From Image-based Motion Analysis to Free-Viewpoint Video

Christian Theobalt

Max-Planck-Institut für Informatik
Saarbrücken, Germany

Dissertation zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing)
der Naturwissenschaftlich-Technischen Fakultät I
der Universität des Saarlandes

Eingereicht am 20. Oktober 2005 in Saarbrücken.

Betreuender Hochschullehrer — Supervisor
Prof. Dr. Hans-Peter Seidel, MPI Informatik, Saarbrücken, Germany

Gutachter — Reviewers
Prof. Dr. Hans-Peter Seidel, MPI Informatik, Saarbrücken, Germany
Prof. Dr. Markus Gross, Eidgenössische Technische Hochschule Zürich, CH
PD Dr. Marcus Magnor, MPI Informatik, Saarbrücken, Germany
Dekan — Dean
Prof. Dr. Jörg Eschmeier, Universität des Saarlandes, Saarbrücken, Germany

Datum des Kolloquiums — Date of Defense
27. Dezember 2005 — December 27th, 2005

Christian Theobalt
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
theobalt@mpi-sb.mpg.de

Abstract

The problems of capturing real-world scenes with cameras and automatically analyzing the visible motion have traditionally been in the focus of computer vision research. The photo-realistic rendition of dynamic real-world scenes, on the other hand, is a problem that has been investigated in the field of computer graphics. In this thesis, we demonstrate that the joint solution to all three of these problems enables the creation of powerful new tools that are beneficial for both research disciplines.

Analysis and rendition of real-world scenes with human actors are amongst the most challenging problems. In this thesis we present new algorithmic recipes to attack them. The dissertation consists of three parts:

In part I, we present novel solutions to two fundamental problems of human motion analysis. Firstly, we demonstrate a novel hybrid approach for marker-free human motion capture from multiple video streams. Thereafter, a new algorithm for automatic non-intrusive estimation of kinematic body models of arbitrary moving subjects from video is detailed.

In part II of the thesis, we demonstrate that a marker-free motion capture approach makes possible the model-based reconstruction of free-viewpoint videos of human actors from only a handful of video streams. The estimated 3D videos enable the photo-realistic real-time rendition of a dynamic scene from arbitrary novel viewpoints. Texture information from video is not only applied to generate a realistic surface appearance, but also to improve the precision of the motion estimation scheme. The commitment to a generic body model also allows us to reconstruct a time-varying reflectance description of an actor's body surface which allows us to realistically render the free-viewpoint videos under arbitrary lighting conditions.

A novel method to capture high-speed large scale motion using regular still cameras and the principle of multi-exposure photography is described in part III.

The fundamental principles underlying the methods in this thesis are not only applicable to humans but to a much larger class of subjects. It is demonstrated that, in conjunction, our proposed algorithmic recipes serve as building blocks for the next generation of immersive 3D visual media.

Kurzfassung

Die Entwicklung neuer Methoden der optischen Erfassung und Analyse dynamischer Szenen ist eines der wichtigsten Ziele der computergestützten Bildverarbeitung. Während sich die Bildverarbeitung auf den Analyseaspekt konzentriert, richtet die Computergrafik ihr Augenmerk auf die fotorealistische Darstellung be-

wegter Szenen. Im Rahmen dieser Dissertation wird veranschaulicht, dass es für beide Forschungsdisziplinen von großem Vorteil ist, Erfassung, Analyse und Synthese bewegter Szenen nicht getrennt sondern gemeinsam zu erforschen.

Zu den wichtigsten und schwierigsten Problemen für beide Disziplinen gehören die automatische Auswertung und die realistische künstliche Darstellung menschlicher Bewegung. In dieser Dissertation beschreiben wir neue algorithmische Rezepte, umd diese schwierigen Aufgaben zu lösen. Die Arbeit besteht aus drei Teilen.

In Teil I stellen wir neue Lösungsansätze für zwei Kernprobleme der menschlichen Bewegungsanalyse vor, die Erfassung von mathematischen Bewegungsparametern und die Erzeugung eines kinematischen Menschenmodells. Der erste Lösunsgansatz ist ein neuartiges hybrides Verfahren zur Berechnung menschlicher Bewegungsparameter aus mehreren Videoströmen. Die zweite Methode ermöglicht die vollautomatische Erzeugung eines kinematischen Skelettmodells für beliebige sich bewegende Objekte aus Multivideodaten. Der Hauptvorteil beider Algorithmen liegt darin, dass sie keine optischen Markierungen in einer Szene benötigen.

Teil II dieser Dissertation beschreibt einen neuen modellbasierten Ansatz zur Berechnung und Darstellung dreidimensionaler Videos von Menschen. Ein Betrachter kann die errechneten 3D Videos auf dem Computer in Echtzeit abspielen und interaktiv einen beliebigen neuen Blickwinkel auf die Szene auswählen. Der Kernbaustein des Verfahrens ist ein Algorithmus zur markierungsfreien Form- und Bewegungsanalyse aus Multivideodaten. Um der Person aus beliebigen neuen Blickwinkeln ein fotorealistisches Aussehen zu verleihen, wird mit Hilfe der Bilddaten eine dynamische Oberflächentextur erzeugt. Da dieser 3D Video Algorithmus auf einem generischen Körpermodell basiert, kann man noch einen Schritt weiter gehen und die dynamischen Reflektionseigenschaften der Körperoberfläche abschätzen. Auf diese Weise können dreidimensionale Videos auch unter neuen Beleuchtungsszenarien realistisch wiedergegeben werden.

Ein neues Verfahren zur optischen Analyse sehr schneller Bewegungen wird in Teil III dieser Arbeit vorgestellt. Statt teurer und komplizierter Hochgeschwindigkeitskameras verwendet dieser Ansatz einfache digitale Fotokameras und das Prinzip der Multiblitzfotografie.

Obwohl die hier vorgestellten Verfahren vornehmlich der Analyse und Darstellung menschlicher Bewegungen dienen, sind die grundlegenden Prinzipien auch auf andere dynamische Szenen anwendbar. In ihrer Gesamtheit bilden die hier erläuterten Algorithmen wichtige Bausteine für die Entwicklung der nächsten Generation interaktiver dreidimensionaler Medien.

Summary

In computer vision, it has always been a core research interest to develop algorithms that enable optical capturing and automatic analysis of the visible motion in a dynamic scene. Researchers in computer graphics, on the other hand, used to focus on the inverse problem of generating photo-realistic virtual renditions of dynamic scenes that resemble the real-world equivalent as closely as possible. In recent years, a convergence between the fields has been observed. Ever more powerful imaging technology and computing hardware make it feasible to reconstruct photo-realistic models of real-world scenes from captured image data.

Amongst the most challenging scenes, both in terms of motion analysis and realistic rendition, are scenes involving human actors. In this thesis, we develop algorithmic solutions that enable the optical acquisition of these scenes, the automatic analysis of the visible motion, and their realistic rendition. Furthermore, we show that by integrating solutions to all three problems into one consistent pipeline, novel immersive 3D renditions of humans in motion can be created. This dissertation consists of three parts:

Part I begins with the description of a studio for recording multiple synchronized video streams that we have designed and constructed. The multi-view video material that we acquire in this facility serves as input to our video-based methods for motion analysis and free-viewpoint video reconstruction. Thereafter, two novel solutions to fundamental problems of optical human motion analysis are presented.

The first one is a hybrid method for marker-free full body human motion capture from multi-view video. It jointly uses dynamic shape-from-silhouette volumes and locations of salient body features in the image planes to fit a sophisticated body model to the motion.

The second method enables the fully-automatic reconstruction of kinematic skeleton models of arbitrary moving subjects from multiple video streams. It does with practically no a priori information about the structure of the actor and does not require optical markings on the body. In order to infer the skeleton structure, it analyzes the motion of primitive shapes that have been fitted to dynamic shape-from-silhouette volumes.

In the second part of the thesis, we describe a model-based approach for reconstructing free-viewpoint videos of human actors from only a handful of video streams. The core component of the method is a silhouette-based analysis-by-synthesis approach that enables us to shape-adapt a generic human body model, and to capture the motion of the actor. A realistic time-varying surface appearance of the actor is generated by texturing the model with the appropriately weighted

input video frames. The method enables the photo-realistic rendition of the dynamic scene from arbitrary novel viewpoints in real-time.

In a first extension, we demonstrate that the texture information from camera images can also be used to augment the precision of the motion capture method.

Furthermore, our commitment to a generic body model enables us to not only reconstruct the time-varying scene geometry but also a dynamic surface reflectance model from multi-view video. Our reflectance description comprises a bidirectional reflectance distribution function (BRDF) for each surface point and a time-varying normal field. By this means, 3D videos can be photo-realistically displayed under arbitrary novel lighting conditions.

Standard video cameras are ideal for capturing scenes in which all elements only move at moderate speed. For capturing rapid motion, however, specialized expensive high-frame-rate video equipment would be needed. We have thus developed a novel cost-effective method for capturing high-speed large scale motion that is described in part III. It uses regular digital photo cameras and the principle of multi-exposure photography. We show that this novel measurement principle enables us to capture the rapidly changing articulated hand motion parameters and the motion parameters of the flying ball during a baseball pitch. The highly accurate motion data enable us to create renditions that give new insights into the captured course of motion.

The fundamental principles of the methods described in this thesis are not only applicable to humans but to a much larger class of subjects. Each algorithm can be regarded as a solution to a particular sub-problem in image-based analysis of dynamic scenes. However, we demonstrate that in particular their interplay in larger systems enables innovative novel applications.

Zusammenfassung

Die Entwicklung neuer Algorithmen zur optischen Erfassung und Analyse der Bewegung in dynamischen Szenen ist einer der Forschungsschwerpunkte in der computergestützten Bildverarbeitung. Während im maschinellen Bildverstehen das Augenmerk auf der Extraktion von Informationen liegt, konzentriert sich die Computergrafik auf das inverse Problem, die fotorealistische Darstellung bewegter Szenen. In jüngster Vergangenheit haben sich die beiden Disziplinen kontinuierlich angenähert, da es eine Vielzahl an herausfordernden wissenschaftlichen Fragestellungen gibt, die eine gemeinsame Lösung des Bilderfassungs-, des Bildanalyse- und des Bildsyntheseproblems verlangen.

Zwei der schwierigsten Probleme, welche für Forscher aus beiden Disziplinen eine große Relevanz besitzen, sind die Analyse und die Synthese von dynamischen Szenen, in denen Menschen im Mittelpunkt stehen. Im Rahmen dieser Dissertation werden Verfahren vorgestellt, welche die optische Erfassung dieser Art von Szenen, die automatische Analyse der Bewegungen und die realistische neue Darstellung im Computer erlauben. Es wid deutlich werden, dass eine Integration von Algorithmen zur Lösung dieser drei Probleme in ein Gesamtsystem die Erzeugung völlig neuartiger dreidimensionaler Darstellungen von Menschen in Bewegung ermöglicht. Die Dissertation ist in drei Teile gegliedert:

Teil I beginnt mit der Beschreibung des Entwurfs und des Baus eines Studios zur zeitsynchronen Erfassung mehrerer Videobildströme. Die im Studio aufgezeichneten Multivideosequenzen dienen als Eingabedaten für die im Rahmen dieser Dissertation entwickelten videogestützten Bewegunsanalyseverfahren und die Algorithmen zur Erzeugung dreidimensionaler Videos.

Im Anschluß daran werden zwei neu entwickelte Verfahren vorgestellt, die Antworten auf zwei fundamentale Fragen in der optischen Erfassung menschlicher Bewegung geben, die Messung von Bewegungsparametern und die Erzeugung von kinematischen Skelettmodellen. Das erste Verfahren ist ein hybrider Algorithmus zur markierungslosen optischen Messung von Bewegunsgparametern aus Multivideodaten. Der Verzicht auf optische Markierungen wird dadurch ermöglicht, dass zur Bewegungsanalyse sowohl aus den Bilddaten rekonstruierte Volumenmodelle als auch leicht zu erfassende Körpermerkmale verwendet werden. Das zweite Verfahren dient der automatischen Rekonstruktion eines kinematischen Skelettmodells anhand von Multivideodaten. Der Algorithmus benötigt weder optischen Markierungen in der Szene noch a priori Informationen über die Körperstruktur, und ist in gleicher Form auf Menschen, Tiere und Objekte anwendbar.

Das Thema das zweiten Teils dieser Arbeit ist ein modellbasiertes Verfahren

zur Rekonstruktion dreidimensionaler Videos von Menschen in Bewegung aus nur wenigen zeitsynchronen Videoströmen. Der Betrachter kann die errechneten 3D Videos auf einem Computer in Echtzeit abspielen und dabei interaktiv einen beliebigen virtuellen Blickpunkt auf die Geschehnisse einnehmen. Im Zentrum unseres Ansatzes steht ein silhouettenbasierter Analyse-durch-Synthese Algorithmus, der es ermöglicht, ohne optische Markierungen sowohl die Form als auch die Bewegung eines Menschen zu erfassen. Durch die Berechnung zeitveränderlicher Oberflächentexturen aus den Videodaten ist gewährleistet, dass eine Person aus jedem beliebigen Blickwinkel ein fotorealistisches Erscheinungsbild besitzt. In einer ersten algorithmischen Erweiterung wird gezeigt, dass die Texturinformation auch zur Verbesserung der Genauigkeit der Bewegunsgsschätzung eingesetzt werden kann. Zudem ist es durch die Verwendung eines generischen Körpermodells möglich, nicht nur dynamische Texturen sondern sogar dynamische Reflektionseigenschaften der Körperoberfläche zu messen. Unser Reflektionsmodell besteht aus einer parametrischen BRDF für jeden Texel und einer dynamischen Normalenkarte für die gesamte Körperoberfläche. Auf diese Weise können 3D Videos auch unter völlig neuen simulierten Beleuchtungsbedingungen realistisch wiedergegeben werden.

Teil III dieser Arbeit beschreibt ein neuartiges Verfahren zur optischen Messung sehr schneller Bewegungen. Bisher erforderten optische Aufnahmen von Hochgeschwindigkeitsbewegungen sehr teure Spezialkameras mit hohen Bildraten. Im Gegensatz dazu verwendet die hier beschriebene Methode einfache Digitalfotokameras und das Prinzip der Multiblitzfotografie. Es wird gezeigt, dass mit Hilfe dieses Verfahrens sowohl die sehr schnelle artikulierte Handbewegung des Werfers als auch die Flugparameter des Balls während eines Baseballpitches gemessen werden können. Die hochgenau erfaßten Parameter ermöglichen es, die gemessene Bewegung in völlig neuer Weise im Computer zu visualisieren.

Obgleich die in dieser Dissertation vorgestellten Verfahren vornehmlich der Analyse und Darstellung menschlicher Bewegungen dienen, sind die grundlegenden Prinzipien auch auf viele anderen Szenen anwendbar. Jeder der beschriebenen Algorithmen löst zwar in erster Linie ein bestimmtes Teilproblem, aber in Ihrer Gesamtheit können die Verfahren als Bausteine verstanden werden, welche die nächste Generation interaktiver dreidimensionaler Medien ermöglichen werden.

Acknowledgements

First and foremost I would like to thank my supervisor Prof. Dr. Hans-Peter Seidel who gave me the opportunity to do research in such an excellent and inspiring environment as the Max-Plack-Institut für Informatik (MPI). He gave me the freedom to pursue my own ideas and supported my work by giving me his scientific advice and providing me with the technical equipment I needed.

I am also indebted to Dr. Marcus Magnor who has been an invaluable scientific and personal advisor in all of my research. We have worked together on all of the projects that are described in this thesis, and I am thankful to him for being a reviewer of this dissertation.

Furthermore, I would like to thank Prof. Dr. Markus Gross who kindly agreed to serve as an external reviewer, which I am grateful for.

