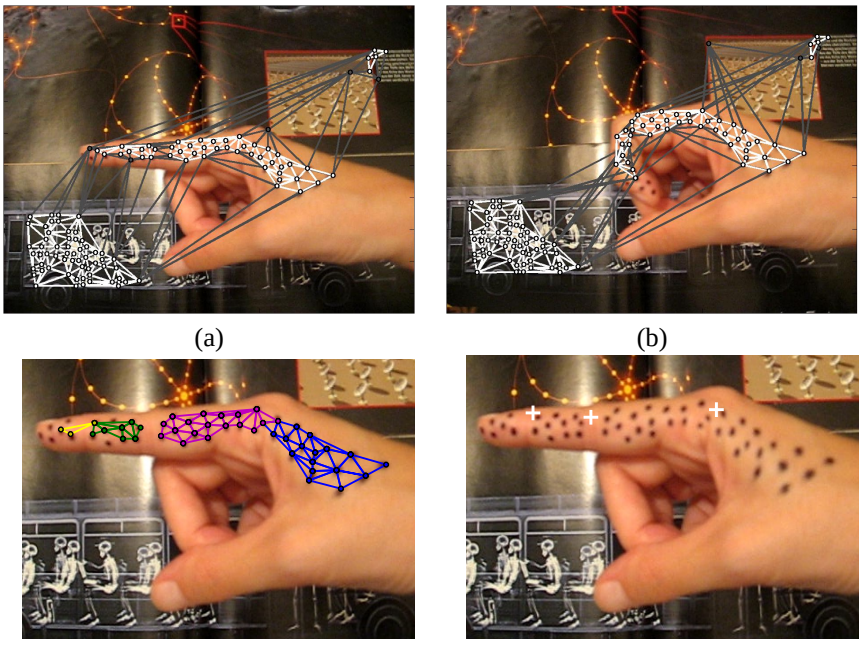
Sequence 2: This video shows the in-plane motion of a finger. Information about the video: 640×480 pixels, 674 frames, 112 target points and level of difficulty $= (3, 3, 2, 2.5, 2, 0, 0)$. The grade 2.5 for target results from the partial non-rigid motion, which appears due to the behavior of skin.

Figure 3.10 visualizes the outcome of the experiment on sequence 2. Even though there is non-rigid motion due to the skin of the finger, the proposed approach is able to find the four rigid entities of the target structure in the foreground (finger). Additionally, three points of articulation are correctly identified and connect the four rigid entities.

Sequence 3: In this synthetic sequence the task is again to identifying the rigid entities of the target structure and to find the point of articulation. Information about the video: 640×480 pixels, 151 frames, 210 target points and level of difficulty $= (3, 3, 2, 3, 2, 0, 0)$. The grade 3 for the difficulty of the target results from the non-rigid behavior of the parts of the articulated target.

In Figure 3.11, it is shown that the rigid entities and their connecting points of articulation are identified. This is possible even under the non-rigid nature of the entities, because the global motion (rotation around the point of articulation) is more significant (traveled distance between two frames is bigger) than the local non-rigid motion.

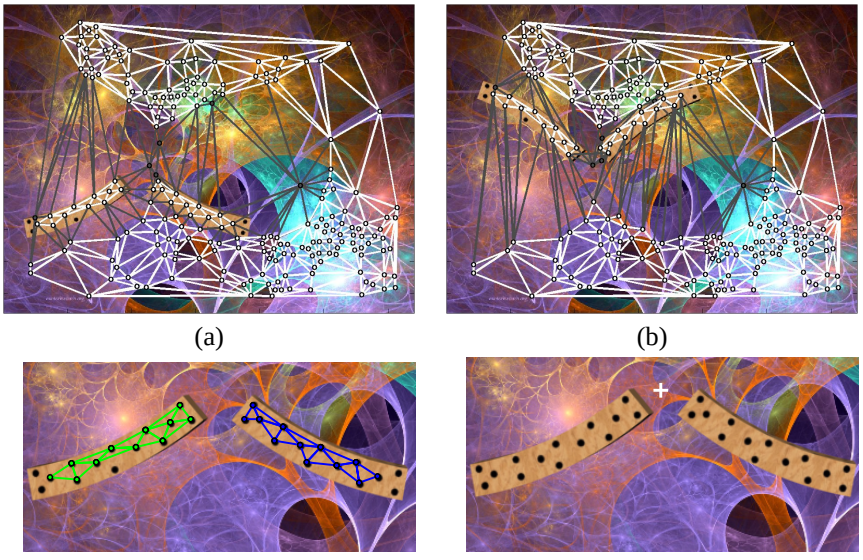
All in all, the proposed approach is able to successfully identify rigid entities and points of articulation in videos which are meant for initialization purposes. In videos with a higher level of difficulty for the factors occlusion and distractors, the quality of the results may decrease. The quality of the output highly depends on the provided trajectories (observation of motion), thus on the performance of the tracking approach.



(a) (b)

(c) (d)

Figure 3.10: Sequence 2: (a) and (b) show the deformation of the triangulation in two selected frames. (c) Result of grouping. (d) Detected points of articulation. Reprinted from [13] with kind permission from Springer Science and Business Media.



(a) (b)

(c) (d)

Figure 3.11: Sequence 3: (a) and (b) show the deformation of the triangulation in two selected frames. (c) Result of grouping. (d) Detected point of articulation. Reprinted from [13] with kind permission from Springer Science and Business Media.

CHAPTER 4

Finding Temporal Correspondences

This chapter presents the ideas for finding temporal correspondences based on a graph model. Section 4.1 motivates and introduces the concept. Section 4.2 describes appearance-based tracking on the example of Mean Shift tracking. Section 4.3 summarizes the selected publications Paper C and D and discusses the most important results.

4.1 Motivation and Concept

Tracking an object of interest in a video is often solved by tracking the corresponding target patch based on its appearance. The employed target model is called appearance-based model. These appearance-based models enable simple trackers [26] to quickly and efficiently find correspondences over time. They describe the appearance of a target patch by its texture or with feature descriptors, which can be as simple as a color histogram of a region of interest. Finding the best correspondence of a target patch is solved by searching for a position, where the extracted visual information is as similar as possible to the appearance-based model. Unfortunately, tracking based on appearance alone often fails to overcome challenging situations like distractors and occlusions (see Section 2.3).

This thesis proposes an approach for tracking target structures which are represented by graph models. Graph models offer high representational power allowing to describe both, the appearance and also the structure of the elements of a target structure. The structure within a target structure encodes spatial relationships and dependencies. It is an important invariant, i.e. it does not change due to the targets' motion. From such a graph model, structural cues can be deduced and enable a tracking approach to deal with tracking tasks of a high level of difficulty. Unfortunately, tracking target structures by finding the most similar graph to the graph model (graph matching) is in general NP-hard. Recently, Solnon et al. [91] showed that in the case

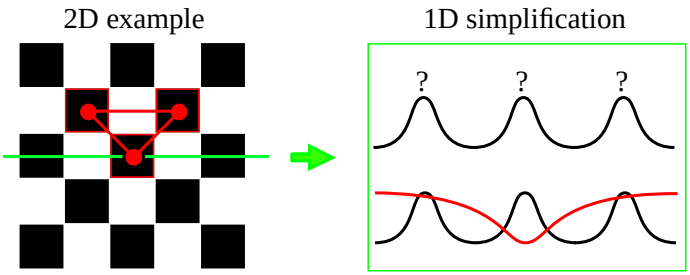


Figure 4.1: Dealing with distractors with the help of structural cues deduced from a graph model. Left: The target structure consists of three target patches, which are the three marked black fields of this checkerboard. As the appearance of all target patches is the same, it is a challenging or even impossible task to find correspondences without mixing up the targets among each other or with other black fields on the board. Right: In the top, one can see a 1D function of the similarity in appearance with three local maxima (at each black field crossed by the green line). Without further information, finding the correct correspondences is error-prone. In the bottom, one can see the same 1D function of similarity in appearance, but with an additional curve representing the deviation from the spatial structure in the target structure (i.e. in the graph model on the left). Now, finding the correspondence can be solved by finding the position, where the similarity in appearance is maximized and the deviation in spatial structure is minimized.

of planar graphs, the matching is P-hard. Nevertheless, degree of the polynominal may be high. Furthermore, the survey by Conte et al. [27] shows that research on graph matching in the temporal domain (video analysis) has been sparse over 30 years. Graph models and graph-based methods are typically used in areas like 2D and 3D image analysis, document processing, biometric identification and image databases [27].

In this thesis, a novel method for finding temporal correspondences for graph models is proposed. The basic concept of the approach is to track an object of interest as a target structure, which is represented by a graph model. The vertices of the graph model represent target patches, which are described by feature descriptors, and its edges encode spatial relationships. Instead of graph matching a novel tracking approach is used, where the correspondence of each vertex in the model is found by combining the hypothesis of a simple appearance-based tracker with structural cues deduced from the graph model. The hypothesis of the appearance-based tracker for each vertex can be generated by any arbitrary tracker (in this thesis: Mean Shift). This hypothesis is combined in an iterative process with the structural cues extracted from the graph model. For each vertex of the graph model a correspondence is found, where similarity in appearance is locally maximized and deviation in structure is locally minimized. The proposed approach approximates a globally optimal solution, which could be achieved by graph matching. Figure 4.1 illustrates and motivates the concept based on an example with distractors.