My special thanks go to all my former and present colleagues in the Computer Graphics Group at the MPI. Without their cooperation, their professional advice and without the inspiring discussions that we had, many of my research projects would have been impossible. I also thank them for contributing to the great atmosphere in the group. In particular, I owe thanks to Naveed Ahmed, Edilson de Aguiar, Irene Albrecht, Joel Carranza, Jörg Haber, Hendrik Lensch, Ming Li, Pascal Schüler, Holger Theisel, and Gernot Ziegler who were co-authors on some of my papers. To Christian Rössl and Hartmut Schirmacher I am very grateful for their technical advice, especially when I was a new PhD student. I'd also like to thank Marcus Weber for contributing to the success of the baseball project.

Many people kindly allowed me to record them for my research. Anna Hagermark and Harald Krytinar gave us the possibility to record their impressive dancing performance for the free-viewpoint video project. Edda Happ, Kolja Kähler, and Kuangyu Shi also acted as models for our research. Without the help of Thorsten Dehm from the Saarlouis Hornets, who was a very patient and persistent baseball pitcher, the project on motion capture of rapid events would have been impossible. To all of them I owe many thanks.

Many thanks also go to ATI Corporation who greatly supported my research by awarding me a fellowship.

Without the help of non-scientific employees of the institute, it would have been impossible to build our multi-view video acquisition studio and the measurement facility for the baseball project. Thus, my special thanks go to Michael Laise and Axel Köppel from the MPI technical staff for helping us in setting up both systems. I'd also like to thank the Rechnerbetriebsgruppe for kindly providing us with sufficient storage capacity for our data.

Finally, I'd like to thank my whole family and in particular my parents, Ingeborg and Franz-Josef Theobalt, who always supported and encouraged me. I'd also like to thank Alexandra Chapko for beeing always there for me.

Contents

Chapter 1
Introduction

Humans possess many senses to perceive their environment, but none of them is such a rich source of information to them as the visual sense. The explanation for this predominance can be found in evolution theory. Vision provides spatially accurate information from a distance. It enables humans to efficiently recognize enemies and to analyze their motion, as well as to track the movements of a prey. The combination of eye and visual cortex in the the brain forms a very powerful system for capturing and analyzing visual impressions of the environment [Palmer99].

However, from our own daily experience we know that the visual sense is not only a powerful analytical tool but also a rich source of psychological stimuli. Joy, sadness, or compassion are just a few feelings which can be induced by visual impressions. Visual media, such as television or cinema, capitalize on this fact that visual stimuli are the gate to the human fantasy. They can trigger the feeling of immersion into a virtual environment exposed to the viewer.

Two disciplines of computer science, computer vision and computer graphics, are dedicated to the visual sense. The former one intends to simulate and enhance the analytical capabilities of the human visual system through cameras and computational image analysis. The latter one aims at generating photo-realistic synthetic renditions of scenes that are visually indistinguishable from their real-world equivalents. In recent years, researchers from both disciplines have learned that the problems of optical scene capture, scene analysis and scene rendition should not be treated separately. The advent of ever more powerful computers and advanced imaging sensors has rendered it feasible to generate virtual models of real-world scenes by reconstructing them from image data.

Amongst the most important real-world scenes, both for researchers working in computer vision and computer graphics, are scenes involving human actors. Here, the most challenging problem for the vision researcher is to estimate a mathematical model of human motion from the captured image data. The graph-

ics researcher is facing the problem of creating photo-realistic virtual humans that can fool even the human eye which is not forgiving the slightest inaccuracy in appearance. In this thesis we develop algorithmic solutions that enable the optical *acquisition* of these scenes, the automatic *analysis* of the visible motion, and their realistic *rendition*.

In principle, each of the methods that we propose can be regarded as a solution to one of these sub-problems. However, in particular their interplay in larger systems enables us to develop novel applications. To proof this, we show that mathematical models of human motion and dynamic human appearance that have been reconstructed from image data, can be used to generate novel free-viewpoint renditions. The methods described in this thesis are tailored to scenes involving human actors. However, the fundamental principles are applicable to a much larger class of scenes, and we will elaborate more on this in the respective chapters of this work.

1.1 Structure of the Thesis and Main Contributions

In Chap. 2 we give some technical and theoretical background that is important for the understanding of the chapters to follow. Chapters 3 through 10 are divided in three parts according to their main focus. We conclude in Chap. 13 with a description of future perspectives. The systems and algorithms that form the scientific basis of this thesis have been published before in a variety of peer-reviewed conference and journal articles. The main scientific contributions as well as the appropriate references are briefly summarized in the following.

1.1.1 Part I: Marker-free Optical Human Motion Analysis

In Part I of the thesis, novel algorithmic solutions to two core problems of human motion analysis from video are presented, *motion capture* and *model estimation*. The former one is the problem of inferring a mathematical description of human motion from image data. The latter one is the problem of automaticlly constructing an appropriate virtual body representation.

In Chap. 3 we illustrate the importance of both problems, review related work from the literature, and give theoretical and technical background information. The nuts and bolts of a flexible and versatile studio that we have designed and built in order to record synchronized multi-view video streams are described in Chap. 4 [Theobalt03c]. The multi-view video (MVV) streams are the input data to all algorithms that are described in Part I and Part II of this thesis.

In Chap. 5 we present a novel hybrid approach to model-based marker-free optical motion capture [Theobalt02a, Theobalt02b, Theobalt04e]. It jointly uses real-time voxel-based visual hull reconstruction and feature tracking to estimate the motion of a human skeleton from multiple video streams.

Human motion capture methods require a model of the body that represents its shape and kinematic properties. We present a novel non-intrusive approach to estimating a human body model from multiple synchronized video streams in Chap. 6 [Theobalt04d, de Aguiar04]. It reconstrcuts a sequence of shape-from-silhouette models and fills each volume with simple shape primitives. From their motion over time a complete kinematic skeleton is reconstructed even though no a priori information about the recorded subject is available. The method is equally appropriate for estimating the kinematic structure of both human an animal subjects.

1.1.2 Part II: Capturing Appearance and Motion - Free-Viewpoint Video

Part II of this thesis illustrates that a motion capture approach can serve as the core component of a model-based system for reconstructing free-viewpoint videos of human actors. In Chap. 7 we describe the scope of 3D video in general and free-viewpoint video in particular, and give some technical and theoretical background information.

A novel model-based system for reconstructing and rendering free-viewpoint videos of human actors from multi-view video is presented in Chap. 8 [Carranza03, Theobalt04b, Magnor04]. The central element of the method is a newly-developed silhouette-based analysis-by-synthesis approach. This approach is used for customizing a generic body model such that it matches its real-world equivalent, and for capturing the pose of the human at each time step of a multi-view video sequence. This method also lends itself to a parallel implementation that exploits the compartmentalized nature of the pose determination problem [Theobalt03b]. A realistic dynamic surface appearance of the human is generated by projectively texturing the model with the appropriately blended input camera views. The free-viewpoint videos can be rendered in real-time and the virtual viewpoint can be arbitrarily changed.

In Chap. 9 we propose an augmented version of the original silhouette-based motion capture method that incoporates texture information into the pose estimation process [Theobalt03a, Theobalt04c]. We have developed a predictor-corrector-scheme in which a 3D motion field is reconstrcuted from 2D optical flows that enables the correction of pose inaccuracies after silhouette-fitting.

If virtual environments shall be augmented with 3D renditions of real-world

people, one has to realistically display them under the novel virtual lighting conditions. To serve this purpose, the surface reflectance properties have to be known. We thus further enhance our free-viewpoint video approach in Chap. 10 such that it is able to not only capture dynamic scene geometry but also dynamic surface reflectance properties from multi-view video [Theobalt05]. To serve this purpose, we have developed a dynamic reflectometry approach that allows us to capture a bidirectional reflectance distribution function for each surface point, as well as a time-varying normal field from only a handful of video streams. In order to optimize the multi-view texture-to-model consistency prior to reflectance estimation we have also developed a novel image-based warp-correction method. This way, relightable 3D videos are generated that can be rendered in real-time on standard graphics hardware.

1.1.3 Part III: High-Speed Motion Estimation - Exploring the Limits of Photo Camera Technology

While the methods presented in the first two parts where dedicated to analyzing and rendering human motion of moderate speed, in Part III we examine ways to capture and visualize very rapid motions. In Chap. 11 we illustrate the importance of high-speed motion capture in general and our approach in particular. Furthermore, some technical background is presented by reviewing related work in the field.

A common way to capture image data of a high-speed event is to record with an expensive high-frame-rate video camera. In Chap. 12 we present a novel cost-effective principle to acquire high-speed motion that has a large spatial extent [Theobalt04a]. Our method employs the principle of multi-exposure photography using regular off-the-shelf digital photo cameras. We demonstrate the performance of the principle by capturing both the parameters of motion of the flying ball as well as the pose parameters of the pitcher's hand during a baseball pitch. Our data enable visualizations of the high-speed events from arbitrary novel viewpoints.

Chapter 2

Preliminary Techniques and Basic Definitions

In this chapter some general theoretical background is given and elementary techniques are described that many of the projects in this thesis capitalize on.

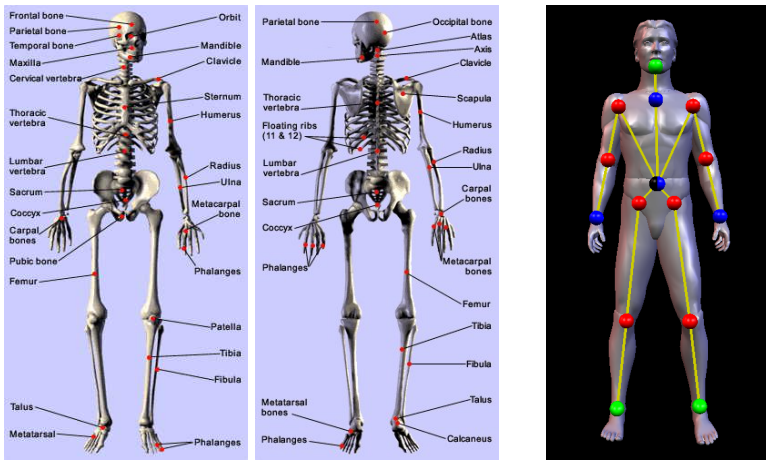
We begin in Sect. 2.1 with a description of general principles of how to model the shape, the appearance and the kinematics of a human in a computer. Although we have developed customized body models in the course of each of the projects described in this thesis, they all are based on common principles.

Video and photo cameras are the sensors with which we capture all the information we need, in order to estimate body motion and to reconstruct 3D videos. To us it is of fundamental importance to simulate the imaging process of the cameras by means of a mathematical camera model. The correspondence between a real camera and its computational equivalent, the process of camera calibration, and the imaging geometry of camera pairs are outlined in Sect. 2.2.

We conclude this chapter in Sect. 2.3 with a description of image processing techniques that are applied in several of the projects that form the basis of this thesis.

2.1 The Human Body and its Digital Equivalent

The human body is a highly complex system. Both its optical appearance as well as its physical and kinematic properties are the result of the interplay of many physiological components. Already the appearance of the skin, for example, is the result of a non-trivial light interaction on the body surface, fine-grain structural pigmentation, and the deformation of muscles and connective tissue.



(a) (b)

Figure 2.1: (a) Anatomical skeleton of the human body (images taken from [myd]). (b) A digital body model that mimics the geometry and the kinematics of a human.

The kinematic properties of the human body are mainly determined by its bone skeleton (Fig. 2.1a). It consists of 206 bones and more than 200 interconnecting joints [Sobotta01]. Muscles that are attached to the bones via tendons are the actuators of the body that move it into a certain stance. A realistic computational body model has to comprise appropriate representations for the kinematics as well as the appearance of the real human. Representation methods that serve this purpose are described in the following two subsections.

2.1.1 Modeling the Kinematics of the Human Body

The equivalent of the human skeleton in a computational model is a *kinematic skeleton*. It mathematically models a hierarchical arrangement of joints and interconnecting bones. A kinematic skeleton follows the principle of the *kinematic chain* [Murray94]. A kinematic chain is a linear arrangement of connected rigid body segments. The relative orientation between one segment and the subsequent element in te chain is controlled via a *rigid body transformation*. A rigid body transformation jointly describes a rotational and a translational transformation between the local coordinate frames of adjacent rigid bodies. In consequence, a kinematic chain is a hierarchical structure. Transformations at a higher level of the hierarchy (i.e. closer to the initial element in the chain) influence all segments on the succeeding hierarchy levels, but no segment on the

preceding levels. The human skeleton is usually approximated by a collection of kinematic sub-chains, e.g. the arm or the leg, which originate from a common root joint located in the torso area. In Fig. 2.1b the skeleton of a body model employed in the Chaps. 8, 9, and 10 is illustrated. To keep the model complexity moderate only the most important joints in the human skeleton are represented.

We have seen that the pose of a human can be specified via rigid body transformations. The space of all rigid body transformations in 3D is a group known as the special Euclidean group $SE(3)$. It is common practice to specify an element of $SE(3)$ as a linear transformation of homogeneous coordinates (i.e. as a linear transformation in the projective space $\mathbb{P}^3$, see [Hartley00] for a detailed introduction to projective spaces). If $\mathbf{p} = (x,y,z)^T$ is a point in three-dimensional Euclidean space, then $\overline{\mathbf{p}} = (x,y,z,1)^T$ is its equivalent in homogeneous coordinates. Vice versa a point $\overline{\mathbf{p}} = (x,y,z,q)^T$ is the homogeneous representation of the Euclidean point $\mathbf{p} = (x/q, y/q, z/q)^T$. A 3D rigid body transform in projective notation is a 4×4 matrix of the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{R} & \vec{t} \\ 0 & 1 \end{bmatrix} \tag{2.1}$$

where $\vec{t} \in \mathbb{R}^3$ is the translational component, and $\mathbf{R}$ is a 3×3 matrix controlling the rotational component. The space of 3×3 rotation matrices $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3\times3} \mid \mathbf{R}\mathbf{R}^T = \mathbf{I},\ det\mathbf{R} = \pm 1\}$ forms a group under matrix multiplication.

If all the rigid body transformations in kinematic chain are known, the pose of the chain is uniquely determined. Let's consider the example of a kinematic chain consisting of the three connected segments A,B and C, A being the root. Let the point $\overline{\mathbf{e_C}} = (x_C, y_C, z_C, 1)^T$ be defined in the local frame of segment C. Then its coordinates $\overline{\mathbf{e_A}} = (x_A, y_A, z_A, 1)^T$ with respect to the frame attached to segment A evaluate to

$$\overline{\mathbf{e_A}} = \mathbf{P_{AB}}\mathbf{P_{BC}}\overline{\mathbf{e_C}} \tag{2.2}$$

$\mathbf{P_{AB}}$ is the relative rigid body transformation between segments A and B, and $\mathbf{P_{BC}}$ is the relative rigid body transformation between segments B and C.

In a kinematic skeleton the translational components of rigid body transformations are implicitly represented by the bone lengths. The joints model the rotational component. Since the bone lengths are constant, the pose of the skeleton is fully-specified by the rotation parameters for each joint. Only for the root the translation has to be set.

An element of $SO(3)$ has at most three degrees of freedom, and thus there are more compact ways to specify rotations than via the full matrix. The three most widely-used rotation parameterizations are described in the following. They are also applied in later chapters of this thesis:

Euler angles Here, the idea is to parameterize the transformation as a product of three rotations around specific coordinate axes. Most widely used are the ZYZ-Euler angles in which the matrix $\mathbf{R}(\alpha,\beta,\gamma)$ is a product of a rotation around the z-axis $\mathbf{R}_z(\alpha)$ by an angle α, a rotation around the transformed y-axis $\mathbf{R}_y(\beta)$ by an angle β, and a rotation around the transformed z-axis $\mathbf{R}_z(\gamma)$ again by an angle γ:

$$\mathbf{R}(\alpha,\beta,\gamma) = \begin{bmatrix} cos\alpha & -sin\alpha & 0 \\ sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \begin{bmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Euler angles are a local parameterization of $SO(3)$ and thus singularities (commonly referred to as gimbal lock) can occur. Different sequences of rotation axes are also feasible [Murray94].

Quaternions Quaternions give a global parameterization of $SO(3)$. A quaternion is a generalization of complex numbers and represented as a vector quantity of the form

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \quad q_i \in \mathbb{R}, i = 0,\ldots,3 \quad (2.4)$$

where q_0 is the scalar component and $\vec{q} = (q_1,q_2,q_3)$ is the vector component. A convenient shorthand notation is $\mathbf{q} = (q_0,\vec{q})$. The set of quaternions is a 4-dimensional vector space over the reals and forms a group with respect to quaternion multiplication [Murray94]. Any rotation around a unit axis $\vec{\omega} = (\omega_x,\omega_y,\omega_z)$ by angle θ can be represented by a unit quaternion of the form

$$\mathbf{q} = (cos(\theta/2, \vec{\omega} sin(\theta/2)) \quad (2.5)$$

Combined rotations can be compactly expressed by quaternion multiplication.

Axis-angle An element of $SO(3)$ can be parameterized via a unit rotation axis $\vec{\omega} = (\omega_x,\omega_y,\omega_z)$ and an angle θ by which to rotate around this axis. The corresponding rotation matrix is obtained via Rodriguez' Formula as:

$$\mathbf{R} = \mathbf{I} + sin(\theta)\hat{\mathbf{W}} + (1 - cos(\theta))\hat{\mathbf{W}}^2 \quad \text{with} \quad \hat{\mathbf{W}} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (2.6)$$

Inferring the axis and angle parameters from the matrix R is a bit more involved and described in detail in [Murray94].

It depends on the specific application which of the parameterizations is most appropriate. They differ in mathematical properties, modeling power (local,

global), memory consumption, and computational efficiency (see [Eberly02] for an instructive comparative evaluation). Not all joints provide all three degrees of freedom. Sometimes, such as in a hinge, only one degree of freedom is provided. These constraints can be transformed into appropriate numerical constraints in the parameter space.

In the course of this thesis we will develop several algorithmic solutions to the problem of inferring transformation parameters for the human body from image data. In the context of each of these methods we will describe which rotation parameterization was found to be appropriate.

2.1.2 Modeling the Appearance of the Human Body

There are two main elements that contribute to the realistic appearance of a virtual human, the geometry of the body and the texture of the surface. The surface geometry of the body is typically modeled by means of a triangle mesh. The vertices of the mesh are attached to the bones such that the moving skeleton moves the body surface accordingly. There are single-skin and segmented surface representations.

In a segmented model, each body part is represented by a separate triangle mesh. Each vertex is assigned to exactly one bone. The body model shown Fig. 2.1b belongs to this category.

In a single-skin model, vertices that are in the spatial neighborhood of a joint are weightedly assigned to either of the two adjacent bones. This way, skin deformations due to joint bending can be represented, a technique commonly referred to as vertex skinning [Fernando04].

One can even take one step further and model the skin deformations due to the activity of the muscles in the human body [Kähler03].

The second component contributing to a realistic look of a virtual human is the surface texture. One way of reproducing the appearance of a real person is to reconstruct a consistent surface texture from photographs. A static texture, however, cannot reproduce details, such as wrinkles, that change with the body pose.

A dynamic surface texture that incorporates such time-varying details can also be reconstructed from photographs if for each pose that the model strikes multiple images are available (Chap. 8).

Even a dynamic surface texture can only faithfully reproduce the look of a person under fixed illumination conditions. If one wants to render a person captured in the real world under arbitrary novel lighting conditions, a mathematical description for the surface reflectance has to be derived (Chap. 10).

In the most general case, surface appearance must be phenomenologically described by a twelve-dimensional function [Rusinkiewicz00]. Typically, how-

ever, phosphorescence and fluorescence effects as well as subsurface scattering can be ignored, which significantly reduces reflectance representation dimensionality. In most cases, a six-dimensional function suffices, known as the spatially-varying *bidirectional reflectance distribution function* (BRDF) f_r. It is defined at all surface points $\vec{x}$ as the ratio of outgoing radiance L_o in hemispherical direction $\hat{v} = (\omega_o, \theta_o)$ to incoming irradiance $L_i \cos\theta_i \, d\omega_i$ arriving from direction $\hat{l} = (\omega_i, \theta_i)$:

$$f_r(\hat{v}, \vec{x}, \hat{l}) = \frac{dL_o(\vec{x}, \hat{v})}{L_i(\vec{x}, \hat{l}) \cos\theta_i \, d\omega_i} \tag{2.7}$$

While in its general form the BRDF describes any surface reflectance characteristics, in computer graphics, real-world BRDFs are regularly represented using parametric models that consist of diffuse object albedo and an analytical expression for the specular/glossy reflection component. By varying parameter values, parametric BRDF models can represent a wide range of different reflectance characteristics with the same mathematical expression.

Two parametric BRDF models will play a major role in our project on relightable free-viewpoint video reconstruction (Chap. 10), the Phong model and the Lafortune model. The empirical Phong model [Phong75] is an isotropic reflectance model that consists of diffuse object color and a specular lobe

$$f_r^{rgb}(\hat{l}, \hat{v}, \vec{x}, \rho) = k_d^{rgb} + \frac{k_s^{rgb}}{\hat{n} \cdot \hat{l}} (\vec{r}(\hat{l}) \cdot \hat{v})^{k_e} \tag{2.8}$$

Given the surface normal $\hat{n}$, the reflection vector is defined as $\vec{r}(\hat{l}) = \hat{l} - 2(\hat{l} \cdot \hat{n})\hat{n}$. For diffuse and specular color, we have to consider the red, green, and blue color channel separately. Seven model parameters $(k_d^{rgb}, k_s^{rgb}, k_e)$ then describe diffuse object color, specular color, and the Phong exponent which controls the size of the specular lobe.

The Lafortune model [Lafortune97] is an extension of the Phong model. It can additionally incorporate off-axis specular peaks, backscattering and even anisotropy:

$$\begin{aligned} f_r^{rgb}(\hat{l}, \hat{v}, \vec{x}, \rho) &= k_d^{rgb} \\ &+ \sum_i [C_{x,i}^{rgb}(l_x v_x) + C_{y,i}^{rgb}(l_y v_y) + C_{z,i}^{rgb}(l_z v_z)]^{k_{e,i}} \end{aligned} \tag{2.9}$$

Besides diffuse color k_d^{rgb}, the model includes several specular lobes i whose individual direction, specular albedo and directedness are defined by $(C_{x,i}^{rgb}, C_{y,i}^{rgb}, C_{z,i}^{rgb}, k_{e,i})$. The vectors $\vec{l} = (l_x, l_y, l_z)$ and $\vec{v} = (v_x, v_y, v_z)$ are the normalized vectors corresponding to the hemispherical directions $\hat{l}$ and $\hat{v}$. We refer the interested reader to [Lensch04] for a more detailed elaboration on reflectance models.

2.2 The Camera and its Mathematical Equivalent

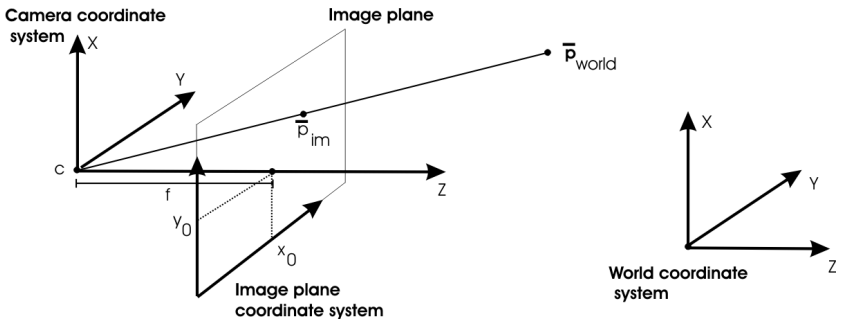


Figure 2.2: Illustration of the mathematical camera model that simulates the imaging process of a real-world CCD camera.

A camera captures an impression of a 3D scene in the 2D image plane. A lens collects the incident illumination and deflects light rays towards a focal point. The deflected rays finally form an image of the observed scene in the image plane. In analogue cameras a photographic material is employed to capture the image. In a digital camera an array of photosensitive cells assembled in a CCD chip serve the same purpose [Janesick01]. In order to incorporate the process of image formation into an algorithmic framework, a mathematical description for the mapping between 3D world space and 2D image space is required.

2.2.1 A Mathematical Model of a CCD Camera

The image formation process of a CCD camera is modeled by means of a pinhole camera model, which is mathematically described by a projective linear transformation [Hartley00]. Both the photo and video cameras employed in our research feature a CCD imaging sensor. Let $\overline{\mathbf{p}}_{world} = (p_x, p_y, p_z, 1)^T$ be a point that is specified in the world coordinate frame. Then its projected location in the image plane $\overline{p}_{im}$ of the camera evaluates to:

$$\overline{p}_{im} = \mathbf{KO}\overline{\mathbf{p}}_{world} = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & -\mathbf{Rc} \\ 0 & 1 \end{bmatrix} \overline{\mathbf{p}}_{world} \tag{2.10}$$

$\mathbf{R}$ is the 3×3 rotation matrix that represents the orientation of the camera's local coordinate frame with respect to the world coordinate frame, and $\mathbf{c} \in \mathbb{R}^3$ are the Euclidean world coordinates of the camera's center of projection. The parameters

of **R** and **c** are called the *external parameters* of the camera. The matrix **K** is commonly referred to as the calibration matrix, its entries are called the *intrinsic parameters* of the camera. The principal point in the image plane, i.e. the intersection of the optical axis with the image plane, is at position (x_0, y_0). The coefficients $\alpha_x = fm_x$ and $\alpha_y = fm_y$ represent the focal length of the camera in terms of pixel dimensions in x and y direction respectively.f is the focal length of the camera, and m_x and m_y are the numbers of pixels per unit distance in image coordinates in x and y direction respectively. Thus, a CCD camera model has 10 degrees of freedom.

The physical properties of lenses make the image formation process geometrically deviate from the ideal pinhole model. Geometric deviations typically arise in the form of radial or tangential image distortion artifacts [Jain95].

Radial distortion originates from the fact that a physical lens bents light rays towards the optical center by more or less than the ideal amount. It's effect in the image plane can be modeled by a polynomial in the radial distance from the image plane center.

Most off-the-shelf camera lenses are actually composed of several individual lenses. Tangential distortion effects are due to the fact that the individual lenses in an optical system of a camera do not properly align with respect to the overall optical axis [Weng90].

2.2.2 Camera Calibration

In order to simulate the properties of a real camera, one needs to determine the parameters of mathematical models that optimally reflect the geometric and photometric imaging properties of the real device. This process is termed *calibration*.

The most important calibration step is *geometric calibration* in which the parameters of the imaging model detailed in Sect. 2.2.1 are estimated. Most calibration algorithms proposed in the literature [Tsai86, Heikkila96, Jain95] derive the camera parameters from images of a calibration object with known physical dimensions, such as a checkerboard pattern. An optimization method modifies the model until the predicted appearance of the calibration object optimally aligns with the captured images. In order to mimic the imaging properties of a physical camera in a rendering library like OpenGL one needs to transform the calibrated camera model into the mathematical camera framework applied by this library. This conversion is applied in most of the projects in this thesis and, for the OpenGL system, it is described in detail in [Li01].

If reconstruction from images is the goal, not only the geometric imaging properties but also the photometric imaging properties of the imaging sensors have to be calibrated. Most cameras don't establish a linear relationship between intensity values in the captured scene and pixel values in the image. A response

curve of the camera can be estimated vie *photometric calibration* that enables us to establish such a linear relationship in a post-processing step.

Furthermore, the tristimulus color values (e.g. RGB) recorded for a color patch in the scene depend not only on the spectral reflectance of the patch, but also on the spectrum of illumination and on the spectral response of the imaging sensor. To ensure correct color acquisition under a given illumination setup, a *color calibration* step has to be performed. The simplest color calibration procedure is white balancing. White balancing computes multiplicative scaling factors from an image of a purely white or gray object. A more detailed elaboration on photometric and color calibration can be found in [Goesele04].

2.2.3 Camera Pairs

A pair of cameras whose viewing directions converge is commonly referred to as a *stereo pair*. Stereo images of a scene can be used to derive 3D structural information. If a stereo pair is fully-calibrated, i.e. the intrinsic and extrinsic parameters for both cameras are known, the metric 3D position of a point $\mathbf{p}$ visible in both cameras can be calculated via a procedure called *triangulation* (Fig. 2.3a). The position $\mathbf{p}$ is estimated by computing the intersection point of two rays, $r1$ and $r2$. The ray $r1$ originates in the center of projection of camera 1, $c1$, and penetrates the image plane in the position $p1$ to which the 3D point projects. Ray $r2$ is constructed in the same way for camera 2. Due to measurement noise the rays will most certainly not truly intersect, and thus it is common practice to approximate the 3D position of a point by the point that has the smallest distances to both rays.

The image formation process in a stereo pair of cameras is described by its *epipolar geometry* (Fig. 2.3). It describes the fact that an image point $p1$ in one camera view has a corresponding point $p2$ in the other camera view which lies somewhere on a line $e2$ in the other image, the so-called *epipolar line*. The epipolar geometry of a stereo pair is fully-specified by its *fundamental matrix*. Given this matrix, the epipolar line $e2$ in camera 2 that corresponds to point $p1$ in camera 1 can be directly computed via simple matrix multiplication. This way, the correspondence finding problem reduces to a one-dimensional search problem along a line. In a fully-calibrated camera pair, the fundamental matrix is directly available. However, it can also be inferred from 8 point correspondences between two uncalibrated cameras. The concept of epipolar geometry and the derivation of the fundamental matrix are detailed in [Faugeras93, Hartley00].

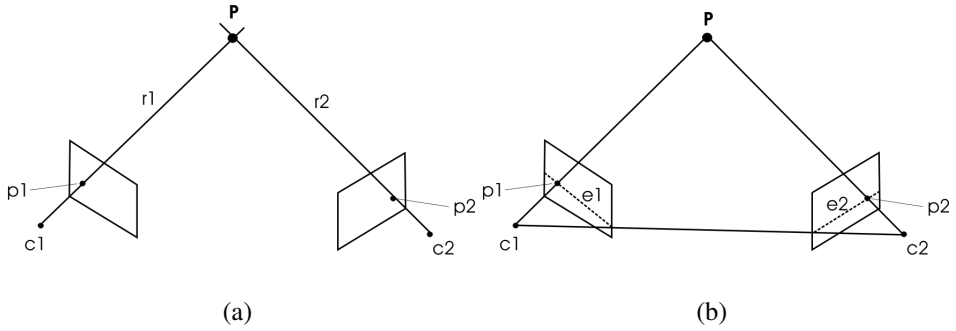


Figure 2.3: (a) Triangulation: The point of intersection of the two rays $r1$ and $r2$ through the respective cameras' centers of projection $c1$ and $c2$ and the respective projected image plane positions $p1$ and $p2$defines the 3D position p of the point. (b) Epipolar geometry: The point $p2$ in camera 2 that corresponds to point $p1$ in camera 1 must lie on an epipolar line $e1$. The inverse relation with flipped indices also holds.

2.3 Important Image Processing Algorithms

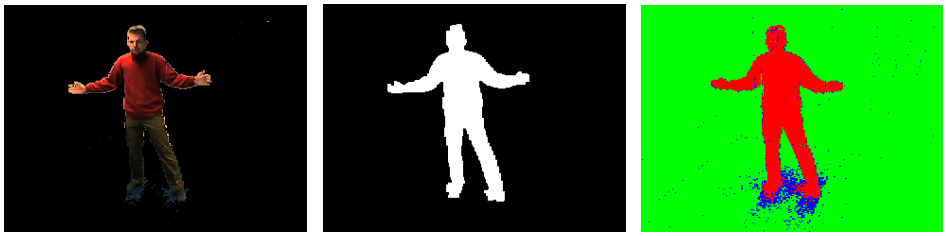
2.3.1 Background Subtraction

In all the research projects detailed in this thesis we pre-process the input image and video data such that a person or an object in the scene foreground is segmented from the scene background. We have decided to use a color-based method originally proposed in [Cheung00]. This approach incorporates an additional criterion which prevents shadows from being erroneously classified as part of the scene foreground. Our subtraction method employs per-pixel color statistics for each background pixel that is represented by a mean image $\Pi = \{\vec{\mu}(x,y) \mid 0 \leq x < \text{width}, 0 \leq y < \text{height}\}$ and a standard-deviation image $\Sigma = \{\vec{\sigma}(x,y) \mid 0 \leq x < \text{width}, 0 \leq y < \text{height}\}$, each pixel value being a 3-vector comprising all three color channels. In order to incorporate the natural variations in pixel intensity due to noise and natural illumination changes into these statistics, they are generated from several consecutive video frames of the background scene without an object in the foreground.