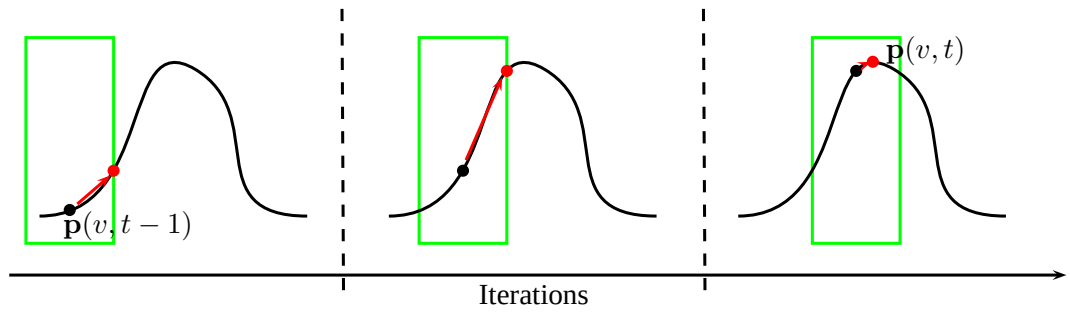


Figure 4.2: Mean Shift tracking procedure. This simple 1D example illustrates the iterative mode seeking process of Mean Shift. Left: Starting from the position of the target v in the previous frame $\mathbf{p}(v, t-1)$, Mean Shift searches within a local neighborhood (green rectangle) for the maximum. An offset vector is generated (red arc) pointing towards the position of the maximum. In appearance-based tracking, this is the position where the similarity in appearance is locally maximized. Middle: The search window is centered around the position of the maximum in the first iteration and the algorithm again looks for the maximum in the local neighborhood. Right: The final position $\mathbf{p}(v, t)$ is found and the algorithm converges as the local maximum of the probability distribution has been found.

4.2 Mean Shift Tracking

The aim of an appearance-based tracker is to find at each frame the position with the highest similarity in appearance with regard to the description of the target. In this thesis, I decided to use a tracking method based on the *Mean Shift* algorithm. The Mean Shift algorithm was proposed in 1975 by Fukunaga et al. [39] and is a non-parametric, iterative procedure. Mean Shift can be used in various tasks like clustering, image segmentation, image smoothing and object tracking [25, 26]. Here, the Mean Shift algorithm is employed to associate the vertices of the graph model from frame to frame finding the locally optimal position for each vertex. Figure 4.2 illustrates the Mean Shift procedure.

My implementation of Mean Shift mainly follows the ideas in [26]. Color histograms are often used in combination with Mean Shift, but as stated by Comaniciu et al. [26] other feature descriptors can be used as well. In this thesis, I employed color histograms (see PaperD) and Sigma Sets (see Paper C) to represent the appearance of the target patches. In the following, it is explained how the Mean Shift algorithm finds temporal correspondences using color histograms as proposed by [26].

In the initial frame, a 3D color histogram is extracted for each vertex v in the graph model from the corresponding target patch at position $\mathbf{p}$ (i.e. the center of mass of the target patch). This color histogram is the *target model* $\hat{q}$ of the target patch and estimates its discrete distribution of color probabilities. Every dimension of the histogram corresponds to one channel of the RBG color space. $\hat{q}$ is usually divided into bins $u = 1 \dots m$ to group similar colors. The discrete

distribution of color probabilities is determined for each bin as follows [26]:

$$\hat{q}_u(\mathbf{p}) = C \sum_{i=1}^{n} k\left(\|x_i^*\|^2\right) \cdot \delta\left(b(x_i) - u\right), \tag{4.1}$$

where C is a normalizing factor such that $\sum_{u=1}^{m} \hat{q}_u = 1$. k is the Epanechnikov kernel [25] and is used to weight the pixels from which the histogram is created by their distance to $\mathbf{p}$. x_i are pixels of the target patch and b is a function mapping a pixel in the 2D image space to the corresponding histogram bin depending on its RGB value. $x_i^* = [0, 1]$ are the normalized positions of the pixels, where the position $\mathbf{p}$ in the center of the patch is $(0, 0)$. The idea behind the weighting with k is that pixels close to $\mathbf{p}$ have a greater influence on $\hat{q}$ than pixels farther away. δ is the Kronecker delta function.

At each frame and in every iteration of the Mean Shift procedure, a candidate model $\hat{p}$ is calculated from the patch within the search window at the current position $\mathbf{p}$ of v by Equation4.1. With this candidate model and the target model the new position $\mathbf{p}$ of the target v is calculated, where this position is the local maximum within the search window [26]:

$$\mathbf{p} = \frac{\sum_{i=1}^{n} x_i \cdot w_i}{\sum_{i=1}^{n} w_i}.$$

w_i weights the pixel positions x_i based on the target and the candidate model:

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{p})}} \cdot \delta\left(b\left(x_i\right) - u\right),$$

Tracking the target patches of a target structure by multiple independent Mean Shift trackers is prone to the problems occurring in difficult tracking tasks. Difficult tracking tasks may come with cluttered backgrounds, articulated or even non-rigid targets, complex and fast motion (high degree of freedom), distractors with a high similarity to the target, and occlusions hiding the visual appearance of the target for a long time. For detailed Information about the level of difficulty of tracking tasks see Section 2.3.

4.3 Summary of Selected Publications

This section summarizes the selected publications about finding temporal correspondences with graph models and graph-based methods and discusses the most important results.

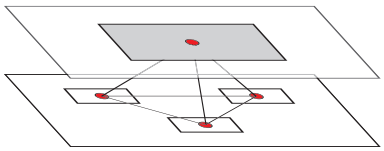


Figure 4.3: A hierarchical spring system represents a target structure by an attributed graph pyramid. The vertex in the top level represents a target patch. In the bottom level, there are multiple vertices, each representing a small target patch. The edges encode the spatial relationships in the bottom and parent-child links in-between the levels. Reprinted from [12] with permission from Elsevier.

4.3.1 Paper C

Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Multi-scale 2d tracking of articulated objects using hierarchical spring systems. *Pattern Recognition*, 44(4):800–810, April 2011[1]

4.3.1.1 Summary

This paper presents a flexible framework to track rigid and articulated target structures through multiple scales and occlusions in 2D. A rigid target structure consists of related multiple target patches and it is represented by an attributed graph pyramid. The bottom level consists of multiple vertices representing local, discriminative patches extracted from the region covering the target and edges encoding their spatial relationships. In the top level there is only one vertex representing the whole target as one patch, which is connected through vertical edges to all vertices in the bottom level. During tracking, the spatial relationships encoded by the edges of this graph model are enforced in a tolerant and spring-like manner. Hence, the representation proposed in this paper is called hierarchical spring system (HSS). Figure 4.3 shows the levels of a HSS for one target structure.

There are several reasons for the hierarchical nature of the target model. The presented approach aims to be applicable for tracking arbitrary targets. Therefore, it is not possible to generally decide, if it is better to track the target as a target patch or as a target structure consisting of multiple target patches. This decision typically depends on the difficulty of the tracking task (see Section 2.3). In general, it is computationally less expensive to describe and track a target as a single target patch. Describing and tracking a target by only one patch is suitable for targets having a uniform appearance (e.g. one color), where no discriminative local features (patches) can be reliably extracted. If the target moves with high speed (speed > 1, see Equation 2.1 in Section 2.3.8), it is easier to follow a target patch covering the whole area of the target than smaller local target patches. On the other hand, a target structure additional includes structural

[1] Published by Elsevier: http://www.sciencedirect.com/science/article/pii/S0031320310005091

information about the composition of the target, which can be useful in cases of occlusions or distractors. For example, if the degree of occlusion is $> 50\%$ and the target is tracked as a single target patch, it is difficult to estimate the state of the target. If the visible area of a target shrinks by 50%, this can be the result of an occlusion or due to the motion of the target along the z-axis (away from the camera). By tracking a target structure, it is possible to evaluate the properties of all target patches and their spatial relationships. During an occlusion the size of the target patches as well as their spatial distances will not change. Therefore, the state of the hidden target patches can be reconstructed from the visible ones. The idea behind the HSS is to incorporate both possibilities (target patch and target structure), and to combine their strengths and overcome their weaknesses.

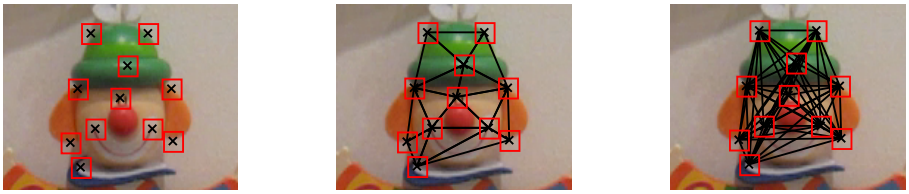
The temporal correspondences are found by a novel, iterative algorithm based on the Mean Shift procedure. Each vertex in the HSS is assigned to a tracker finding its temporal correspondences by combining hypotheses based on appearance and on structure. The structural hypotheses are extracted from the edges in the HSS, which act like springs pushing and pulling the vertices to reduce the deformation of the structure of the graph. Tracking is done in a top-down or bottom-up manner depending on the situation. If the top vertex (one target patch for whole target) can be tracked reliably, the positions of the vertices in the bottom-level (small target patches) are derived from the correspondence (current position) of the top vertex. In ambiguous situations (e.g. during occlusions), tracking is done bottom-up. First the positions of the vertices in the bottom level are determined and then the position of the top-vertex is calculated from them. Switching between these two types of processing allows to increase efficiency as tracking one target patch is computationally less expensive than tracking multiple target patches.

Articulated targets are represented by multiple HSS connected via points of articulation. For example, a human can be represented by ten HSS, one for each body part (head, torso, lower and upper arms, lower and upper legs), connected by nine points of articulation (head-torso, torso-upper arm, etc.). A point of articulation is not related to a visible feature. Its task is to transfer position information between connected rigid parts following the principal that rigid parts always keep the same distance to their points of articulation.

4.3.1.2 Discussion

This section discusses the most important results in greater detail than in PaperC. For Paper C, five experiments were conducted. The following discussion is about experiment 1, 3, and 4. For the convenience of the reader some of the corresponding figures of [12] are displayed in this section.