Background subtraction on a novel frame classifies an image pixel $\vec{p}(p_x, p_y)$ at position (p_x, p_y) as follows. If the color of $\vec{p}(p_x, p_y)$ differs in at least one RGB channel by more than an upper threshold T_u from the background distribution

$$\mid \vec{p}(p_x,p_y)_c - \mu(p_x,p_y)_c \mid > T_u \cdot \vec{\sigma}(p_x,p_y)_c \quad , c \in \{r,g,b\} \tag{2.11}$$

it is classified as foreground. If its difference from the background statistics is smaller than the lower threshold T_l in all channels, it is certainly a background



(a) (b) (c)

Figure 2.4: Video frame after background subtraction (a) and the corresponding silhouette (b). Shadows cast by the person onto the floor (blue) are identified and correctly classified as scene background (c).

pixel. All pixels which fall in between these thresholds are possibly in shadow areas. Shadow pixels are classified by a large change in intensity but only a small change in hue. If $\vec{p}(p_x, p_y)$ is the color vector of the pixel to be classified, and $\vec{\mu}(p_x, p_y)$ is the corresponding background pixel mean color vector, their difference in hue is

$$\Delta = \cos^{-1}\left(\frac{\vec{p}(p_x, p_y) \cdot \vec{\mu}(p_x, p_y)}{\|\vec{p}(p_x, p_y)\| \|\vec{\mu}(p_x, p_y)\|}\right) \tag{2.12}$$

If $\Delta > T_{angular}$ the pixel is classified as foreground, else as shadow. Finally, a 0/1-silhouette mask image for the video frame is computed (Fig. 2.4).

2.3.2 Optical Flow

The optical flow is the projection of the 3D velocity field of a moving scene into the 2D image plane of a recording camera. The determination of the 2D optical flow from spatio-temporal intensity variations in images has been investigated in Computer Vision for many years [Barron94].

A number of simplifying assumptions are typically made to compute the optical flow from the pixel intensities of two subsequent images. First, it is assumed that the change in image intensity is due to translation in the image plane only (intensity constancy constraint)

$$I(u,t) = I(u - \vec{o}t, 0) \tag{2.13}$$

where $\vec{o} = (p,q)^T$ is the optical flow at image point $u = (u,v)^T$, I being the image intensity at coordinates u and time t. From the Taylor expansion of (2.13) and linearization, the *optical flow constraint equation* is derived as

$$\nabla I(u,t) \cdot \vec{o} + I_t(u,t) = 0 \tag{2.14}$$

where $I_t(u,t)$ is the temporal derivative of the image intensity. This is an equation in two unknowns which cannot be solved at a single image plane position without additional constraints. Hence, it is common practice to make additional assumptions about the smoothness of the optical flow field in a local spatial neighborhood to make the problem well-posed.

In the optical flow approach by Lucas and Kanade [Lucas81], a weighted least-squares fit to the local first-order constraints (2.14) is computed by minimizing the functional

$$\sum_{u \in W} W^2(u)[\nabla I(u,t) \cdot \vec{o} + I_t(u,t)]^2 \tag{2.15}$$

where $W(u)$ defines a Gaussian neighborhood around the current position in the image plane for which the optical flow is computed. It is also feasible to employ a hierarchical variant of the Lucas-Kanade approach that incorporates flow estimates from multiple levels of an image pyramid into its final result. In Chap. 9 we employ this method to compute optical flows from which 3D motion fields for body pose update are reconstructed. In Chap. 10 the algorithm is used as a component of an image-based warp-correction scheme.

Part I

Marker-free Optical Human Motion Analysis

Chapter 3

Problem Statement and Preliminaries

Video-based analysis of motion has always been a problem that attracted researchers from computer vision and computer graphics. Amongst the most important types of motion is the motion of humans. Video-based methods that extract mathematical models of human motion are of great relevance in many application scenarios:

The generation of life-like human characters is an important issue in the production of today's computer games and motion pictures. In order for a virtual human to be convincing, not only its visual appearance but also its movements have to comply with the real world equivalent. The eye of a human observer has been trained to notice even the slightest unnaturalness in gait. A motion analysis approach enables capturing all the fine details of human movements from real persons.

Researchers in the field of biomechanics analyze the interplay of the human bone and muscle system while the body is moving [Whittle96]. Thus they have a strong interest in detailed models of human motion that were captured from real world test subjects. Biomechanical motion analysis can also be a great help for coaches in many sports disciplines. The analysis enables a much more detailed impression of which parts of an athlete's course of motion can be improved [Calvert94].

Computer-based analysis of human motion also enables the automatic interpretation of human gestures. It has for long been a goal of Artificial Intelligence to create optical user interfaces that enables software systems to appropriately react to a user's behavior [Pavlovic97, Starner98, Malassiotis02].

The advent of ever more powerful computing and display hardware has paved the trail for new visual media applications. The enormous amount of data that

arises when these media are to be transmitted to the end-user make necessary efficient encoding schemes. Therefore, a trend in the picture coding community can be observed to employ motion information also for the purpose of data reduction. Since many video sequences are centered around human actors, model-based encoding schemes that transmit a 3D model of the person and its motion parameters instead of the full video stream can help to significantly reduce the required bandwidth [Eisert01, Grammalidis01, Weik99]. Hence, the latest video standard by the ISO/OSI Motion Pictures Expert Group, MPEG-4, also provides an algorithmic framework to encode video objects based on their motion parameters [Capin99, ISO/IEC00].

The term human motion analysis denotes a superordinate concept which subsumes many algorithmic subproblems that range from the actual estimation of motion parameters to the interpretation of motion on a semantic level. In our work we focus on two fundamental algorithmic challenges which are at the core of human motion analysis, namely human motion capture and body model estimation:

- **Human Motion Capture**
  Human motion capture is the process of estimating a mathematical description that completely describes a sequence of motions that is performed by a person in the real world. This mathematical representation has two components. The first component is a theoretical model of the person's body structure and kinematic properties. The second component is a set of parameters that describe the subject's motion in terms of this body representation (see Sect. 2.1). The task of a motion capture algorithm is to estimate these parameters of motion. The derivation of an appropriate body model is a separate problem.

- **Body Model Estimation**
  Body Model Estimation is the process of automatically deriving a body representation that models the shape and kinematic properties of a human actor.

A variety of different approaches have been described in the literature which search for answers to these two algorithmic questions. They mainly differ in the physical principle that is used to collect data of a moving subject. Mechanical, electromagnetic, and sonar tracking devices have been developed, but by far the most widely used systems employ image or video data. Unfortunately, many of these approaches require some form of physical interaction with the scene, for instance in the form of an exoskeleton, tracking sensors or optical beacons [Menache95]. However, in many application scenarios any form of interference with the scene in order to estimate motion or skeleton information is totally

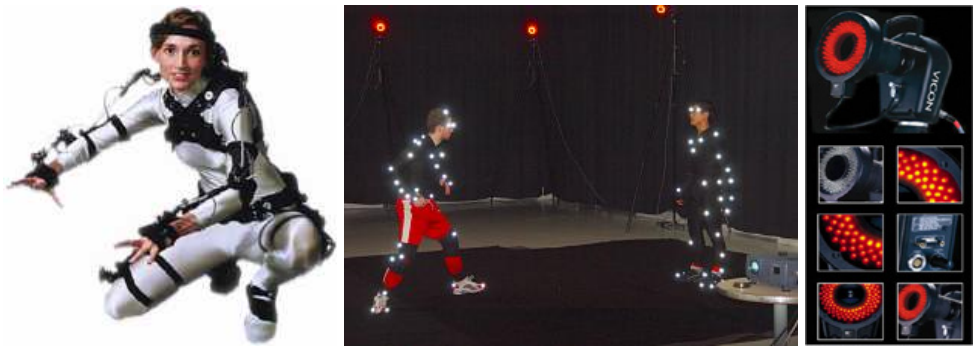
inappropriate. Optical surveillance of humans is only feasible if it can be done without having to physically interact with the subject in a scene. Furthermore, if one intends to not only estimate motion data but also appearance-related information, such as texture data, any form of visual modification of the scene would be obstructive. It is thus of great importance to develop methods that enable motion estimation from raw video dat and that do not require optical markings on the body.

In Part I of this thesis we describe our novel algorithms for marker-free motion parameter estimation and marker-free body model reconstruction from multi-view video. In Chap. 5 a novel hybrid approach for marker-free capture of human motion parameters from multiple synchronized video streams is detailed. It joins the forces of a real-time feature tracking and a real-time volume reconstruction method to fit a multi-layer kinematic skeleton to the motion data at interactive frame rates. In Chap. 6 we describe the nuts and bolts of a method which enables the automatic estimation of a kinematic body model from multiple video streams of a moving person. The approach does with a minimum of a priori information about the body structure of the recorded subject and is applicable to arbitrary moving subjects, including humans, animals and mechanical devices. The synchronized video streams of a moving subject that we need as inputs to both algorithms are recorded in our multi-view video acquisition studio (Chap. 4). The recording facility is not only a fundamental component of our research on human motion analysis but also an important building block in our work on free-viewpoint video and reflectance estimation which is detailed in Part II of the thesis. Although the silhouette-based motion capture algorithm developed there conceptually belongs to Part I of this thesis, we prefer to describe it in the scope of the overall application it was designed for (Chaps. 8,9, and 10).

The remainder of this chapter will briefly review and categorize algorithms and systems from the literature that attack the problems of human motion capture and body model estimation

3.1 Background

In the following subsections we will have a quick look at important related work from the literature. While there are a variety of conceptually different approaches for human motion capture, only very few methods for automatic body model estimation have been developed so far. Although the latter methods form a separate algorithmic category, they are frequently presented in conjunction with a motion estimation algorithm. We begin our review with an explanation of important technical categories of human motion capture methods (Sect. 3.1.1 to Sect. 3.1.3). Thereafter, we briefly discuss algorithms for automatic optical estimation of body



(a) (b) (c)

Figure 3.1: (a) Metamotion^{TM} Gypsy exoskeleton with resistive bending angle sensors. (b) Marker-based optical motion capture system from Vicon. (c) Vicon motion capture camera with ring of LED light sources.

models. We conclude this chapter with a brief look at photo and video camera-based image data acquisition facilities and compare them to the studio that we have built (Sect. 3.1.5.

The following overview is not intended to be a complete review but shall point out exemplary work. The interested reader who wants to explore the field more deeply is referred to the review papers on the field [Gavrila99, Moeslund01, Aggarwal99], as well as the paper by Gleicher et al. which elaborates on key challenges in video-based motion capture [Gleicher02].

3.1.1 Non-optical Human Motion Estimation

In recent years, many different motion tracking systems have been developed that employ quite different non-optical physical principles [Menache95, Moeslund00]. One category are the so-called inside-in-systems, in which the person is required to wear a specially designed suite. This special type of apparel is equipped with fibre-optic or resistance sensors that measure the joints' bending angles. An example of a tracking suite which is equipped with aluminum rods and potentiometers is the Gypsy exoskeleton from Metamotion^{TM} [Met] that is shown in Fig. 3.1a.

Electromagnetic body tracking systems employ multiple small sensors and an external electromagnetic field generator. For human motion capture the sensors are placed at several positions on the body and each sensor's position and orientation is derived from its electromagnetic interaction with the external field [Pol].

Acoustic tracking systems either measure the time of flight of sound pulses

sent between a sender and receiver station, or analyze the phase shift between sent and received signal. Each tracking sensor typically provides position and orientation data. The accuracy of acoustic tracking systems is usually very limited and they are normally not used for tracking the whole body. An example for this category of devices is the Red Baron Ultrasonic Head Tracker by Logitech [Log].

3.1.2 Video-based Motion Estimation using Optical Markers

By far the most widely used commercial systems for human motion capture are marker-based optical acquisition setups. They make use of the principle of moving light displays [Johannson73]. Optical markings, which are either made of a retroreflective material or LEDs are placed on the body of a tracked subject. Several special-purpose high-frame-rate cameras (often with specialized light sources) are used to record the moving person. The locations of the markers in the video streams are tracked and their 3D trajectories over time are reconstructed by means of optical triangulation [Gleicher99, Herda00]. The main algorithmic problems that have to be solved are the unambiguous optical tracking of the markers over time as well es the establishment of marker correspondences across multiple camera views [Ringer02]. Before motion capture commences, the person stands in a so-called *t-pose*, i.e. upright body with arms spread to the sides. From this pose, correspondence between marker positions on the body and individual segments of a hierarchical kinematic skeleton model are established. These correspondences and the marker trajectories enable the estimation of each segment's position and orientation at each time step of a motion sequence. The segment-specific tracking data can be transformed into joint rotation parameters of the skeleton model. Due to self-occlusions it happens frequently that markers on the body appear and disappear in some camera views and wrong across-camera correspondences are established. Thus a post-processing step is always necessary in which the motion data are smoothed and problems due to marker occlusion are manually resolved. In Fig. 3.1b a marker-based optical motion capture system manufactured by Vicon is [Vic] is depicted. Fig. 3.1c shows the employed special-purpose camera with a ring of LEDs. The markers on the body optimally reflect the light emitted from these light sources.

In principle, marker-based systems can not only be used to track humans but also animals and vehicles. The accuracy at which real-world positions and orientations are tracked can reach sub-millimeter level. However, all these advantages come at the cost of having to visually modify a recorded scene completely. In consequence, it is impossible to derive appearance information, such as textures or reflectance samples, from the captured video material.

3.1.3 Marker-free Optical Motion Estimation

Marker-free optical estimation of human motion is a highly complex problem. The problem of inferring the pose of a human from pure image data is, in principle, a search problem in the parameter space of the employed model. There are several factors which make this problem hard.

First, the human body is a very complex articulated system. Even the simplest kinematic skeleton models employed already offer more than 30 degrees of freedom. This fact renders the exhaustive search for a correct body pose completely illusive. It is thus inevitable to incorporate constraints into the search process. Many different types of constraints have been considered. The most popular ones are features in the image, such as edges [Drummond01] or silhouette information [Carranza03], that correspond to certain features of the body model. It is also feasible to employ dynamic models that mathematically describe human motion and that allow for the prediction of the next body pose from the current one. Tracking robustness is increased if the dynamic model is integrated into a stochastic tracking framework [Deutscher00, Sidenbladh00, Sidenbladh02]. Kinematic and dynamic constraints of the real human body can also help to restrict the search range [Herda04].

A second factor that contributes to the algorithmic complexity of marker-free optical motion estimation is the fact that the mapping from the 3D space of the human body to the 2D space of the image domain is not one-to-one. Many different body configurations can theoretically result in the same image.

Existing approaches from the literature that attack the problem of non-intrusive optical pose estimation can be categorized according to many algorithmic criteria. We have decided to distinguish between methods that only extract 2D motion information, and algorithms that recover the full complexity of human motion in 3D. In the following, we describe what strategies are followed in the different approaches in order to handle the problem's complexity.

Estimation in 2D

One general approach to the analysis of human motion has been to bypass a model-based pose recovery step altogether and to describe movements in terms of simple 2D features in the image plane. The computational complexity of these methods is often significantly lower than for approaches that employ a full 3D body representation. Thus, they have been very popular until a couple of years ago when the performance of commercially available computer hardware was still a major bottleneck.

2D algorithms have been widely used for hand tracking and gesture represen-

tation. The features employed are either shape information or movement/location data within the image plane. Typical shape features are x-y shape images and orientation histograms [Freeman96] or Zernike moments [Hunter95]. Trajectory parameters of region centroids are also often used [Starner97, Davis93].

It is also possible to subdivide the image plane into a regular grid, and to use features assigned to each tile to derive specific motion information, e.g. motion periodicity. Typical tile features are the sum of normal flow [Nelson94] or the pixel values themselves [Kjeldsen96, Darrell93].

Another line of search involves statistical shape models. Cootes et al. [Cootes95] employ what they call Active Shape Models. These models are learned in a training stage from example shapes that are described by known feature point locations. A principal component analysis on feature locations is applied to describe example shapes in a low parameter space. The learned model can be used to track deformable objects such as hands (or humans). Baumberg and Hogg [Baumberg94] employ Active Shape Models based on B-splines for tracking the motion of pedestrians.

One step further take all those approaches which make use of some form of a priori structural and kinematic information about the human body. Typically, an explicit 2D shape model is used which is fitted to the observed motion by identifying features in the image plane. The type of model strongly influences what type of features can be used. One can distinguish methods using edge or ribbon features, image regions (or blobs) or simple points. In the approach by Geurtz [Geurtz93] body poses are inferred by fitting 2D ellipsoids to image data by means of a hierarchical curve fitting scheme. Human silhouettes in space-time volumes of video streams are identified and tracked with deformable contour models in the method by Niyogi and Adelson [Niyogi94]. In [Guo94] human motion is tracked in 2D by fitting a stick figure to the skeleton of the person's silhouette.

Ribbon features corresponding to the arms and feet are identified in the approach presented in [Chang96]. Leung and Young [Leung95] use a 2D model that consists of many different elements such as five U-shaped ribbons for the extremities, a trunk, and various joints and midpoints. Moving edges are detected in the silhouettes of a moving person and the model is fitted to these features.

The Pfinder system [Wren97] takes a region-based approach. Each image region, also called blob, is described in statistical terms by Gaussian distributions in position and color space. Each blob corresponds to one specific part of the human body, such as the hand or the head. The actual tracking of the person iterates between the prediction of the appearance of the blobs in the next frame, an assignment of pixels to individual regions, and an update of the blob statistics.

It is also feasible to use point features along the medial axis of a person's

outline for tracking as it is shown in [Cai96]. A simple body model consisting of a head and a trunk representation is fitted to the human motion by analyzing the position and velocity of the point features.

In the W^4 system by Haritaoglu et al. [Haritaoglu98] a frame differencing method is used to identify moving persons in monocular video sequences. Several persons can be tracked at the same time. The motion of individual body parts can also be followed. For this purpose a 2D card board body model is employed whose individual components are fitted to the respective moving sub-regions in the image plane.

Example-based approaches estimate the pose of a person by comparing an image to a database of reference pose images for which the model parameters are known [Sullivan02, Carlsson00].

Appearance-based methods build a body representation of the human from the image data and use this for tracking it over time [Ramanan03]. Typically, the human body is modeled as 2D articulated puppet whose limbs are represented by little rectangular areas that are filled with texture from the actual image data (Fig. 3.2a).

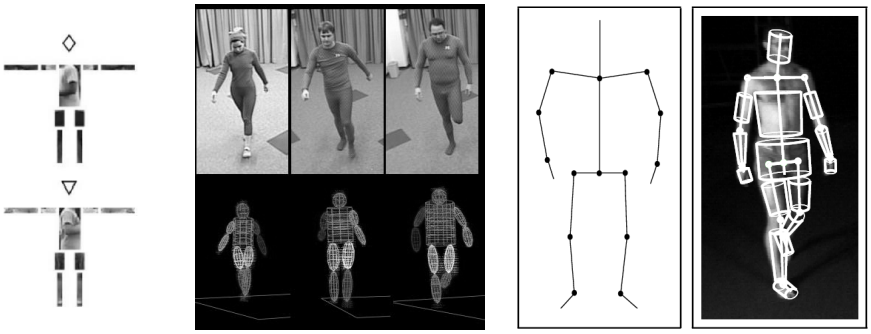
The approaches described in this section can at best extract a certain subset of information about the full human motion. Thus, they are often used as components of gesture recognition or surveillance systems for which it is sufficient to work with a coarser motion representation. However, if the movements of a person have to be captured in their full complexity more sophisticated approaches that employ detailed 3D models are needed.

Estimation in 3D

The most detailed understanding of human movement is obtained through capturing algorithms that estimate the motion parameters of a complete 3D kinematic body model. The employed body model typically consists of a linked kinematic chain of bones and interconnecting joints (Sect. 2.1). The skeleton is fleshed out with simple geometric primitives in order to model the physical outline of the human body. Commonly employed types of shape primitives are ellipsoids [Cheung00, Mikić01], superquadrics [Sminchisescu03, Gavrila96, Kakadiaris96], and cylinders [Sidenbladh00, Goncalves95]. A more sophisticated body model that employs an implicit surface representation generated from a collection of metaballs is presented in [Plaenkers03].

A multitude of different strategies to bring such a 3D body model into optimal accordance with the pose of the human in one or multiple video streams has been investigated.

One possibility is to use a divide and conquer strategy where the motion of each individual body part is tracked separately and mathematical constraints en-



(a) (b) (c)

Figure 3.2: (a) Example 2D models applied in the appearance-based approach by Ramananan et al. (taken from [Ramanan03]). (b) Body model employed by Deutscher et al. in their particle-filter-based 3D approach (taken from [Deutscher00]). (c) Tracking results with the visual-hull-based approach by Mikić et al. (taken from [Mikić01]).

sure the connectivity of the body model on a global level. A very early example of this principle is the approach presented in [Shakunaga91] where the angles between projected sub-parts are identified to solve the pose recovery problem and to enforce the connectivity. In another method [Mittal03] silhouette data are identified in multiple video streams that show the same scene from different camera views. The silhouette of a person in each view is automatically subdivided, individual parts are identified, and each part is separately tracked. Connectivity is enforced in a separate step.

In the pioneering work by O'Rourke and Badler [O'Rourke80] a constraint propagation principle is used to narrow the pose parameter search space. An explicit 3D kinematic model is employed. In the first frame, box-shaped regions in the image plane that correspond to projected joint locations are marked. A constraint propagation scheme is applied based on known distances between the joints in the body model. New body poses are recovered through an iterative refinement. Constraint propagation methods are also employed in other work, e.g. [Chen92].

In [O'Rourke80] a general architectural framework for human motion tracking systems has been proposed which is still used in many marker-free capturing methods. According to this principle, model-based tracking consist of a prediction phase, a synthesis phase, an image analysis phase, and a state estimation phase. In other words, at each time step of a motion sequence the capturing system first makes a prediction of the current pose, then synthesizes a view with the model

in that pose, compares the synthesized view to the actual image, and updates the prediction according to this comparison. Different tracking systems differ in what algorithmic strategy they employ at each stage.

Analysis-through-synthesis methods search the space of possible body configurations by synthesizing model poses and comparing them to features in the image plane. The misalignment between these features and the corresponding features of the projected model drives a pose refinement process [Grammalidis01, Koch93, Martinez95]. In [Plaenkers03] a sophisticated body model consisting of a skeleton and a surface representation based on a collection of metaballs is employed. It is fitted to the motion of a person by aligning it with silhouette and depth data that is reconstructed from multiple camera views.

Physics-based approaches compute forces acting on the model which bring it into optimal accordance with the video footage [Kakadiaris95]. In the algorithm proposed in [Delamarre99] silhouette images of a person in multiple video streams are generated. Forces are computed that are proportional to the misalignment between the outer contours of the projected model and silhouetted in the video footage. In conjunction, all partial forces move the model into the correct stance.

Another category of approaches tries to invert the non-linear measurement equation that maps the state space of the model onto the space of images. One way to invert that measurement equation is to apply inverse kinematics [Yonemoto00], a process known from robotics which computes a body configuration that minimizes the misalignment between projected model and image data. Inverse kinematics inverts the measurement equation by linearly approximating it. The method in [Bregler98] fits a kinematic skeleton model fleshed out with cylindrical limbs to one or several video streams of a moving person. A combination of a probabilistic region model, the twist parameterization for rotations and optical flow constraints from the image enable an iterative fitting procedure. An extension of this idea is described in [Covelle00] where, in addition to the optical flow constraints, also depth constraints from real-time depth image streams are employed.

Some researchers have experimented with comparison-based approaches for 3D pose recovery. In [Log04] the pose of a kinematic skeleton is estimated from monocular video footage. Each video frame is compared to a database of reference images of a person for which the correct body stance is known. The stance corresponding to the optimally matching reference frame is used as a first estimate of the body pose. Since the reference image and the example frame will not show the exact same body pose, an iterative refinement is applied. A conceptually similar method is explained in [Shakhnarovich03]. Here, the main algorithmic novelty is a parameter-sensitive hashing method that enables searching the database of reference frames in sub-linear time with respect to the database size.

Recently, the application of statistical filters in the context of human motion capture has become very popular. Basically, all such filters employ a process

model that describes the dynamics of the human body and a measurement model that describes how an image is formed from the body in a certain pose. The process model enables prediction of the state in the next time step and the measurement model allows for the refinement of the predication based on the actual image data. An important advantage of these tracking filters is that process and measurement noise are implicitly taken care of. If the noise is Gaussian and the model dynamics can be described by a linear model, a Kalman Filter can be used for tracking [Mikić01]. However, the dynamics of the complete human body is non-linear. A particle filter can handle such non-linear systems and enables tracking in a statistical framework based on Bayesian decision theory [Deutscher00] (see also Fig. 3.2b). At each time step a particle filter uses multiple predictions (body poses) with associated probabilities. These are refined by looking at the actual image data (the likelihood). The prior is usually quite diffuse, but the likelihood function can be very peaky. In order to decide for the right peak (i.e. the best body pose) annealing the filter can be helpful [Deutscher00]. The performance of statistical frameworks for tracking sophisticated 3D body models has been demonstrated in several research projects [Drummond01, MacCormick00, Sidenbladh00, Sidenbladh02].

In another category of approaches that have recently become popular, dynamic 3D scene models are reconstructed from multiple silhouette views and a kinematic body model is fitted to them. A system that fits an ellipsoidal model of a human to visual hull volumes in real-time is described in [Cheung00]. The employed body model is very coarse and approximates each limb of the body with only one quadric. In [Mikić01] a system for off-line tracking of a more detailed kinematic body model using visual hull models is presented (Fig. 3.2c). The method described in [Bottino01] also reconstructs scene geometry from silhouettes. They use a two-layer kinematic body model consisting of 15 segments. On the first layer, each limb is represented by its bounding ellipsoid, on the second layer the body surface is described by a closed triangle mesh. Pose recovery is performed by running an optimization in the joint parameters such that the ellipsoids optimally approximate the shape-from silhouette-volume. A refined fit is obtained with the second layer of the body model. Cheung et. al also present an approach for body tracking from visual hulls [Cheung03]. The algorithm we present in Chap. 5 improves the performance of these volume-based methods by taking into account additional information that has been obtained via 2D feature tracking.

3.1.4 Optical Estimation of Body Models

Most marker-based optical motion capture systems are delivered with a software that allows for the automatic fitting of a template kinematic body model to the person [Menache95, Herda00]. To achieve this the person has to stand in a specific initialization pose. The software automatically establishes correspon-

dences between markers on the body and virtual markers on the model. It also rescales the bone lengths such that they match the real-world counterpart. While this procedure can accurately recover the dimensions of the human skeleton it cannot derive any information about the shape of the body surface. In [Allen02] a method is described that captures the deformation of the upper body of a human by interpolating between different range scans. The body geometry is modeled as a displaced subdivision surface. A model of the body deformation in dependence on the pose parameters is obtained by the method described in [Sand03]. A skeleton model of the person is known a priori and the motion is captured with a marker based system. Body deformation is estimated from silhouette images and represented with needles that change in length and whose endpoints form the body surface.

If no optical markers are allowed, the fully-automatic estimation of a kinematic model is a lot more difficult. In most marker-free optical motion capture methods, one assumes that a body model is known beforehand. These a priori models are adapted in shape and proportion to the person under consideration, a process which often requires user interaction.

Only a few researchers have tried to estimate a body model fully-automatically without relying on a priori knowledge about the tracked subject.

In the work by Cheung et al. [Cheung03], a skeleton is estimated from a sequence of shape-from silhouette volumes of the moving person. A special sequence of moves has to be performed with each limb individually in order to make model estimation feasible. In the approach by Kakadiaris et al. [Kakadiaris95] body models are estimated from multiple video streams in which the silhouettes of the moving person have been computed. With their method too, skeleton reconstruction is only possible if a prescribed sequence of movements is followed.

We have thus decided to investigate the algorithmic ingredients of a more general method that enables the fully-automatic estimation of kinematic body models of arbitrary moving subjects from any kind of motion sequence. The result of this effort is the method described in Chap. 6.

3.1.5 Acquisition Facilities for Multi-view Image and Video Data

Today, many researchers in computer graphics and computer vision adhere to a data-driven paradigm, which means that they reconstruct scene representations from image or video data captured in the real-world. Three research areas can be identified in which acquisition of high-quality image data is essential. In our work, we research all three of these areas. Thus, we have to developed multi-

view video acquisition studio that lives up to the requirements of all of them in conjunction (Chap. 4).

One field of research in which high-quality image data are essential is image-based reflectance estimation. There, surface reflectance models of real-world objects are estimated from a series of images taken from different viewing directions and under different incident illumination conditions. To serve this purpose, different acquisition setups consisting of high-resolution still cameras and a set of flexibly arrangeable light sources have been proposed in the literature [Goesele00, Ward92]. From the image samples the BRDF (bidirectional reflectance distribution function) of a test material is determined. In our studio we provide the technological framework that enables us to extend the photo camera-based reflection measurement approach into a method for dynamic reflectometry from video data.

The second important field of research is video-based human motion capture (Sect. 3.1.2 and 3.1.3). Commercial marker-based capturing systems apply several special-purpose high-resolution video cameras that capture the scene at a high frame rate. In order to cope with the immense amount of image data, customized storage hardware is employed [Menache95].

Marker-free motion capture algorithms don't rely on optical beacons in the video footage. In [Cheung00] the volumetric visual hull [Laurentini94] of a person is reconstructed from multiple camera views, and an ellipsoidal body model is fitted to the moving subject. Video acquisition takes place in a *3D Room* that allows recording with up to 48 cameras [Kanade98]. A similar setup for motion capture using reconstructed volumes is proposed in [Luck02]. A different multi-camera system for volume reconstruction that uses a custom-made PC cluster for video processing is described in [Borovikov00]. Several similar video-based human motion capture systems exist that use multi-view camera data as input [Horprasert98, Gavrila96].

The third field of research is 3D video. Here, multi-view video streams are not solely used to capture motion data but also to reconstruct time-varying shape and appearance models of a scene. In a previous stage of their *3D Room*, Narajanan et. al. [Narayanan98] built a dome of over 50 cameras to reconstruct textured 3D models of dynamic scenes using dense stereo. To handle the huge amount of image data, the video streams are recorded on video tape first and digitized off-line. In [Matsuyama02] a multi-camera system is described to record a moving person for reconstruction of the polygonal visual hull in an off-line process. The previous paper is an extension of the original work on polygonal [Matusik01] and image-based visual hulls [Matusik00] in which a multi-camera system for real-time 3D model reconstruction is employed. A system for recording and editing of 3D videos based on the image-based visual hull algorithm is described in [Wuermlin02]. A system for acquisition of multi-view images of a person us-

ing conventional digital cameras is described in [Weik01].

Our multi-view video studio concept and design differs from the acquisition setups mentioned before in that it exhibits a higher flexibility to suit the needs of a variety of different application scenarios. By relying on off-the shelf hardware, we built an easily modifiable but still cost-effective system. Furthermore, our studio allows for fully-digital processing of the video footage from the camera to the PC.