The proposed approach has been evaluated with publicly available videos [45, 4] and videos



(a) (b) (c)

Figure 4.4: Evaluation of tracking with Mean Shift (a) against proposed approach with HSS with planar graph in bottom level (b) and fully connected graph in bottom level (c). Red boxes visualize the area of the small target patches of the bottom level. Reprinted from [12] with permission from Elsevier.

recorded by myself. For all videos, the ground truth for a rigid target was its center position (the positions of the individual bottom vertices could not be evaluated). Tracking with Mean Shift was the baseline approach. The proposed approach was evaluated for two different HSS: (i) planar (triangulated) graph in bottom level and (ii) fully connected graph in bottom level (see Figure 4.4). For all experiments the size of the small target patches in the bottom level was 13×13 pixels. The size of the target patch of the top level resulted from the size of the rigid targets (e.g. bounding rectangle of head, upper arm, lower arm, etc.). Each discussed experiment is rated based on its level of difficulty according to Table 2.2 in Section 2.3.8.

Experiment 1: The task in this experiment was to track a face under heavy occlusions. Figure 4.5 visualizes results and the degree of occlusion over time. Information about the video: 352×288 pixels, 899 frames, 16 target patches in bottom level and level of difficulty $= (1, 3, 1, 1, 1, 2, 2.5)$.

The outcome of experiment 1 is that, considering the total error over the whole video, the best result could be achieved by a HSS with a fully connected bottom level graph. When a target object undergoes challenging occlusions (medium to high difficulty in Table 2.2), there are several vertices in the bottom level, which do not have a direct, neighboring vertex which is visible. Thus, if the bottom level is a planar graph, there is no direct influence from vertices farther away and the propagation of the necessary position information becomes problematic (takes too many iterations). Therefore, the HSS with a fully connected bottom level is superior. The proposed approach (independent of the bottom level layout) finds temporal correspondences by combining hypotheses based on appearance and on structure. This combination is controlled by a gain that is dynamically calculated from the properties of the target (e.g. similarity of current appearance to target model). There are cases (e.g. the occluder looks similar to the target), where the determined gain weights the influence of appearance and structure in a way so that tracking is hindered (e.g. transfer of false position information within the HSS). This is why

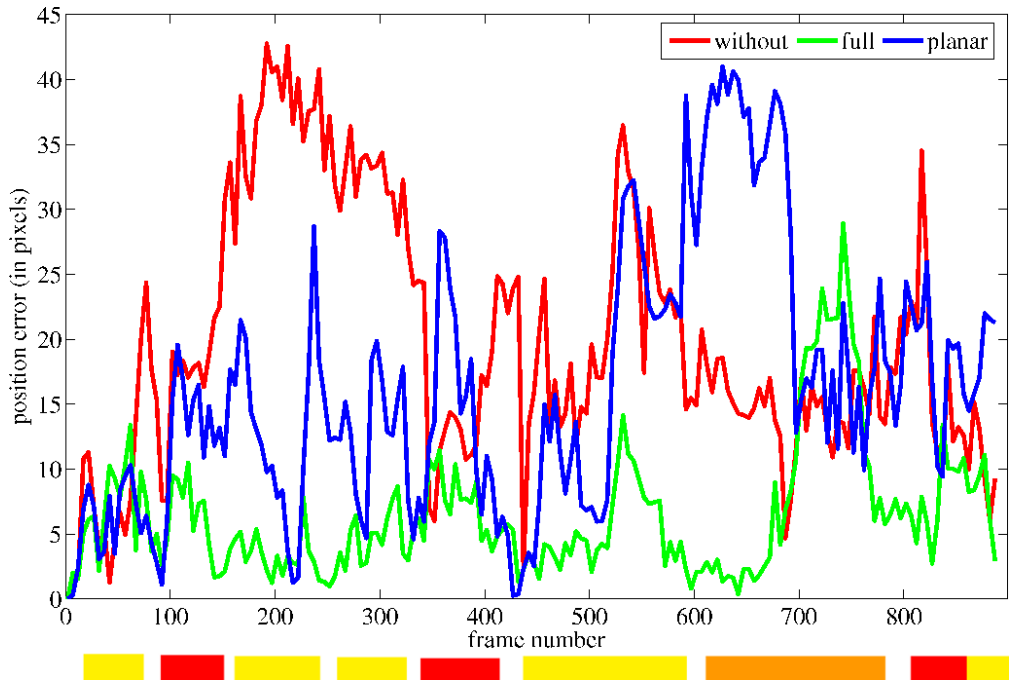


Figure 4.5: Experiment 1: Deviation from ground truth (sum over absolute differences to ground truth). (full) using HSS with a fully connected graph, (planar) using HSS with a triangulated graph, (without) using only tracking with Mean Shift. The color bar at the bottom encodes the degree of occlusion. Yellow: up to 45%, Orange: up to 50% and longer than yellow, Red: up to 62%. Reprinted from [12] with permission from Elsevier.

there are frames, where the result with the HSS (planar and fully connected) is worse than with the baseline approach. The advantage of the baseline approach in such a case is that the targets are tracked independently and tracking errors are not distributed among them. Nevertheless, Figure 4.5 shows that the proposed approach is able to recover from these difficulties.

Experiment 3: The task of this experiment was to track an articulated object through scaling (motion along the z-axis, towards or away from the camera). Figure 4.6 shows the results and the degree of scaling over time. Information about the video: 640×480 pixels, 621 frames, 60 target patches in bottom level and level of difficulty $= (1, 3, 1, 2, 2, 2, 0)$.

The main conclusion from this experiment is that the proposed approaches (both planar and fully connected) result in lower error rates than the baseline approach. In comparison to experiment 1, the proposed approach is not outperformed by the baseline approach in certain frames. Another interesting observation is that for this experiment, there is no big difference between a fully connected and a planar bottom level graph in the HSS. This can be explained as the scaling factor of the target structure is estimated and distributed globally, whereas under

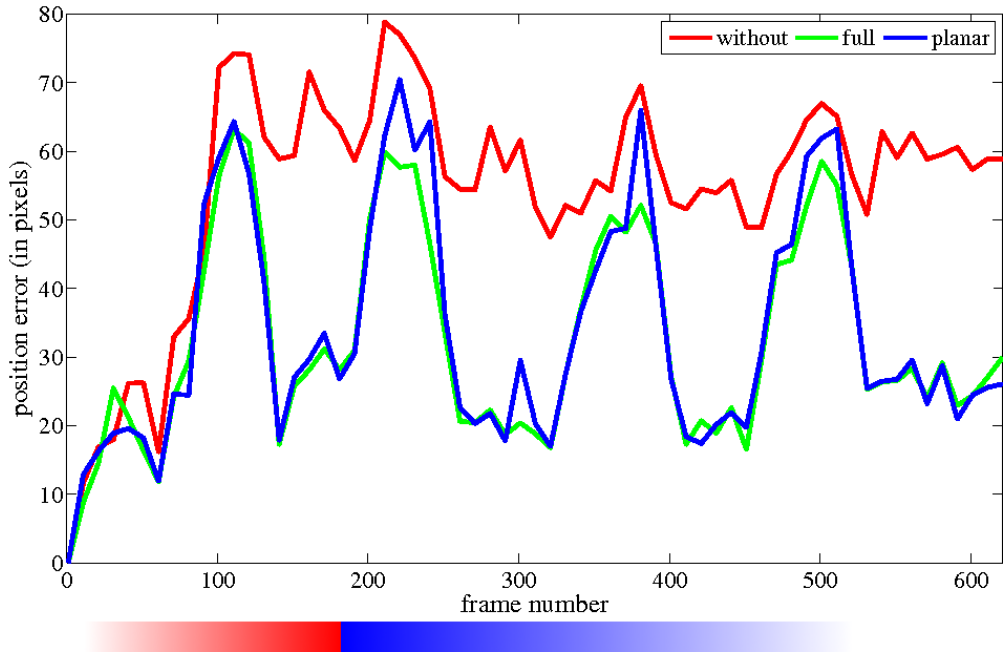
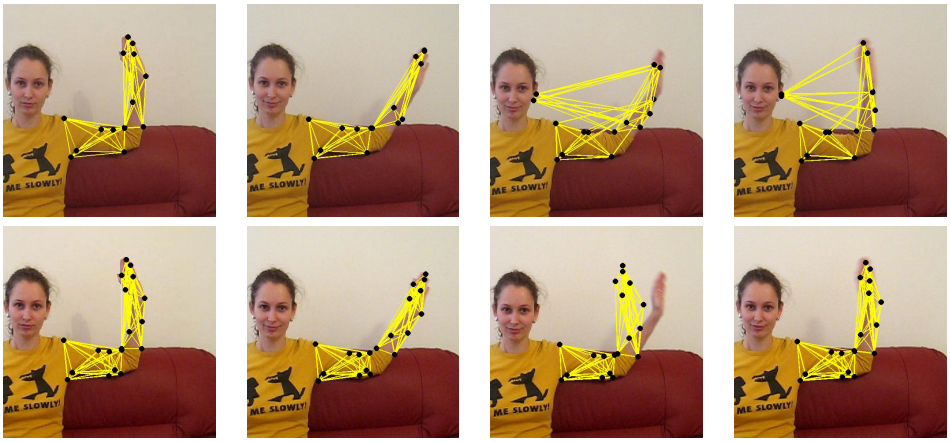


Figure 4.6: Experiment 3: Deviation from ground truth. The position error in pixels is a sum over the absolute difference to ground truth over all rigid parts of the articulated target. The bar on the bottom visualizes the scale change of the target. From 100% in frame 1 to 130% in frame 180 and down to 63% in frame 520. Reprinted from [12] with permission from Elsevier.

occlusion the position information is distributed locally. The peaks in the graphs showing the position error in Figure 4.6 result from the tolerance of the HSS. Minor changes in the spatial configuration of the target patches are compensated by the spring-like behavior of the HSS. Therefore, the change in scaling of the target needs to be remarkable before the HSS reacts and adapts itself.

Experiment 4: This experiment studies the behavior of the HSS under fast motion (speed > 1, see Equation 2.1 in Section 2.3.8). Figure 4.7 shows a selection of interesting frames. Information about the video: 640×480 pixels, 216 frames, 18 target patches in bottom level and level of difficulty $= (1, 3, 1, 2, 2, 2, 0)$.

From Figure 4.7 one can make two relevant observations. In frame 155, tracking with Mean Shift fails to correctly associate two vertices due to a distractor (the color distribution of the face is similar to the hand). The proposed approach suffers due to motion blur, but the spatial arrangement of the target structure stays intact with the help of the structural information and the position information of the upper arm. Hence, the recovery is easier under such conditions. In frame 170, the proposed approach successfully recovers tracking the lower arm, but the baseline



Frame 1 Frame 68 Frame 155 Frame 170

Figure 4.7: Experiment 4: Tracking an articulated object through motion blur. (top) Tracking with Mean Shift and (bottom) my approach with HSS and fully connected graphs. Reprinted from [12] with permission from Elsevier.

approach is still not able to deal with the distractor problem.

All in all, the proposed approach outperforms the baseline approach in cases of occlusion, scaling and fast motion. A HSS with a fully connected bottom level graph delivers equal or superior results in comparison to a planar bottom level graph. An advantage of the proposed combined iterative tracking is that the computational complexity is not influenced by the connectivity of the graph. Hence, using a fully connected bottom level graph does not slow down the tracking. On the contrary, it can speed up the convergence of the combined iterative mode seeking algorithm [66] as the propagation of information is faster (due to the pairwise connection of all vertices).

4.3.2 Paper D

Nicole M. Artner and Walter G. Kropatsch. Structural cues in 2d tracking: Edge lengths vs. barycentric coordinates. In *18th Iberoamerican Congress on Pattern Recognition*, Lecture Notes in Computer Science, page in print. Springer, November 2013[2]

4.3.2.1 Summary

This paper is a continuation of the work in Paper C. The aim of this paper is to analyze the potential of structural information in tracking. In the proposed combined iterative algorithm, temporal correspondences for target structures are found based on an appearance cue and a

[2]Published by Springer: http://www.ciarp.org/xviii/index.php/proceedings

structural cue. The appearance cue is delivered by the Mean Shift algorithm and is based on the similarity to the appearance stored in the model. Structural cues are directly deduced from the current spatial configuration of the target patches in comparison to the initial configuration stored in the model. In Paper C, the spatial relationships of the vertices in the bottom level graph are enforced by the spring-like behavior of the edges. Thus, the structural cue used for HSS are the edge lengths. This structural cue is called *edge cue* in Paper D. The experimental evaluation (see Section 4.3.1.2) shows that the information propagation in triangulated, planar graphs is not efficient enough in challenging situations (e.g. during occlusions). Therefore, one contribution of this paper is a novel structural cue for triangulated bottom level graphs based on barycentric coordinates. This structural cue is called *triangle cue* in Paper D. Barycentric coordinates were already introduced by August Ferdinand Möbius in 1827. They are frequently used in computer graphics [76, 51], but also mathematics [101] and computer vision [29, 86]. Barycentric coordinates allow to describe the position of each vertex in a triangulated graph based on the positions of the three corners (vertices) of any triangle in the graph. The position $\mathbf{p}$ of vertex v can be calculated by a linear combination of the three vertices $\{v_1, v_2, v_3\}$ of a triangle as follows:

$$\mathbf{p}(v) = (x, y, 1)^T = (\beta_1, \beta_2, \beta_3) \cdot \begin{pmatrix} \mathbf{p}(v_1)^T, & 1 \\ \mathbf{p}(v_2)^T, & 1 \\ \mathbf{p}(v_3)^T, & 1, \end{pmatrix}$$

where $\beta_1 + \beta_2 + \beta_3 = 1.0$ are the so-called coefficients.

In addition to the novel structural cue, this paper systematically analyzes the performance of the proposed combined iterative tracking approach with the help of synthetic videos. This synthetic videos allow the comparison of the results against exact ground truth data. A drawback of the experimental evaluation of Paper C is the coarse ground truth data, which does not allow to determine the error in each vertex, but only the deviation from the center of mass of a rigid target structure (e.g. head and torso). Furthermore, the ground truth data of the videos recorded by myself is manually depicted, which already incorporates a certain error.

The focus of the evaluation in this paper lies on different DOF of motion, distractors, occlusions and noise (see Section 2.3). Furthermore, this paper is the first attempt to study the influence of different parameters of the algorithm on the results.

4.3.2.2 Discussion

This section discusses the most important results of Paper D. For the convenience of the reader some of the corresponding figures of [14] are displayed in this section.

Both structural cues and the baseline approach were evaluated with 36 synthetic videos with a size of 400×600 pixels and a length of 10, 30 or 37 frames (2 DOF: 10 frames, 3 DOF: 37 frames, 4 DOF: 30). The target structure in all videos consists of nine target patches and their spatial relationships, which are represented by a graph model. Please note that the proposed approach is not limited to nine vertices. The size of the target patches is 11×11 pixels. Each video can be rated based on its level of difficulty according to Table2.2 in Section 2.3.8. Three different test sets were created for two different target structures:

Test set 1: three videos with level of difficulty $= (1, 2, 1, 1, 1$ to $2, 2, 0)$, where 1 to 2 for motion results from increasing DOF from 2 to 4;

Test set 2: nine videos with level of difficulty $= (1, 2, 1, 1, 1$ to $2, 2, 1$ to $2)$, where 1 to 2 for occlusion results from different degrees from 11% to 66%;

Test set 3: six videos with level of difficulty $= (2, 2, 1, 1, 1$ to $2, 2, 0)$, where 2 for input results from induced noise (Gaussian white noise and Salt & Pepper 10 %).

This results in 36 videos, which were evaluated with different parameter sets. I used three different choices (0–2) for the weight (gain) mixing the appearance cue with structural cue and three different orderings (0–2) of the vertices in the combined, iterative algorithm. All in all, this results in nine different parameters combinations $\{00, 01, 02, 10, 11, 12, 20, 21, 22\}$ and 324 $(36 \cdot 9)$ test cases for each cue.

Figures 4.9 and 4.10 show the results of the edge cue (edge length) and the triangle cue (barycentric coordinates) in comparison to the baseline approach. The difference between Figure 4.9 and 4.10 lies in the target structure. For the results in Figure 4.9 a target structure with equally distributed target patches is used, which results in a graph, where each face has the same geometry and the faces are equal-sided. Figure 4.10 is based on a target structure, where the faces differ in their geometry and no face is equal-sided. The curves in both figures visualize the mean error in a vertex at each frame. This error is calculated as the Euclidean distance from the ground truth position averaged over all vertices in a graph and all video sequences in the corresponding test set.

The most important observation is that in all test cases the best result of the triangle cue outperforms the best result of the edge cue (and the baseline approach). By choosing the best parameter set for the triangle cue it is possible to achieve a total error per vertex (averaged over all 324 test cases) of only $\approx 1,06$ pixels. In contrast to that, the best parameter set for the edge cue results in a total error per vertex of $\approx 6,15$ pixels.

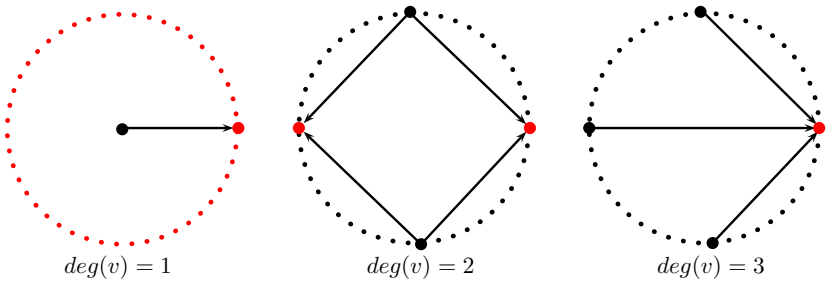


Figure 4.8: Ambiguity of structural cue based on edge length. (a) Vertex degree 1, all positions on circle are minima. (b) Vertex degree 2, two minima. (c) Vertex degree 3, one unique minimum.

Looking at all test cases in Figures 4.9 and 4.10, the best parameter set for the triangle cue (barycentric coordinates) is 20 and the worst is 00. The edge cue (edge lengths) performs best with the set 00 and is the worst with 10.

The edge cue is inferior to the triangle cue, because the reliability of the edge cue highly depends on the layout of the graph. As can be seen in Figure 4.8, the edge cue is ambiguous for vertices with a degree smaller than two.

As edges are a one dimensional entity, they are only capable of providing distance information. Therefore, if a triangle flips it may not be noticed. This problem can be avoided with the triangle cue by checking the signs of the coefficients of the barycentric coordinates. If a triangle flipped the signs of the coefficients change.

Another weakness of the edge cue was already stated in Paper C: the information propagation in a triangulation can become problematic (e.g. in cases of occlusion). The edge cue is determined from the direct neighbors of a vertex (path of length one). Hence, it is a local cue with no direct influence of vertices from farther away. The triangle cue of a vertex can be determined from any other triangle (face) in a graph. Therefore, it is possible to propagate position information within one iteration from a visible triangle to a hidden triangle independent of their distance (path length) in the graph.