Chapter 4

Seeing the World through Multiple Eyes - A Studio for Multi-view Video Recording

All our video-based research projects, both in motion capture and free-viewpoint video, require high-quality video footage as input. However, video data recorded from just one camera perspective do not live up to our requirements. In contrast, our research demands multi-view video (MVV) streams, i.e. video material captured from multiple frame-synchronized cameras. In order to record these data, we have designed and constructed a special-purpose acquisition facility [Theobalt03c].

Our studio is designed as a flexible and versatile recording environment that jointly meets the demands of a variety of research efforts undertaken in our group. To illustrate the diversity of requirements that we are facing, the demands which are characteristic to two of the projects presented in this thesis are briefly summarized in the following:

- **Volume-based Marker-free Motion Capture (Chap. 5)**
  For this application controlled lighting conditions are needed to enable robust background subtraction. Furthermore, sufficient network bandwidth and CPU peformance are required to allow for simultaneous volume reconstruction, data transmission and real-time motion analysis.

- **Free-Viewpoint Video (Chaps. 8, 9,and 10)**
  In Free-Viewpoint Video it is important to acquire video material at high frame rates and at a high image resolution. Since the video frames serve as input both to texture reconstruction and silhouette-based motion estimation,

the lighting conditions need to be controllable. This way, realistic estimation of surface appearance and motion parameters becomes feasible. If it is the goal to estimate surface reflectance properties as well, the camera system has to deliver frames with high resolution, as well as with a high color precision.

The requirement analysis of the different application scenarios leads us to a general design concept of the studio. Our acquisition room is intended to be a universal recording environment for different research projects in 3D video, surround vision and video-based motion analysis. Thus, flexibility and versatility with respect to camera placement, recording conditions and the way of processing (online or off-line) are important design criteria. To keep the cost as well as the administrative overhead moderate, off-the-shelf hardware is preferred over special-purpose equipment. On the performance side, the requirements to the system are challenging. The setup must be able to acquire and process video data in real-time. At the same time it also has to provide the necessary storage capacity and bandwidth for recording and saving of multi-video streams. Intermediate storage of video data on analog media, such as video tapes, is not acceptable since we want to keep the hardware and administrative costs low. Furthermore, we want to prevent a deterioration of image quality due to the conversion into digital form.

The above concept has been put into practice in the individual components of our studio, such as the layout of the room (Sect. 4.1), the camera system (Sect. 4.2), the lighting equipment (Sect. 4.3), and the software library (Sect. 4.4). We have been using the studio for more than three years in many projects. In the course of time we have been constantly improving the camera system in order to keep up with the progress in imaging technology. Our first set of CCD video cameras, *camera system - evolution I*, represented the state-of-the-art at the time of purchase (Sect. 4.2.1). It has been successfully employed in our research on marker-free motion capture and free-viewpoint video. However, the demands with respect to frame resolution and color quality that we were facing in our research on joint motion and reflectance capture could not be met by the old camera system. Thus, we moved to a new generation of acquisition hardware, *camera system - evolution II*, whose imaging sensors fulfill our needs (Sect. 4.2.2).

4.1 Studio Layout

The spatial dimensions of the studio have to be large enough to enable multi-view recording of dynamic scenes from sufficiently distant and widely-spaced camera positions. In our choice of an appropriate room we were constrained by the current availability situation at the MPII. We finally decided to install the studio in

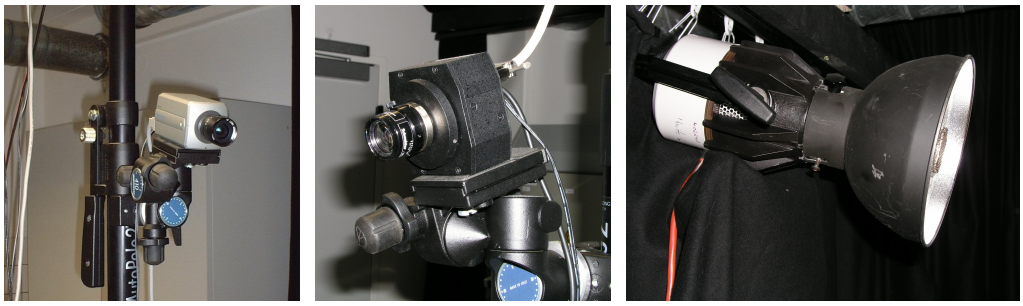
(a) (b)

Figure 4.1: (a) The control room of the studio. (b) Recording stage with checkerboard calibration pattern, black curtains and mounting poles for the cameras.

a terminal room of approximately 11 by 5 m in size. The ceiling has a height of approximately 3 m. An area of around 1.5 m by 5 m in size at the end of the studio serves as a control room in which all the computing hardware is installed. The remaining area of the studio, which can optionally be surrounded by opaque black curtains, is the stage used for multi-view recording. Unfortunately, the dimensions of the room do not allow us to put cameras in an overhead position over the recording area. In Fig. 4.1a,b the recording stage and the control room are shown.

4.2 Camera Systems

In the course of time we have employed two different camera systems in order to keep up with the forefront of imaging technology. The technical properties of each acquisition setup will be detailed in the following. Although both setups significantly differ in the capabilities of the employed imaging sensors, they have components in common. One of these components are the eight telescope poles (Manfrotto^{TM} Autopole [Manfrotto]) with attached 3-degree-of-freedom mounting heads (Manfrotto^{TM} Gear Head Junior [Manfrotto]) which we use as stands for our cameras. These telescope poles are jammed between the floor and the ceiling and enable us to reposition a camera within seconds to any arbitrary position in the studio.



(a) (b) (c)

Figure 4.2: (a) Sony^{TM} DFW-V 500 IEEE 1394 camera on mounting pole. (b) Imperx^{TM} MDC1004 camera. (c) K5600^{TM} Jokerbug 400 spot light with daylight spectrum.

4.2.1 Camera System - Evolution I

Our first camera system consists of eight Sony^{TM} DFW-V500 cameras which are controlled via an IEEE 1394 connection (Fig. 4.2a). Their CCD sensors have a frame resolution of 640x480 pixels. The maximal possible frame rate is 30 fps if the internal trigger of the camera is employed. External synchronization via a trigger pulse sent through a proprietary connector is also feasible. However, the frame rate in external trigger mode is 15 fps at the most. For distributing a trigger signal we constructed a device that broadcasts a pulse from the parallel port of a PC to all cameras.

Each camera delivers frames in YUV:4:2:2 format. Since the image data are readily available in digital form, a frame grabber board is not required. The cameras provide a high number of adjustable parameters that determine the image quality. Different parameter sets that were found to be optimal for different scenes can be stored in internal memory channels.

The computing infrastructure behind the camera system consists of four standard Linux PCs featuring AMD Athlon 1.1 GHz CPUs, graphics board with Nvidia GeForce 3 GPUs and 768 MB of memory. The PCs are used for camera control, video data processing, and data storage. The IEEE1394 bus bandwidth of 400 Mbit/s is sufficiently high to control two cameras from one host PC. Different project-specific software architectures can be easily implemented. All the software for camera control and recording has been developed by ourselves.

Camera system evolution I has proven its applicability in our research on human motion capture (Chap. 5 and Chap. 6) and (non-relightable) free-viewpoint

video (Chap. 8 and Chap. 9).

4.2.2 Camera System - Evolution II

In our second camera system we employ eight Imperx^{TM} MDC1004 single chip CCD cameras. Their imaging sensor has a resolution of 1004x1004 pixels and provides a color depth of 12 bits per sensor cell. The chip is equipped with a Bayer mosaic such that colored images can be reconstructed from the measured intensities via an appropriate reconstruction filter. The CCD sensor is connected to two controller chips. With both controllers activated the camera provides a sustained frame rate of 48 fps at full resolution. In dual-chip mode the photometric responses of the left and right half of the sensor do not comply such that an intra-frame color adjustment step is necessary. With only one chip activated, 25 fps at full resolution are feasible and no color balancing in the images is required.

The cameras are linked to a control PC which is equipped with 8 high-performance frame grabber boards. Each frame grabber is connected to one of the cameras through a Camera Link^{TM} interface. For maximal data throughput, each capture card is equipped with an onboard SCSI interface which enables the direct streaming of image data to a RAID system. In total we employ eight parallel RAID systems which enable real-time storing of eight full-resolution full frame rate video streams. The cameras are synchronized via a trigger pulse that is broadcasted to each capture card. The system was ordered out-of-the box according to our specifications. The manufacturer [Cos] also provided us with a custom-made control software (VideoSavant Pro [Vid]).

The performance of this acquisition setup fully unfolds in our project on joint motion and reflectance capture of dynamic scenes (Chap. 10).

4.3 Lighting Equipment

While the camera system is one factor that fundamentally influences the image quality of MVV streams, the lighting equipment is the second one. In order to establish the appropriate illumination conditions for different applications, it is important that lighting installations can be easily modified. For our purposes it is necessary that both an ambient scene lighting, as well as a more focused spot light kind of illumination can be set up.

In order to achieve this flexibility we employ three types of light sources. The first set of lamps are three rows of Siteco^{TM} louver luminaire [Siteco] neon tubes with attached reflectors that are positioned over the middle of the recording stage. These lamps produce a very intense diffuse illumination of the whole scene which produces only very smooth shadows. The same effect could have been reached

with spot lights and diffusers, but due to their large mounting form they are not suitable for installation in a room with a low ceiling. In case a more focused and intense illumination is desired, we have three spot lights at our disposal that can be mounted right below the ceiling using special clamps. Alternatively, a wooden frame installed along the top rim of the walls can be applied to mount the cameras.

For our project on joint motion and reflectance capture we have purchased an additional spot light (K5600 Jokerbug 400 [K56]) which emits light with a daylight spectrum (Fig. 4.2c). Different lenses can be used to modify the shape of the beam according to our needs.

In order to minimize the influence of light entering the room through the windows, and to keep the impact of indirect illumination from the walls at a minimum, the recording area can be completely surrounded by opaque black molleton. Optionally, indirect illumination reflected off the floor, as well as the visual appearance of cast shadows can be minimized by rolling out a black carpet.

4.4 Software Library and Algorithmic Toolbox

The standard software collection available on the computers in the studio provides a standard set of tools and libraries that we have developed to control the hardware components of our studio. A basic API provides the code for camera control of camera setup-evolution I and network communication. Multiple tools are at our disposition to perform basic image processing and camera calibration. In particular, our standard library provides implementations of the background subtraction scheme and the optical flow methods that are described in Chap. 2 Sect. 2.3.1 and Chap. 2 Sect. 2.3.2 respectively. The tools we employ to perform geometric and color calibration of our multi-camera setup are explained in more detail in the following.

4.4.1 Geometric Camera Calibration

For determining the intrinsic and extrinsic parameters of our cameras (Chap. 2 Sect. 2.2), we have two methods at our disposal. Both tools employ images of a calibration object of known structure and dimension to estimate parameters of a mathematical camera model (Sect. 2.2). For determining the extrinsic parameters of each camera, we typically apply the algorithm proposed by Tsai [Tsai86]. Our software identifies the internal corner positions of this pattern (with known world space position) in each of the camera views. A numerical minimization now tunes the parameters of the mathematical camera until the measured and predicted corner positions in the image plane comply. The Tsai algorithm can also estimate the

(a) (b)

Figure 4.3: (a) Small checkerboard used for intrinsic parameter estimation. (b) Color pattern used for multi-view color adjustment.

camera's intrinsic parameters (i.e. image plane origin, focal length, field of view and first order radial lens distortion) from the same calibration image.

An even better estimate of the camera's intrinsic properties is obtained from an image of a smaller calibration pattern which is positioned in front of the camera such that the whole field of view is covered (Fig. 4.3a).

An alternative approach for estimating internal and external camera parameters is the algorithm by Heikkila et al. [Heikkila96]. This algorithm also jointly estimates intrinsic and extrinsic parameters from images of a know calibration object. In contrast to the Tsai method it models the lens aberrations much more accurately by considering radial and tangential lens distortions up to second order [Jain95].

We have implemented both calibration tools such that they agree on a common file format for camera parameters. In our projects we frequently use Heikkila's method for the intrinsic parameters, undistort the calibration images accordingly, and finally estimate camera position and orientation by means of the Tsai algorithm.

4.4.2 Color Calibration and Multi-view Color Adjustment

To ensure a faithful color reproduction, all cameras are white-balanced before recording commences.

If two different cameras of the same model are used to capture the exactly same scene from the exactly same viewpoint they will, nonetheless, record images that differ in their color values. The reasons for these discrepancies are the joint result of noise and slight physical differences in the built-in camera components.

From physics we know that, if the illumination is constant, a purely diffuse surface in a scene will have the same color, no matter from what direction one looks at it. In order to establish such a multi-view color consistency in a multi-camera system a multi-view color adjustment has to be performed.

To this end, we estimate for each camera a trilinear transformation of the RGB color values. The parameters of this transformation are estimated from images of a diffuse calibration pattern which consists of an array of 237 uniformly colored squares with purely lambertian reflectance. We have a large (around 1.5 m by 1 m) printed version of this pattern (Fig. 4.3b) as well as a small version at our disposal. The color values for the latter one have been photometrically calibrated with a spectral reflectometer (see Gretag MacBeth homepage for details [Gre]).

With the large pattern, we typically perform relative photometric calibration. This means that we define one camera to be the reference camera. For each remaining camera, a color transformation is computed such that the color values of the pattern in the reference view are reproduced. In our projects, we usually employ this relative adjustment algorithm.

Absolute photometric calibration is also feasible if the small calibration pattern is applied. In both cases, we estimate the transformation in a least-squares sense.

Chapter 5

Marker-free Volumetric Motion Capture from Video

In this chapter we explain the nuts and bolts of a novel hybrid approach for full-body human motion capture from multi-view video streams that does not depend on optical markings in the scene [Theobalt02a, Theobalt02b, Theobalt04e]. The algorithm joins the forces of silhouette-based 3D scene reconstruction and 2D feature tracking in order to fit a kinematic skeleton to the motion data. A sequence of voxel-based 3D scene models is reconstructed from multi-view video footage by means of a visual hull reconstruction. The information contained in the volume data is enhanced by additional information on how salient features of the human body move over time. The features' motion is estimated by means of a color-based tracking algorithm. In combination, the two types of information enable fitting of a multi-layer kinematic skeleton to the motion data without having to rely on optical beacons on the body. The sequence of recovered joint parameters fully specifies the captured movements.

Recently, several approaches have been presented in the literature, that capture motion from visual hull sequences only (see Chap. 3 Sect. 3.1.3). However, these methods typically cannot robustly handle motion sequences in which limbs frequently move very close to the torso and merge with the torso geometry in the visual hull. In contrast, due to the appliance of a feature tracking method, our approach can handle these cases faithfully.

Our method introduces the following new scientific concepts:

- Joint employment of dynamic 3D scene reconstruction and feature tracking for the estimation of human motion parameters.
- A multi-layer skeleton-model that is specifically tailored to needs of the proposed algorithm.

In the following the algorithmic components will be detailed and results we obtained with a prototype implementation will demonstrate the performance of the method.

5.1 Overview

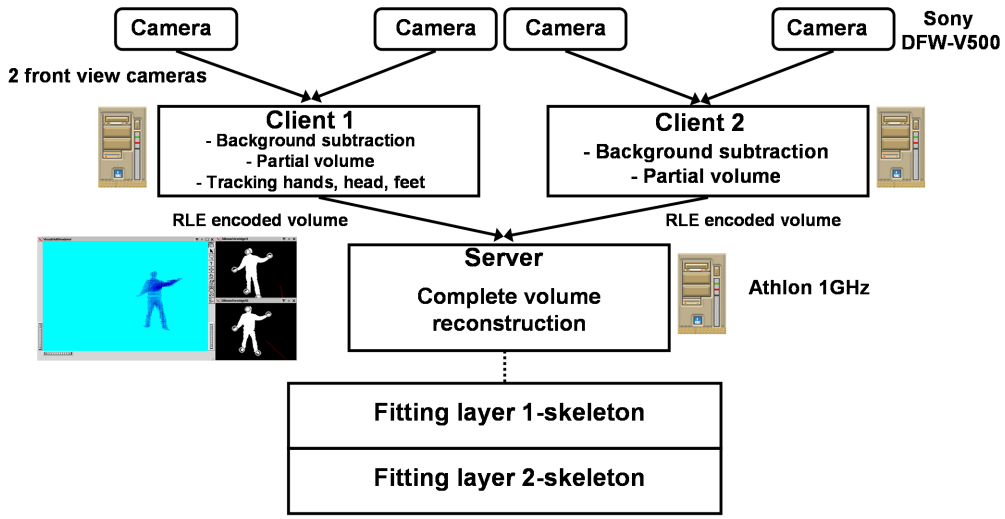


Figure 5.1: System architecture of our prototype implementation.