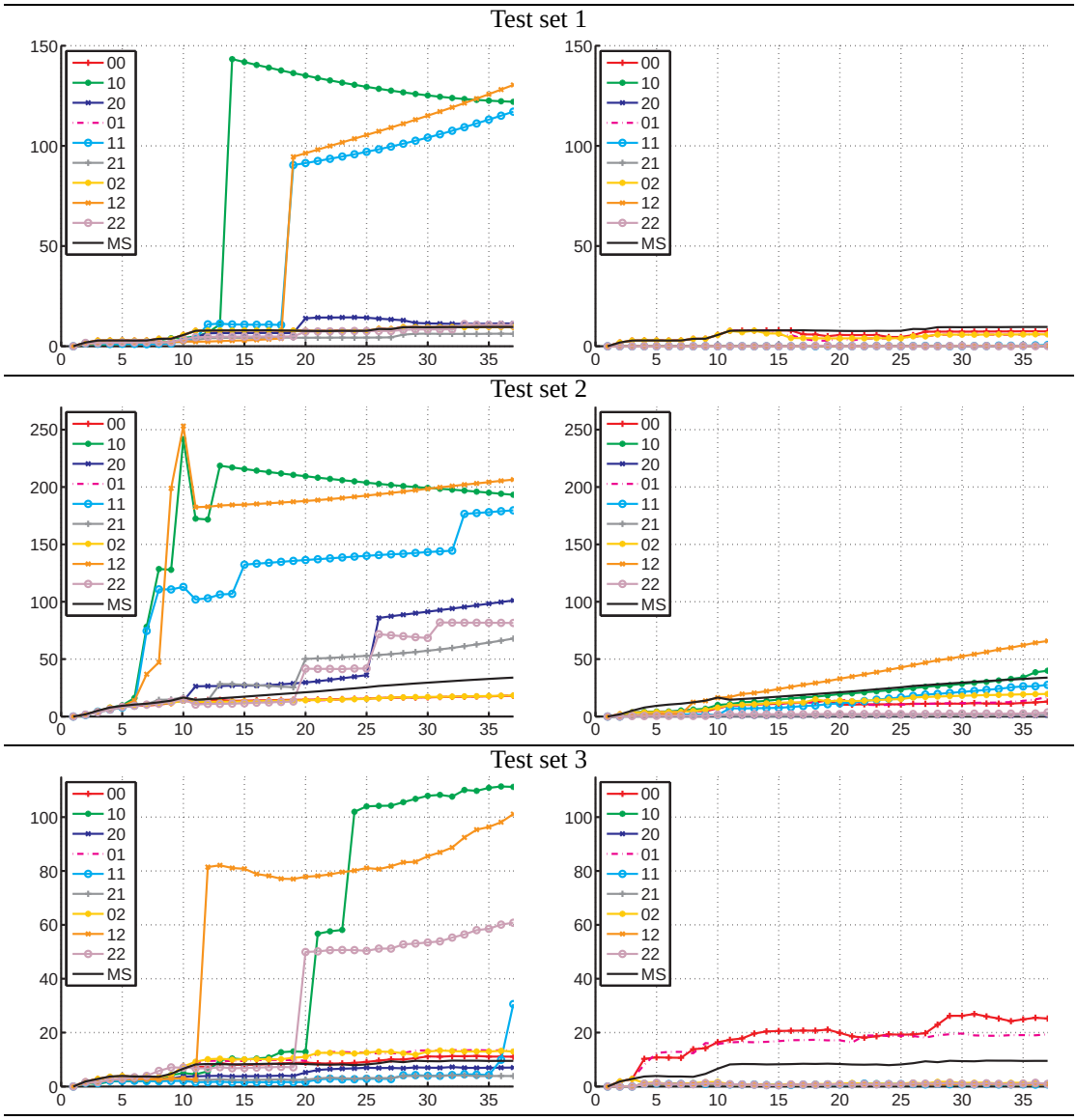


Figure 4.9: Results with regular-sized triangulation (i.e. triangles are equal-sided). Left: edge cue; Right: triangle cue. Vertical axis: error; Horizontal axis: frame. MS = Mean Shift (baseline approach). Reprinted from [14] with kind permission from Springer Science and Business Media.

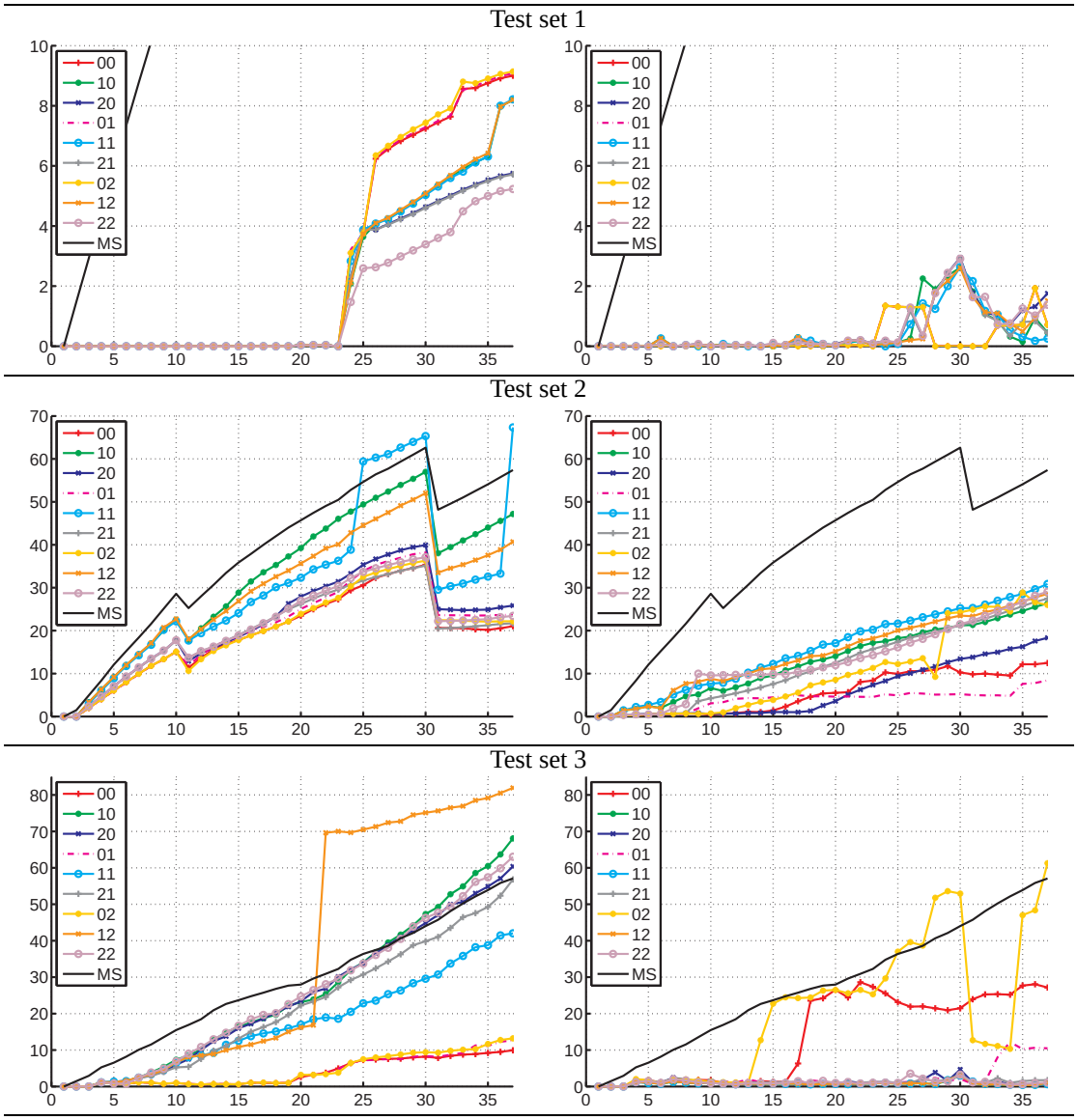


Figure 4.10: Results with irregular-sized triangulation (i.e. triangles are unequal-sided). Left: edge cue; Right: triangle cue. Vertical axis: error; Horizontal axis: frame. MS = Mean Shift (baseline approach). Reprinted from [14] with kind permission from Springer Science and Business Media.

CHAPTER 5

Concluding Remarks

In this concluding chapter the original contributions of this thesis are summarized and possible future work is presented.

5.1 Contributions

The main contributions of this thesis are within the field of tracking, more specifically in the field of tracking related multiple targets.

One core contribution is a novel approach for the initialization of target models, which is presented in Papers A and B. The approach is a hierarchical grouping framework based on irregular dual graph pyramids, which solves the initialization task with an idea inspired by cognitive psychology [58]. In a nutshell, the idea is: *"things that move together belong together"*. Even though, there is related work which also solves the task by grouping pixels or features based on their motion [98], the approach in this thesis additionally considers spatial proximity and relationships. The proposed approach is fully automatic and does not require any prior knowledge about the scene or user interaction. Furthermore, the proposed hierarchical grouping framework allows the usage of different grouping criteria depending on the motion in the video (i.e. motion in the image plane or motion out of the image plane). To the best of my knowledge, there is no work on irregular pyramids to initialize target models based on the motion of tracked target points. The local to global grouping in the irregular pyramid enables the approach to deal with non-rigid motion and tracking errors up to a certain degree (i.e. as long as the motion which results from non-rigid behavior or from tracking errors is insignificant in comparison to the global motion, the grouping will be successful). Besides the identification of the rigid entities in the scene, the approach proposed in Paper B additionally finds the points of articulation connecting the rigid entities of articulated targets. Thus providing a complete model for articulated targets.

Furthermore, the output provides more information than related works as it is a hierarchical description, which can be useful for coarse-to-fine tracking approaches.

Another core contribution is a novel, combined iterative tracking approach for target structures of arbitrary rigid or articulated targets. These target structures consist of related multiple target patches, which are represented by a hierarchical (Paper C) or planar graph model (Paper D). The correspondences for these target structures are established by the proposed tracking approach, which combines appearance and structural cues to iteratively search for the locally optimal solution. Any standard appearance-based tracker (in this thesis Mean Shift) can be employed to deliver the appearance cue (i.e. a correspondence for a target patch based on the similarity to the appearance information stored in the model). The structural cue is deduced from the graph model. In Paper C, the structural cue is determined from the spatial relationships and dependencies encoded in the edges of the graph model. The structural cue simulates the behavior of springs, pushing and pulling the target patches to reduce structural deformations (this is also the reason why the representation is called hierarchical spring system). Paper D proposes a novel additional structural cue based on barycentric coordinates derived from the faces (triangles) of the graph model. For articulated targets, each rigid entity (e.g. body part) is modeled by a separate graph model and points of articulation connect these graph models to transfer position information between them. The rigid entities are allowed to move independent of each other while keeping a fixed distance towards their point(s) of articulation. This novel tracking approach enables a simple appearance-based tracker like Mean Shift to solve challenging tracking tasks by incorporating a graph model and structural cues in a combined iterative tracking process.