The algorithmic concepts that form the basis of this chapter are implemented in a prototype system whose components are illustrated in Fig. 5.1. The system makes use of camera equipment 1 and the associated computing infrastructure in our multi-view video studio (see Chap. 4).

Functionally, the system is partitioned into an online and an off-line component. In the online component, the image data are acquired and segmented, and sequences of voxel-volumes are reconstructed from silhouette images. In addition, the 3D locations of the hand, the head and the feet are tracked in one dedicated stereo pair of cameras. Once these data have been acquired and stored, a kinematic body model is fitted to each time step of video in the off-line component.

Using the software and hardware infrastructure of our multi-view acquisition studio, we implement the online component as a distributed client-server system, and run the off-line component on the server computer only.

In the online system, there are up to three clients, each of which is running on a 1.1 GHz single processor Athlon^{TM} PC. One client controls two Sony^{TM} DFW-V500 IEEE1394 video cameras that deliver color frames with a resolution of 320x240 pixels. For our tests, we run the system with two client machines and four cameras. In real-time, each client performs a background subtraction (Sect. 5.3), as well as computes a partial shape-from-silhouette (visual hull) volume using the two client camera views only. In addition, the client controlling the two front view cameras identifies and tracks the positions of hand, head and feet at an interactive frame rate (see Sect. 5.3 and Sect. 5.4). The partial visual hulls from both clients are transferred to the server PC which builds the complete visual hull and optionally displays it. The server also distributes the trigger signals to the cameras for synchronization. Due to the employment of a hierarchical client-server scheme the software architecture scales well to more cameras and more clients. The data acquired in a recording session with the online system (visual hull data, 3D feature locations) are stored to disk in real-time.

The off-line system which runs on a single PC applies these data in order to fit a multi-layer kinematic skeleton model to the motion sequence. The multi-layer skeleton model is adapted to the shape of the person using multi-view frames of a dedicated initialization pose which the person strikes before motion capture starts (Sect. 5.2). Thereafter, a two-layer kinematic skeleton model is fitted to the volume and feature position data (Sect. 5.6).

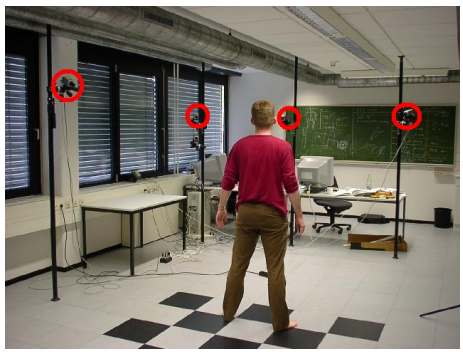


Figure 5.2: Scene setup: Camera studio, four visible cameras are encircled in red.

The person whose motion is captured is supposed to move inside a confined volume of space. The cameras are arranged in a convergent setup around the center of the scene. We require that the pair of cameras which tracks the body

features is always observing the person from nearby positions in front (Fig. 5.2). Tracking robustness is enhanced by having the person move barefooted. Due to the functional separation of the cameras, the person is supposed to face the front stereo pair of cameras, only allowing for limited rotations around the vertical body axis.

5.2 Initialization

In the first frame of video, the person strikes an initialization pose, facing the two front view cameras, with both legs next to each other and stretching the arms horizontally away from the body at maximum extent (a pose commonly referred to as the *T-pose*).

The bone lengths in the kinematic skeleton model need to be adjusted to the body dimensions of its moving real-world counterpart. One possibility is to measure the bone lengths manually and load them into the application. Alternatively, a semi-automatic procedure is feasible in which the user marks shoulder, hip, elbow and knee positions in the two front view images showing the person in the initialization pose. The positions of the head, the hand and the feet in both of the images are found via a structural analysis of the person's silhouettes (Sect. 5.3). Having the image plane locations of corresponding joints and features in a stereo pair of images, their 3D positions are found via triangulation. The thicknesses of the arms and legs are determined with user interaction.

5.3 Silhouette Subdivision

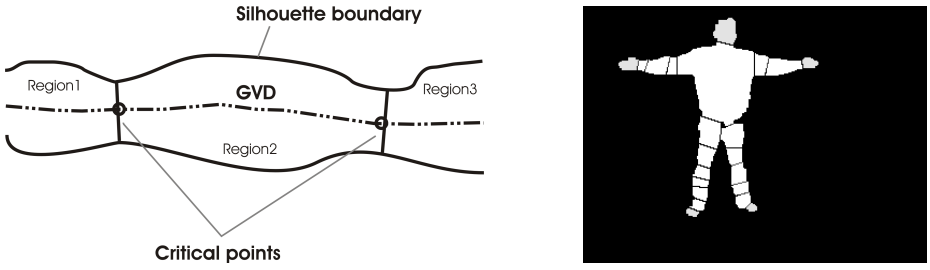
Each client in our prototype system performs a real-time color-based background subtraction (see Chap. 2 Sect. 2.3.1) on each video frame to compute the silhouette of the person. On two video frames that show the person in the T-pose and that were captured with the front stereo pair of cameras a silhouette subdivision step is performed. By means of this segmentation, the initial positions and color ranges of the head, the hand, and the feet in the image planes are determined.

The two front-view silhouettes are subdivided into topological regions by means of a Generalized Voronoi Diagram (GVD) decomposition (see Fig. 5.3). This algorithm is commonly used to segment areas of free space in cognitive topological maps of mobile robots [Rowat79, Thrun98, Latombe91], a problem very similar to ours. The Generalized Voronoi Diagram is the set of all points in the silhouette which is equidistant to at least two silhouette boundary points. Hence, it represents one way of computing a medial axis of the silhouette area.

The GVD point set is used to segment each silhouette into distinct topological

regions by searching for critical points, i.e. points locally minimizing the clearance to the silhouette boundary. These points mark the centers for separation lines between adjacent regions in the silhouette. The separation lines pass through the critical point and connect the two boundary pixels that are closest to it (Fig. 5.3a). In the silhouettes of the initialization pose the boundaries to the head, the hands and the feet are marked by constrictions. Thus, the five features appear as separate topological regions in the silhouettes. The proposed decomposition scheme subdivides a silhouette into many small regions (Fig. 5.3b). The connectivity of the recovered silhouette regions can be represented by a graph connecting the region centers. In the case of the human silhouette in the initialization pose, the five terminating nodes in the connectivity graph correspond to the head, the hands and the feet of the person.

The advantage of a topological region identification is that it enables the discovery of hands, head and feet of arbitrary color. This way a specific feature color can be determined for each tracked subject and no a priori assumptions about skin color have to be made. The specific feature color is thereafter used in our color-based feature tracking algorithm (Sect. 5.4).



(a) (b)

Figure 5.3: (a) GVD with critical points. (b) Silhouette segmented by Generalized Voronoi Diagram decomposition.

5.4 Tracking Selected Body Parts

To track the motion of selected body parts in the image plane, we employ a fast color-based tracking scheme. We use a continuously adaptable mean-shift algorithm which is capable of tracking the mean of dynamically changing probability distributions in real-time. Originally the algorithm has been developed for face tracking [Bradski98, Fukunaga90]. The workflow of the method is shown in

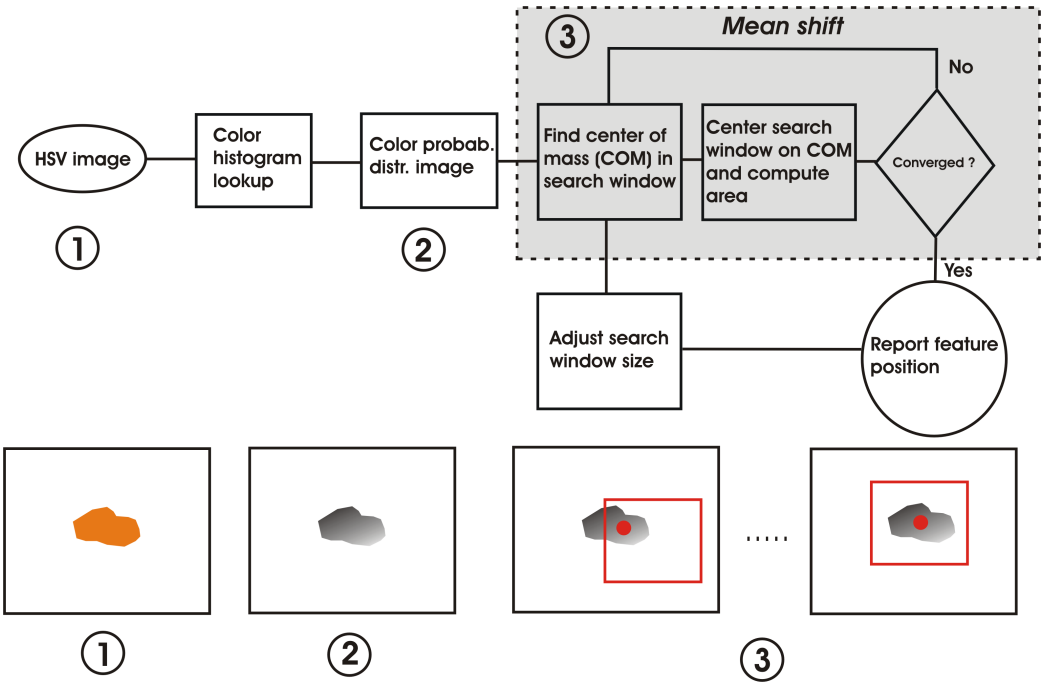


Figure 5.4: Continuously-adaptable mean shift workflow. Some algorithmic components in the diagram are visually illustrated by the images in the lower part of the figure.

Fig. 5.4. It is capable of following the image plane location of a moving image region which is specified by a range of colors of its member pixels.

For one time step of video, the algorithm works as follows. Assuming that an approximate position of the image region is known, only a sub-window of the image plane needs to be considered. For each pixel within this sub-window a probability of belonging to the tracked region is approximated by looking up its color in the histogram of allowed region colors. The entry in the histogram bin associated to the pixel color is an estimate of the pixels probability of belonging to the region (after appropriate scaling; a process also known as histogram back-projection). All pixel membership likelihoods are stored in a monochromatic probability image. Now the core of the tracking method, the mean shift algorithm, is applied. The mean shift performs the following steps iteratively:

1) Compute the mean of the probability distribution within a search window.

2) Re-center the search window at the detected mean.

3) Repeat from step 1 until convergence. The algorithm converges if the

change in the mean position is below a threshold.

When the mean shift has terminated, the search window size is adapted before the tracker proceeds to the next time step of video.

In our system, we apply the above algorithm in the following way: A separate tracker is applied to follow the location of each body feature in both front camera views. Each tracker only considers a sub-region of the image plane in the neighborhood of every individual feature. The HSV color is the principal cue used for tracking. Since the locations and extents of the head, the hands and the feet in the image planes at time t are known, average colors for each region can be derived. These values are used to define tolerance intervals in color space centered at the mean color. For the colors in these intervals, color histograms are computed based on the video frames with the person in initialization position. At time $t = 0$ each tracker's search window is initialized at the mean positions of the head, hand and feet regions that were found during silhouette subdivision. For all subsequent time steps the algorithm proceeds in the way which is shown in Fig. 5.4. In our implementation we modified the probability image computation step slightly. In addition to the histogram back-projection we also provide the possibility to simply feed a 0/1-silhouette mask image as probability image into the tracking scheme. This image can be derived very efficiently from the overall silhouette image and our experience shows that the coarse approximation of the probability distribution does not significantly deteriorate the tracking quality. Fig.5.5 shows a screen-shot of our system where the tracked body parts are encircled.

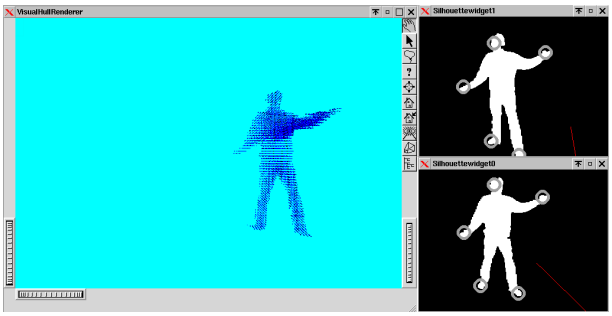


Figure 5.5: A screen-shot of the server application showing the visual hull (l) and silhouettes with tracked feature locations (r).

This method is subject to some limitations that are typical to color-based feature tracking schemes. First, we implicitly assume that the colors of the head, the hands and the feet are sufficiently different from the colors of the clothes that the person wears. Requiring that the person moves barefooted is one feasible way to

fulfill this constraint for most types of apparel. Furthermore, in situations where the different tracked regions merge in the image plane, the trackers may be mislead.

From their locations in the image planes of the front stereo pair the 3D positions of the body parts are computed via triangulation. We assume that the tracked centroids of the hands correspond to the projected wrist joint locations, the centroids of the feet to the ankle joint locations, and the centroid of the head to the model root joint.

5.5 Volume Reconstruction

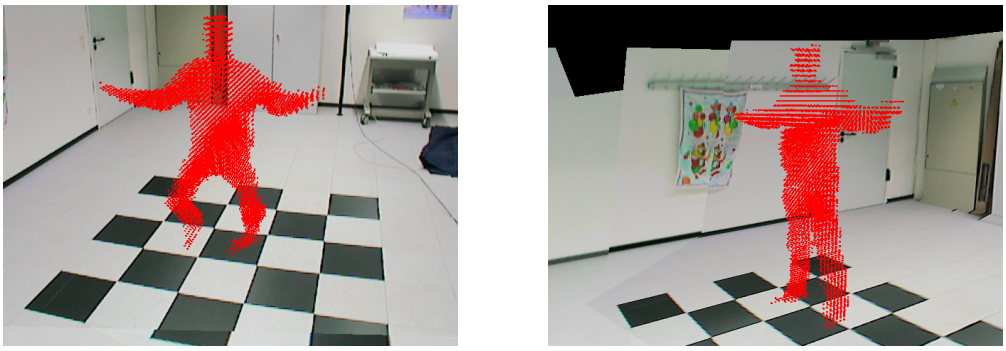


Figure 5.6: Visual hulls reconstructed from four camera views. The volumes are rendered into a model of the acquisition room.