All in all, the main goal of this thesis, which is to study the potential of graph-based representations and methods in tracking, could be achieved.

5.2 Future Work

The proposed approach for the initialization of target models is based on analyzing motion encoded in trajectories. At the moment, this approach is only able to process complete trajectories. A complete trajectory is the result of tracking a target point from the first until the last frame. In future, I plan to also incorporate incomplete trajectories, which frequently appear in videos with targets moving out of the image plane. For example if a target rotates around its major axis some target points will disappear and new target points will appear (i.e. incomplete trajectories). The approach presented in Paper A is able to deal with motion out of the image plane, but it is only able to process a reduced set of trajectories (only the complete ones). Thus, the resulting target model contains less information. By incorporating incomplete trajectories the output of

my approach will improve. Paper B presents an approach to identify the points of articulation connecting the rigid entities of articulated targets. At the moment, this approach is limited to motion in the image plane. Therefore, a logical task for future work is to extend this approach to motion out of the image plane. Combined with the previously mentioned future work, the resulting approach will be able to initialize target models consisting of rigid entities and their points of articulation based on arbitrary trajectories in 2D and 3D.

The combined iterative tracking approach offers many possibilities for future work. Currently (in Paper C and D), the tracking is limited to motion with four DOG (degrees of freedom): translation along the x-, y-, and z-axis (e.g. target is moving closer to the camera or farther away) and rotation around the z-axis. In future, I plan to extend the tracking to six DOF. This extension requires structural cues in 3D. The simple structural cue based on distances in PaperC will be replaced by the structural cue based on barycentric coordinates as they are defined for n dimensions. Furthermore, the results in Paper D showed that this novel structural cue is superior. As long as the target moves with only four DOF, it is sufficient to update the appearance information in the target model. When the target moves with six DOF, it becomes necessary to also update the target patches and their relationships. New target patches will appear and should be included in the target model, which requires an update of the structural information. Existing target patches may become invisible, if the target rotates around its major axis. In such cases it will be helpful to pay special attention to the "borders" of the target, because this is where changes will take place.

Bibliography

[1] TWIST-CV project: http://www.prip.tuwien.ac.at/twist/.

[2] Biomotion Lab: http://www.biomotionlab.ca/.

[3] A. Acharya and S. Meher. Robust video denoising for better subjective evaluation. In *2011 International Conference on Image Information Processing*, pages 1–5. IEEE, 2011.

[4] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust fragments-based tracking using the integral histogram. In *IEEE Conference on Computer Vision and Pattern Recognition, 2006*, pages 798–805. IEEE, 2006.

[5] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1442–1456, 2011.

[6] Nicole M. Artner, Csaba Beleznai, and Walter G. Kropatsch. Tracking using a hierarchical structural representation. In *33rd Workshop of the Austrian Association for Pattern Recognition*, volume 254, pages 249–260. Austrian Computer Society, May 2009.

[7] Nicole M. Artner, Adrian Ion, and Walter Kropatsch. Tracking articulated objects using structure. In *Computer Vision Winter Workshop 2009*, pages 51–58. Pattern Recognition and Image Processing Group, TU Wien, February 2009.

[8] Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Coarse-to-fine tracking of articulated objects using a hierarchical spring system. In *13th International Conference on Computer Analysis of Images and Patterns*, volume 5702 of *Lecture Notes in Computer Science*, pages 1011–1018, Münster, Germany, September 2009. Springer.

[9] Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Rigid part decomposition in a graph pyramid. In *14th International Congress on Pattern Recognition*, volume 5856 of *Lecture Notes in Computer Science*, pages 758–765, Mexico, November 2009. Springer.

[10] Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Tracking objects beyond rigid motion. In *7th IAPR-TC-15 International Workshop on Graph Based Representations in Pattern Recognition*, volume 5534 of *Lecture Notes in Computer Science*, pages 82–91, Venice, Italy, May 2009. Springer.

[11] Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Hierarchical spatio-temporal extraction of models for moving rigid parts. *Pattern Recognition Letters*, 32(16):800–810, December 2011.

[12] Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Multi-scale 2d tracking of articulated objects using hierarchical spring systems. *Pattern Recognition*, 44(4):800–810, April 2011.

[13] Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Spatio-temporal extraction of articulated models in a graph pyramid. In *8th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition*, volume 6658 of *Lecture Notes in Computer Science*, pages 215–224, Münster, Germany, May 2011. Springer.

[14] Nicole M. Artner and Walter G. Kropatsch. Structural cues in 2d tracking: Edge lengths vs. barycentric coordinates. In *18th Iberoamerican Congress on Pattern Recognition*, Lecture Notes in Computer Science, page in print. Springer, November 2013.

[15] Nicole M. Artner, S. B. López Mármol, C. Beleznai, and W. G. Kropatsch. Kernel-based tracking using spatial structure (received best paper award). In *Challenges in the Biosciences: Image Analysis and Pattern Recognition Aspects. Proceedings of 32nd OEAGM Workshop*, volume 232, pages 103–114. Austrian Computer Society, May 2008.

[16] Nicole M. Artner, Salvador B. López Mármol, Csaba Beleznai, and Walter G. Kropatsch. Tracking by hierarchical representation of target structure. In *Joint IAPR International Workshop on Structural and Syntactic Pattern Recognition*, volume 5342 of *Lecture Notes in Computer Science*, pages 441–450. Springer, December 2008.

[17] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[18] H. Belghit, N. Zenati-Henda, A. Bellabi, S. Benbelkacem, and M. Belhocine. Tracking color marker using projective transformation for augmented reality application. In *International Conference on Multimedia Computing and Systems*, pages 372–377. IEEE, 2012.

[19] A. G. Bharatkumar, K. E. Daigle, M. G. Pandy, Q. Cai, and J. K. Aggarwal. Lower limb kinematics of human walking with the medial axis transformation. In *Workshop on Motion of Non-Rigid and Articulated Objects*, pages 70–76, Austin, Texas, USA, 1994. IEEE.

[20] S. Birchfeld. KLT: An implementation of the kanade-lucas-tomasi feature tracker. http://www.ces.clemson.edu/ stb/klt/, Oktober 2013.

[21] M. Bister, Jan Cornelis, and Azriel Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, 11(9):605–617, 1990.

[22] G.R. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *4th IEEE Workshop on Applications of Computer Vision*, pages 214–219, 1998.

[23] A.T.S Chan, H.V. Leong, and Shui Hong Kong. Real-time tracking of hand gestures for interactive game design. In *International Symposium on Industrial Electronics*, pages 98–103. IEEE, 2009.

[24] Duan-Yu Chen, Sheng-Wen Shih, and Hong-Yuan Mark Lia. Human action recognition using 2-d spatio-temporal templates. In *International Conference on Multimedia and Expo*, pages 667–670, Beijing, China, 2007. IEEE.

[25] Dorin Comaniciu and Peter Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[26] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, 2003.

[27] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3):265–298, 2004.

[28] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition, 2005*, volume 1, pages 886–893. IEEE, 2005.

[29] F. Dornaika and F. Davoine. On appearance based face and facial action tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(9):1107–1124, 2006.

[30] Ahmed Elgammal. Background subtraction: Theory and practice. Augmented Vision and Reality, pages 1–21. Springer Berlin Heidelberg, 2013.

[31] Baojie Fan, Yingkui Du, Yang Cong, and Yandong Tang. Active drift correction template tracking algorithm. In *19th International Conference on Image Processing (ICIP)*, pages 397–400, 2012.

[32] Pedro F. Felzenszwalb. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79, 2005.

[33] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[34] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[35] Robert Fisher, Ken Dawson–Howe, Andrew Fitzgibbon, Craig Robertson, and Emanuele Trucco. *Dictionary of Computer Vision and Image Processing*. Wiley, 2005.

[36] K. Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *IEEE Conference on Computer Vision and Pattern Recognition, 2012*, pages 1846–1853. IEEE, 2012.

[37] M. Frey, P. Giovanoli, H. Gerber, M. Slameczka, and E. Stussi. Three-dimensional video analysis of facial movements: A new method to assess the quantity and quality of the smile. *Plastic and Reconstructive Surgery*, 104(7):2032–2039, 1999.

[38] Yun Fu and T.S. Huang. hmouse: Head tracking driven virtual computer mouse. In *IEEE Workshop on Applications of Computer Vision, 2007*, pages 30–30. IEEE, 2007.

[39] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

[40] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009*, pages 1746–1753. IEEE, 2009.

[41] J. Gemmell, K. Toyama, C.L. Zitnick, T. Kang, and S. Seitz. Gaze awareness for video-conferencing: a software approach. *MultiMedia, IEEE,* 7(4):26–35, 2000.

[42] A.B. Godbehere, A. Matsukawa, and K. Goldberg. Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In *American Control Conference (ACC), 2012*, pages 4305–4312, 2012.

[43] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.

[44] V. Grabe, H.H. Bulthoff, and P.R. Giordano. Robust optical-flow based self-motion estimation for a quadrotor uav. In *International Conference on Intelligent Robots and Systems*, pages 2153–2159. IEEE, 2012.

[45] Ralph Gross and Jianbo Shi. The cmu motion of body (mobo) database. Technical Report CMU-RI-TR-01-18, Robotics Institute, Pittsburgh, PA, June 2001.

[46] Frank Harary. *Graph Theory*. Addison-Wesley, 1969.

[47] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[48] Y. Haxhimusa, A. Ion, and W.G. Kropatsch. Evaluating hierarchical graph-based segmentation. In *18th International Conference on Pattern Recognition,* volume 2, pages 195–198. IEEE, 2006.

[49] Y. Haxhimusa and W. G. Kropatsch. Segmentation graph hierarchies. In *International Workshops on Structural, Syntactic, and Statistical Pattern Recognition S+SSPR*, volume 3138 of *Lecture Notes in Computer Science*, pages 343–351, 2004.

[50] H. Himberg, Y. Motai, and A. Bradley. A multiple model approach to track head orientation with delta quaternions. *Transactions on Cybernetics*, 43(1):90–101, 2013.

[51] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH, pages 19–26. ACM, 1993.

[52] Yanlong Huang, De Xu, Min Tan, and Hu Su. Trajectory prediction of spinning ball for ping-pong player robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3434–3439, 2011.

[53] D.E. Ilea, C. Duffy, L. Kavanagh, A. Stanton, and P.F. Whelan. Fully automated segmentation and tracking of the intima media thickness in ultrasound video sequences of the common carotid artery. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 60(1), 2013.

[54] Adrian Ion, Hubert Hausegger, Walter Kropatsch, and Yll Haxhimusa. How humans describe short videos. In *Proceedings of the Second International Cognitive Vision Workshop*, 2006.

[55] Adrian Ion, Yll Haxhimusa, and Walter G. Kropatsch. A graph-based concept for spatiotemporal information in cognitive vision. In Luc Brun and Mario Vento, editors, *Graph-Based Representations in Pattern Recognition*, volume 3434 of *Lecture Notes in Computer Science*, pages 223–232. Springer Berlin Heidelberg, 2005.

[56] S. Iwasawa, K. Ebihara, J. Ohya, and S. Morishima. Real-time estimation of human body posture from monocular thermal images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997*, pages 15–20. IEEE, 1997.

[57] Jinwei Jiang and A. Yilmaz. Good features to track: A view geometric approach. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 72–79. IEEE, 2011.

[58] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973.

[59] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2756–2759, 2010.

[60] Walter G. Kropatsch. Building irregular pyramids by dual-graph contraction. *IEE Proceedings - Vision, Image and Signal Processing*, 142(6):366–374, December 1995.

[61] Walter G. Kropatsch, Yll Haxhimusa, and Adrian Ion. *Applied Graph Theory in Computer Vision and Pattern Recognition*, volume 52 of *Studies in Computational Intelligence*, chapter Multiresolution Image Segmentations in Graph Pyramids, pages 3–42. Springer, 2007.

[62] Walter G. Kropatsch, Yll Haxhimusa, and Adrian Ion. Multiresolution image segmentations in graph pyramids. In Abraham Kandel, Horst Bunke, and Mark Last, editors, *Applied Graph Theory in Computer Vision and Pattern Recognition*, volume 52 of *Studies in Computational Intelligence*, pages 3–41. Springer Berlin Heidelberg, 2007.

[63] Ido Leichter, Michael Lindenbaum, and Ehud Rivlin. Mean shift tracking with multiple reference color histograms. *Computer Vision and Image Understanding*, 114(3):400 – 408, 2010.

[64] Olivier Lézoray and Leo Grady. *Image Processing and Analysis with Graphs: Theory and Practice. Digital Imaging and Computer Vision Series*. CRC Press, Taylor and Francis Group, 2012.

[65] Rui Li, Ming-Hsuan Yang, Stan Sclaroff, and Tai-Peng Tian. Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *9th European Conference on Computer Vision*, volume 3952 of *Lecture Notes in Computer Science*, pages 137–150. Springer Berlin Heidelberg, 2006.

[66] Xiangru Li, Zhanyi Hu, and Fuchao Wu. A note on the convergence of the mean shift. *Pattern Recognition*, 40(6):1756 – 1762, 2007.

[67] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679. IEEE, 1981.

[68] Walter G. Kropatsch Mabel Iglesias-Ham, Edel Bartolo García-Reyes and Nicole Maria Artner. Convex deficiencies for human action recognition. *Journal of Intelligent & Robotic Systems*, 64(3):353–364, January 2011.

[69] E. Maggio and A. Cavallaro. Multi-part target representation for color tracking. In *IEEE International Conference on Image Processing, 2005*, volume 1, pages I–729–32. IEEE, 2005.

[70] Emilio Maggio and Andrea Cavallaro. *Video Tracking: Theory and Practice*. Wiley, 2011.

[71] Salvador B. López Mármol, Nicole M. Artner, Mabel Iglesias, Walter G. Kropatsch, Markus Clabian, and Wilhelm Burger. Improving tracking using structure. In *Computer Vision Winter Workshop 2008*, pages 69–76. Slovenian Pattern Recognition Society, February 2008.

[72] Salvador B. López Mármol, Nicole M. Artner, Adrian Ion, Walter G. Kropatsch, and Csaba Beleznai. Video object segmentation using graphs. In *13th Iberoamerican Congress on Pattern Recognition*, volume 5197 of *Lecture Notes in Computer Science*, pages 733–740. Springer, September 2008.

[73] L. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):810 –815, June 2004.

[74] Denys J.C. Matthies, Ngo Dieu Huong Nguyen, Shaunna Janine Lucas, and Daniel Botz. Moving shapes: a multiplayer game based on color detection running on public displays. In *15th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 558–563, New York, NY, USA, 2013. ACM.

[75] Peter Meer. Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, 45(3):269 – 294, 1989.

[76] Mark Meyer, Alan Barr, Haeyoung Lee, and Mathieu Desbrun. Generalized barycentric coordinates on irregular polygons. *Journal of Graphics Tools*, 7(1):13–22, 2002.

[77] C. Micheloni, B. Rinner, and G.L. Foresti. Video analysis in pan-tilt-zoom camera networks. *IEEE Signal Processing Magazine*, 27(5):78–90, 2010.

[78] Jaroslav Nešetřil, Eva Milková, and Helena Nešetřilová. Otakar borůvka on minimum spanning tree problem translation of both the 1926 papers, comments, history. *Discrete Mathematics*, 233(1–3):3 – 36, 2001.

[79] K.C. Ng. Music via motion: transdomain mapping of motion and sound for interactive performances. *Proceedings of the IEEE*, 92(4):645–655, 2004.

[80] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart. A uav system for inspection of industrial facilities. In *Aerospace Conference*, pages 1–8. IEEE, 2013.

[81] Wee-Hong Ong, T. Koseki, and L. Palafox. Unsupervised human activity detection with skeleton data from rgb-d sensor. In *5th International Conference on Computational Intelligence, Communication Systems and Networks*, pages 30–35. IEEE, 2013.

[82] C. Poff, H. Nguyen, T. Kang, and M.C. Shin. Efficient tracking of ants in long video with gpu and interaction. In *IEEE Workshop on Applications of Computer Vision*, pages 57–62, 2012.

[83] M. Popa, L. Rothkrantz, Zhenke Yang, P. Wiggers, R. Braspenning, and Caifeng Shan. Analysis of shopping behavior based on surveillance system. In *International Conference on Systems Man and Cybernetics*, pages 2512–2519. IEEE, 2010.

[84] J.A. Rivera-Bautista, A. Marin-Hernandez, and L.F. Marin-Urias. Using color histograms and range data to track trajectories of moving people from a mobile robot platform. In *22nd International Conference on Electrical Communications and Computers (CONIELECOMP)*, pages 288–293, 2012.

[85] Azriel Rosenfeld. Arc colorings, partial path groups, and parallel graph contractions. Technical Report TR-1524, University of Maryland, Computer Science Center, July 1985.

[86] M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3-d tracking. In *11th International Conference on International Conference on Computer Vision*, pages 1–8, 2007.

[87] O. Schreer, I. Feldmann, N. Atzpadin, P. Eisert, P. Kauff, and H. J W Belt. 3d presence - a system concept for multi-user and multi-party immersive 3d videoconferencing. In *5th European Conference on Visual Media Production*, pages 1–8, 2008.

[88] Mubarak Shah, Krishnan Rangarajan, and Ping-sing Tsai. Motion trajectories. *Pattern Recognition*, 23:1138–1149, 1992.

[89] J. Shi and C. Tomasi. Good features to track. In *IEEE, Conference on Computer Vision and Pattern Recognition, 1994*, pages 593–600. IEEE, 1994.

[90] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[91] Christine Solnon, Guillaume Damiand, Colin Higuera, and Jean-Christophe Janodet. On the complexity of submap isomorphism. In Walter G. Kropatsch, NicoleM. Artner, Yll Haxhimusa, and Xiaoyi Jiang, editors, *Graph-Based Representations in Pattern Recognition*, volume 7877 of *Lecture Notes in Computer Science*, pages 21–30. Springer Berlin Heidelberg, 2013.

[92] Shih-Yu Sun, M. Gilbertson, and B.W. Anthony. 6-dof probe tracking via skin mapping for freehand 3d ultrasound. In *10th International Symposium on Biomedical Imaging*, pages 780–783. IEEE, 2013.

[93] A. Tremeau and P. Colantoni. Regions adjacency graph applied to color image segmentation. *Image Processing*, 9(4):735–744, 2000.

[94] A. Veeraraghavan, R. Chellappa, and M. Srinivasan. Shape-and-behavior encoded tracking of bee dances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):463–476, 2008.

[95] L. Wang, H. Yan, H.-Y. Wu, and C. Pan. Forward-backward mean-shift for visual tracking with local-background-weighted histogram. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1480–1489, 2013.

[96] Daniel Weinland, Remi Ronfard, and Edmond Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, 2006.

[97] Xianliang Wu, J. Housden, N. Varma, YingLiang Ma, D. Rueckert, and K. Rhode. Catheter tracking in 3d echocardiographic sequences based on tracking in 2d x-ray sequences for cardiac catheterization interventions. In *10th International Symposium on Biomedical Imaging*, pages 25–28. IEEE, 2013.

[98] Jingyu Yan and Marc Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):865–877, 2008.

[99] A. Yilmaz, L. Xin, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1531–1536, 2004.

[100] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006.

[101] Paul Yiu. The uses of homogeneous barycentric coordinates in plane euclidean geometry. *International Journal of Mathematical Education in Science and Technology*, 31(4):569–578, 2000.

[102] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004.

[103] Quanshi Zhang, Xuan Song, Xiaowei Shao, Ryosuke Shibasaki, and Huijing Zhao. Unsupervised skeleton extraction and motion capture from 3d deformable matching. *Neurocomputing*, 100(0):170 – 182, 2013. Special issue: Behaviours in video.

[104] H. Zhao and R. Shibasaki. A real-time system for monitoring pedestrians. In *7th IEEE Workshops on Application of Computer Vision*, volume 1, pages 378–385, 2005.

[105] Junda Zhu, Yuanwei Lao, and Y.F. Zheng. Object tracking in structured environments for video surveillance applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(2):223–235, 2010.

[106] W.W. Zou, P.C. Yuen, and R. Chellappa. Low-resolution face tracker robust to illumination variations. *IEEE Transactions on Image Processing*, 22(5):1726–1739, 2013.

Part II

Selected Publications

PAPER **A**

Hierarchical Spatio-Temporal Extraction of Models for Moving Rigid Parts

Published in [11] by Elsevier.

Important notice: This is the electronic version of my thesis. It does not include the selected papers in Part II, but only their references.

PAPER **B**

Spatio-Temporal Extraction of Articulated Models in a Graph Pyramid

Published in [13] by Springer.

Important notice: This is the electronic version of my thesis. It does not include the selected papers in Part II, but only their references.

PAPER **C**

Multi-scale 2D Tracking of Articulated Objects Using Hierarchical Spring Systems

Published in [12] by Elsevier.

Important notice: This is the electronic version of my thesis. It does not include the selected papers in Part II, but only their references.

PAPER **D**

Structural Cues in 2D Tracking: Edge Lengths vs. Barycentric Coordinates

Published in [14] by Springer.

Important notice: This is the electronic version of my thesis. It does not include the selected papers in Part II, but only their references.

Part III

Appendix

APPENDIX **A**

Curriculum Vitae

Personal Data

First name:	Nicole
Middle name:	Maria
Last name:	Artner
Birth date:	23.01.1983
Place of birth:	Oberpullendorf, Austria
Nationality:	Austria

Education

Since 2007	PhD student, Computer Science, Vienna University of Technology, Austria
2005 — 2007	Master, Digital Media, University of applied science Hagenberg, Austria
2002 — 2005	Bachelor, Media Technology and Design, University of applied science Hagenberg, Austria
1997 — 2002	General qualification for university entrance, Business school, Oberpullendorf, Austria

Career History

Since June 2010	Assistant at Vienna University of Technology under Prof. Walter G. Kropatsch
Feb. 2012 — Aug. 2012	Internship with Information and Media Processing Labs. of NEC, Kawasaki, Japan
Jan. 2010 — Dec. 2010	Junior researcher at Austrian Institute of Technology (AIT)
Nov. 2007 — Dec. 2009	Project assistant at Austrian Institute of Technology (AIT), FWF-project: TWIST-CV

Career Related Activities

2013	Editor of Special Issue in Elsevier's journal of Pattern Recognition on GbR2013
2013	Co-Chair of 15th Workshop on Graph-based Representations in Pattern Recognition, Proceedings: LNCS Springer
2013	Member of program committee of 15th International Conference on Computer Analysis of Images and Patterns (CAIP)
2013	Member of program committee of 18th Computer Vision Winter Workshop
2010	Member of program committee of 15th Computer Vision Winter Workshop
2009	Additional reviewer of 13th International Conference on Computer Analysis of Images and Patterns (CAIP)
2009	Local organization of 14th Computer Vision Winter Workshop

Awards

2008	Best Student Paper Award at OAGM 2008 for "Kernel-Based Tracking Using Spatial Structure" [6]

List of Publications

In the following all publications of the author are group by their type and listed in chronological order. In total, this list consists of 25 publications which are made up of six journal papers, 14 peer-reviewed conference and workshop papers, one master thesis and four other publications.

Journal papers

1. Chieh-Han John Tzou, Nicole Artner, Walter Kropatsch, and Manfred Frey. Three-dimensional surface-imaging systems. *Plastic and reconstructive surgery*, 131(4):668e–670e, April 2013.
2. Chieh-Han John Tzou, Igor Pona, Eva Placheta, Alina Hold, Maria Michaelidou, Nicole M. Artner, Walter G. Kropatsch, Hans Gerber, and Manfred Frey. Evolution of the 3-dimensional video system for facial motion analysis: Ten years experiences and recent developments. *Annals of Plastic Surgery*, 69(2):173–185, August 2012.
3. Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Hierarchical spatio-temporal extraction of models for moving rigid parts. *Pattern Recognition Letters*, 32(16):800–810, December 2011.
4. Walter G. Kropatsch Mabel Iglesias-Ham, Edel Bartolo García-Reyes and Nicole Maria Artner. Convex deficiencies for human action recognition. *Journal of Intelligent &*

Robotic Systems, 64(3):353–364, January 2011.

5. Adrian Ion, Nicole M. Artner, Gabriel Peyré, Walter G. Kropatsch, and Laurent D. Cohen. Matching 2d and 3d articulated shapes using the eccentricity transform. *Computer Vision and Image Understanding*, 115(6):817–834, June 2011.
6. Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Multi-scale 2d tracking of articulated objects using hierarchical spring systems. *Pattern Recognition*, 44(4):800–810, April 2011.

International, peer-reviewed conference and workshop papers

7. Nicole M. Artner and Walter G. Kropatsch. Structural cues in 2d tracking: Edge lengths vs. barycentric coordinates. In *18th Iberoamerican Congress on Pattern Recognition*, Lecture Notes in Computer Science, page in print. Springer, November 2013.
8. Samuel de Sousa, Nicole M. Artner, and Walter G. Kropatsch. On the evaluation of graph centrality for shape matching. In *Graph-Based Representations in Pattern Recognition, 9th IAPR-TC-15 International Workshop*, volume 7877 of *Lecture Notes in Computer Science*, pages 204–213. Springer, May 2013.
9. Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Spatio-temporal extraction of articulated models in a graph pyramid. In *8th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition*, volume 6658 of *Lecture Notes in Computer Science*, pages 215–224, Münster, Germany, May 2011. Springer.
10. Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Rigid part decomposition in a graph pyramid. In *14th International Congress on Pattern Recognition*, volume 5856 of *Lecture Notes in Computer Science*, pages 758–765, Mexico, November 2009. Springer.
11. Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Coarse-to-fine tracking of articulated objects using a hierarchical spring system. In *13th International Conference on Computer Analysis of Images and Patterns*, volume 5702 of *Lecture Notes in Computer Science*, pages 1011–1018, Münster, Germany, September 2009. Springer.
12. Nicole M. Artner, Adrian Ion, and Walter G. Kropatsch. Tracking objects beyond rigid motion. In *7th IAPR-TC-15 International Workshop on Graph Based Representations in Pattern Recognition*, volume 5534 of *Lecture Notes in Computer Science*, pages 82–91, Venice, Italy, May 2009. Springer.
13. Nicole M. Artner, Csaba Beleznai, and Walter G. Kropatsch. Tracking using a hierarchical structural representation. In *33rd Workshop of the Austrian Association for Pattern Recognition*, volume 254, pages 249–260. Austrian Computer Society, May 2009.

14. Nicole M. Artner, Adrian Ion, and Walter Kropatsch. Tracking articulated objects using structure. In *Computer Vision Winter Workshop 2009*, pages 51–58. Pattern Recognition and Image Processing Group, TU Wien, February 2009.
15. Nicole M. Artner, Salvador B. López Mármol, Csaba Beleznai, and Walter G. Kropatsch. Tracking by hierarchical representation of target structure. In *Joint IAPR International Workshop on Structural and Syntactic Pattern Recognition*, volume 5342 of *Lecture Notes in Computer Science*, pages 441–450. Springer, December 2008.
16. Salvador B. López Mármol, Nicole M. Artner, Adrian Ion, Walter G. Kropatsch, and Csaba Beleznai. Video object segmentation using graphs. In *13th Iberoamerican Congress on Pattern Recognition*, volume 5197 of *Lecture Notes in Computer Science*, pages 733–740. Springer, September 2008.
17. Adrian Ion, Nicole M. Artner, Gabriel Peyré, Salvador B. López Mármol, Walter G. Kropatsch, and Laurent Cohen. 3d shape matching by geodesic eccentricity. In *Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE Computer Society, June 2008.
18. Nicole M. Artner, S. B. López Mármol, C. Beleznai, and W. G. Kropatsch. Kernel-based tracking using spatial structure (received best paper award). In *Challenges in the Biosciences: Image Analysis and Pattern Recognition Aspects. Proceedings of 32nd OEAGM Workshop*, volume 232, pages 103–114. Austrian Computer Society, May 2008.
19. Salvador B. López Mármol, Nicole M. Artner, Mabel Iglesias, Walter G. Kropatsch, Markus Clabian, and Wilhelm Burger. Improving tracking using structure. In *Computer Vision Winter Workshop 2008*, pages 69–76. Slovenian Pattern Recognition Society, February 2008.
20. Stephan A. Drab and Nicole M. Artner. Motion detection as interaction technique for games & applications on mobile devices. *Pervasive Mobile Interaction Devices (PERMID 2005) Workshop at the Pervasive*, pages 48–51, 2005.

Masterthesis

21. Nicole M. Artner. Analyse und Reimplementierung des Mean-Shift Tracking-Verfahrens. Master's thesis, Fachhochschul-Masterstudiengang, Digitale Medien, Hagenberg, Hagenberg, Austria, August 2007.

Other

22. Walter G. Kropatsch, Nicole M. Artner, Yll Haxhimusa, and Xiaoyi Jiang, editors. *Proceedings of the 9th IAPR - TC-15 Workshop on Graph-based Representations in Pattern*

Recognition, volume 7877 of *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, May 2013.

23. Fuensanta Torres, Walter G. Kropatsch, and Nicole M. Artner. Predict pose and position of rigid objects in video sequences. *Proceedings of International Conference on Systems, Signals and Image Processing, IWSSIP*, 2012.
24. Walter G. Kropatsch, Adrian Ion, and Nicole M. Artner. Describing when and where in vision (abstract only). In *16th Iberoamerican Congress on Pattern Recognition*, volume 7042 of *Lecture Notes in Computer Science*, page 25. Springer, November 2011.
25. Nicole M. Artner. A comparison of mean shift tracking methods. In *12th Central European Seminar on Computer Graphics*, pages 197–204, April 2008.