The second type of input data that we apply to infer a person's motion parameters are volume data of the moving person which we reconstruct by means of a shape-from-silhouette approach. From the silhouettes of the moving person, we reconstruct a voxel-based approximation to what is commonly termed the visual hull [Laurentini94] by intersecting the back-projected silhouette cones from each camera view (Fig. 5.7). Our implementation of visual hull reconstruction is a voxel carving method and is similar to the algorithms presented in [Cheung00] and [Luck01].
Our voxel carving approach carves the visual hull of the person out of a box in space in which the person is allowed to move and which is subdivided into a regular grid of volume elements. From camera calibration, the camera matrices are known. Thus, it is possible to compute the projected 2D image plane location of every point in 3D in each camera view. The distributed voxel carving implementation in the online system classifies voxels as follows. On each client PC every voxel in the grid is simultaneously projected into the views of the two cameras

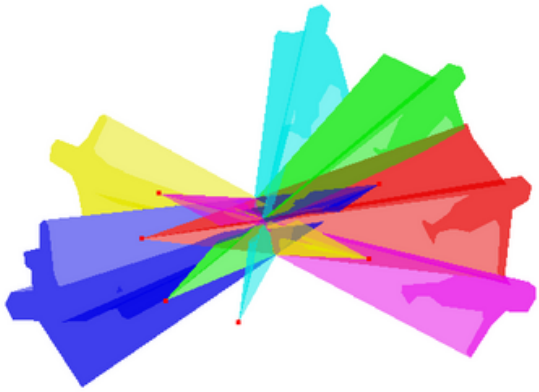


Figure 5.7: Illustration of the principle of visual hull reconstruction. The visual hull is the intersection of all silhouette cones back-projected into the scene from each camera's center of projection. (Figure courtesy of Ming Li)

connected to it. If the volume element projects into the silhouettes of the person in both views, it is classified as occupied space. Otherwise it is classified as empty. This way, each client computes a partial visual hull from two camera views that it controls. The partial hulls from each client i, $\mathcal{V}_i$, are run-length-encoded and transferred to the server application. On the server, the complete visual hull $\mathcal{V}_{\text{complete}}$ is constructed by intersecting the volumes, $\mathcal{V}_{\text{complete}} = \bigcap_i \mathcal{V}_i$. Visual hull reconstruction is significantly sped up by precomputing all projected voxel locations for the whole volume grid and storing them in camera-specific lookup tables. Two example visual hulls reconstructed with our system are shown in Fig. 5.6.

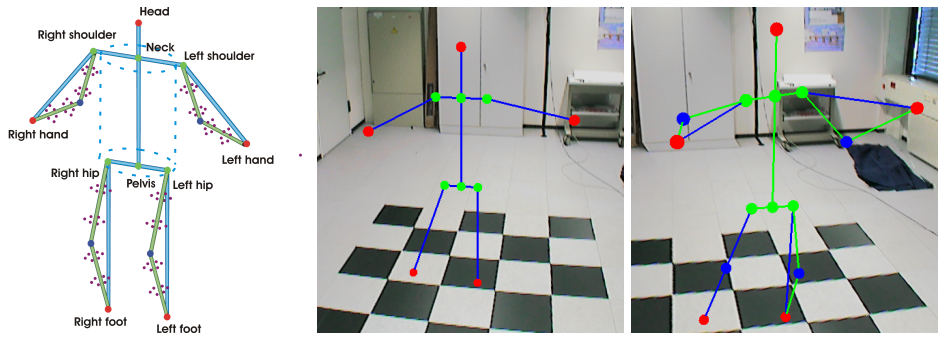
5.6 Skeleton Fitting

Given sequences of 3D locations of head, hands and feet and a sequence of visual hull volumes, the skeleton fitting algorithm estimates a set of motion parameters for each time step. This is achieved by fitting a two-layer hierarchical kinematic skeleton to the motion data (Fig. 5.8).

For each time step of the input motion sequence, the skeleton fitting procedure performs three subsequent steps. In a first step, the orientation of the torso segment is estimated. In a second step layer 1 of the skeleton model is fitted, and in the third step, the pose parameters for the refined layer 2 skeleton are found. The joint parameters for time $t = 0$ are known from the initialization pose. The fitting procedure exploits temporal coherence by starting parameter estimation from the

body pose estimated in the preceding time step. In the following the employed multi-layer skeleton model is detailed. Thereafter, the three fitting steps are explained.

5.6.1 The Multi-layer Kinematic Skeleton



(a) (b) (c)

Figure 5.8: (a) Joint illustration of skeleton layer 1 and layer 2 with the attached cylinder samples and the cylindrical torso area. Skeleton layer 1 (b) and skeleton layer 2 (c) rendered into a model of the camera room after they have been fitted to a body pose.

The human body is modeled as a two-layer kinematic skeleton. The first layer of the model consists of a hierarchical arrangement of 10 bone segments and 7 interconnecting joints. Each joint defines a rigid body transformation represented as a rotation matrix $\mathbf{R}$ between its parent segment and the subsequent kinematic elements in the hierarchy. The translation of the model is specified with an additional translation vector. The root of the hierarchy is located at the head. Since 3 parameters are needed in each joint to define the rotation and 3 additional parameters are needed for the model translation, 24 degrees of freedom (DOFs) are provided on skeleton layer 1.

The second layer enhances layer 1 by providing more detailed representations of upper and forearms, as well as thighs and lower legs (Fig. 5.8). The volumetric extents of the corresponding limbs are modeled by means of point samples taken from cylindrical volumes centered around the bone axis, henceforth called cylinder samples (Fig. 5.8). Thus, on layer 2 every limb is represented via a root joint (shoulder or hip) and two bone segments that are connected via a 1-DOF revolute joint (elbow or knee) (Fig. 5.9b). Although this parameterization is only

an approximation to the full motion range of a human limb it can represent the majority of possible actions with only 4 parameters. The lengths of the layer-2 limb segments, e.g. $l_{forearm}$ and $l_{upperarm}$ for the arm in Fig. 5.9b, are constant and known from initialization. The lengths of the attached layer-1 segments (e.g. l_{whole} for the arm in Fig. 5.9b), i.e. the distance between the root joint of a limb and the tip of the limb, may vary while the person is moving. The layer-1 limb segment and the attached layer-2 bones form a triangle. The bending angle of the middle joint (henceforth denoted by ϕ) can be determined via the cosine theorem (see Sect. 5.6.3). The rotational degree of freedom (henceforth denoted by ρ) of the layer-2 arm and leg constructions around the corresponding layer-1 bone is found via determining an optimal overlap between the visual hull and the volume samples (Sect. 5.6.4).

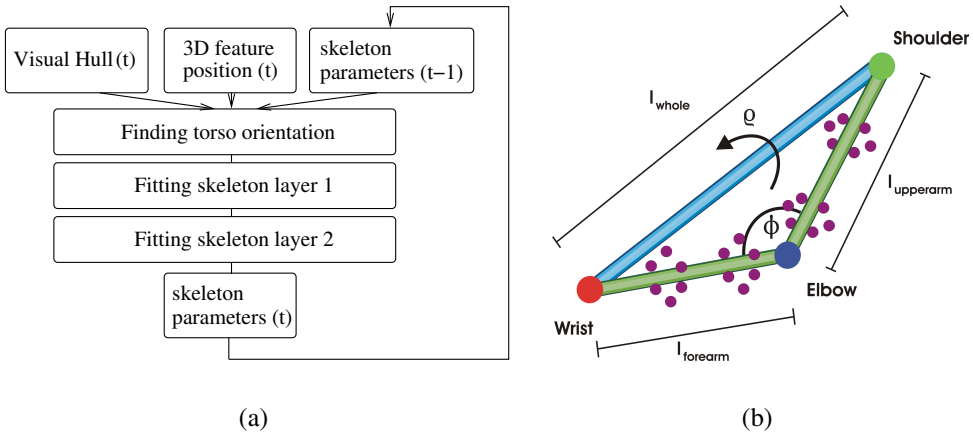


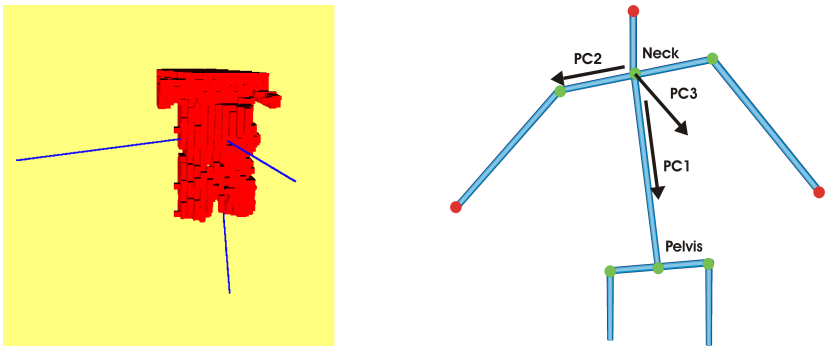
Figure 5.9: (a) Illustration of the interplay between the three skeleton fitting steps. (b) Arm structure on model layer 2.

5.6.2 Step 1: Finding the Torso Orientation

Purely image-based optical tracking of the torso is difficult due to the lack of detectable salient features. Fortunately, we can recover the torso orientation from the volume data. We achieve this by fitting a cartesian coordinate system to the torso voxels by means of a principal component analysis (PCA) [Jolliffe86]. The positions of the torso voxel centers are interpreted as a set of 3-dimensional data points whose coordinate origin is located in the center of gravity of the voxel set. For this set a 3×3 covariance matrix $\mathbf{C}$ is computed. The three eigenvectors of the symmetric matrix $\mathbf{C}$, the principal components (PCs), denote the directions of

strongest variance in the data and are mutually orthogonal. If the data is limited to the voxels corresponding to the torso of the person, the first principal component lies parallel to the spine axis, the second one lies parallel to the connection between the shoulder joints, and the third one is orthogonal to the other two (see Fig. 5.10). We ensure that only torso voxels are used during PCA computation by restricting the data set to those voxels which lie inside a cylindrical volume around the spine bone of the skeleton (Fig. 5.8a).

Since, prior to PCA computation, the correct set of pose parameters for the current time step is not known, we use the orientation of the bounding cylinder found in the previous time step to classify the torso voxels. This approach is feasible because it is valid to assume that the changes in torso orientation between two subsequent body poses are marginal.



(a) (b)

Figure 5.10: (a) The principal components of all the voxels inside the torso. (b) Aligning the skeleton with the recovered torso orientation.

5.6.3 Step 2: Fitting Skeleton Layer 1

The 2D feature tracking (Sect. 5.4) outputs sequences of 3D positions for the head, the hands and the feet. In combination with the information on torso orientation, this enables fitting the layer-1 skeleton. During initialization, the lengths of all layer-1 bones have been determined. Apart from the 4 limb bones, their lengths remain constant in time. To fit the layer-1 skeleton to the five body feature positions, we make a number of simplifying assumptions. The neck bone is assumed to be upright at all time steps, so that the 3D location of the neck joint in world coordinates is known from the 3D location of the head. The model's root, located at the head, is translated to match the triangulated 3D head position at each t.

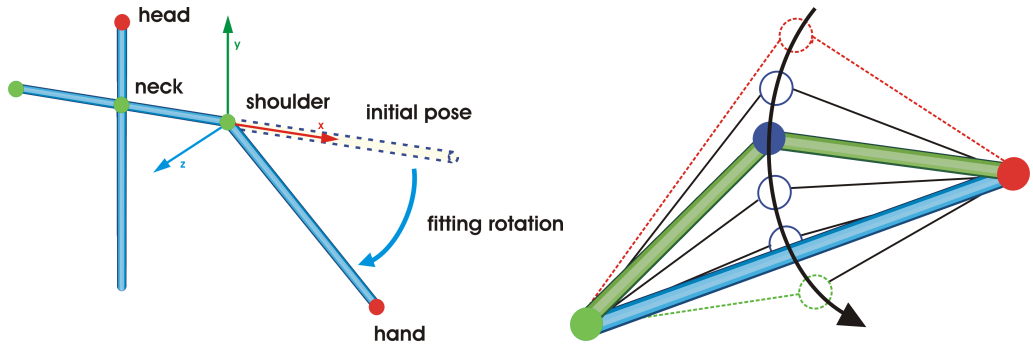


Figure 5.11: (a) Illustration of layer-1 fitting using the left arm segment as an example. (b) Sampling candidate values for rotation angles ρ between search interval bounds (stippled lines).

The principal components of the torso voxels define the rotation of the neck joint at time step t (Fig. 5.10). The rotation matrix $\mathbf{R}_{neck}$ is easily constructed by using the PC vectors. The so created rotation makes the neck joint's local coordinate frame comply with the coordinate frame spanned by the PCs. To keep the hip bones parallel to the floor level, the pelvis joint rotation is set to the inverse neck rotation.

Now the 3D locations of shoulder and hip joint in world coordinates as well as the locations of hands and feet are known. The distances between the left and right shoulder and hand as well as the left and right hip and foot are computed, and the lengths of the corresponding layer-1 segments are accordingly rescaled.

The rotations of the shoulder and the hip joints at the current time step t can now be derived. The direction of each layer-1 arm segment has to comply with the direction of the line connecting 3D shoulder joint and hand position. Using an axis-angle representation, the corresponding rotation matrix can easily be constructed. The rotation axis is given by a vector orthogonal to the layer-1 bone and the line connecting the hand and the shoulder. The rotation angle is the angle between the two lines.

5.6.4 Step 3: Fitting Skeleton Layer 2

Once the pose parameters for the first skeleton layer are found, the additional degrees of freedom of the second model layer are recovered by using the visual hull information. For each limb, the side lengths of the triangle formed by the

layer-1 segment and the attached two layer-2 segments are known. The cosine theorem for triangles [Bronstein91] uniquely determines the angle ϕ (illustrated for the arm in Fig. 5.9) of the revolute joint between the layer-2 segments.

In order to find the rotation angle $\rho(t)$ of each layer-1 arm and leg segment (see also 5.6.1), a maximal overlap between the set of cylinder samples attached to the layer-2 model and the voxel data obtained from the visual hull is computed. The search procedure works as follows, using the arm segment as an example:
We start with the rotation of the arm in the previous frame, $\rho(t-1)$, and rotate the arm segment to ν equidistant angles ξ_l in the interval $[\rho(t-1)-s, \rho(t-1)+s]$, with s defining the search neighborhood size. For each such angle ξ_l a quality measure for the overlap between the cylinder samples and the visual hull, $match_l$, is computed. The value of this quality measure is the larger the better the model fits to the voxel set. To this end, for each cylinder sample, the corresponding voxel of the visual hull that it currently overlaps with is computed. If n is the number of visual hull voxels which overlap with at least one volume sample, then n^k is the overlap match score for the current configuration ξ_l. In our experiments we found out that a value of $k = 4$ produces good results.
Using the set of ν match scores, the final rotation $\rho(t)$ of the arm segment is found by computing the center of mass of the weighted set $\Xi = \{\xi_l \times match_l \mid l = 1, \ldots, \nu\}$, the set of angles ξ_l which have been multiplied by their respective match score. This particular match function is a heuristic which ranks the best overlaps overproportionally high. The same procedure is applied to the leg segments. Although the difference between match scores for neighboring ξ_l can be very small, this approach still allows us to recover small changes in rotation from $t-1$ to t. Degradation of tracking quality due to the accumulation of model fitting errors on layer 2 is prevented by searching for the best fit in a search interval at every time step. Thus, erroneous fits in individual time steps due to noise in the visual hull data do not propagate over time.

Faithful matching results can only be obtained if the respective knee or elbow joint is at least slightly bent. If this is not the case, the rotation angle $\rho(t-1)$ from the previous time step is passed on as $\rho(t)$.

5.7 Results and Discussion

The system has been tested on several sequences of a person performing gymnastics moves. Figure 5.14 shows the skeleton configurations that we found for several body poses. Looking at the scene from different positions, one can see that both layers of the skeleton have been correctly fitted to each stance. The orientations of the shoulders and the torso are also correctly recovered.

In Fig. 5.12 the temporal evolution of the rotation angle ρ on layer 2 of the

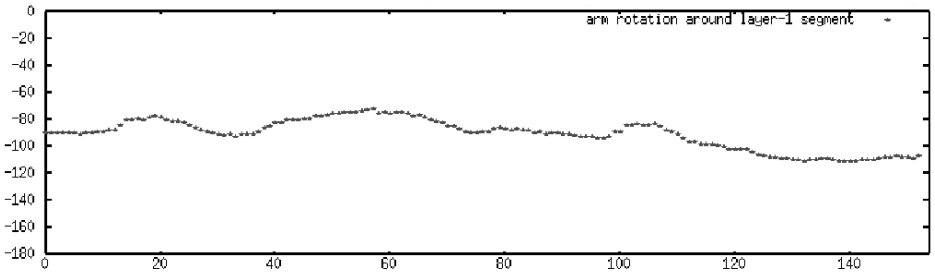


Figure 5.12: Plot of the rotation angle ρ in the arm on model layer 2 for a sequence of time steps.

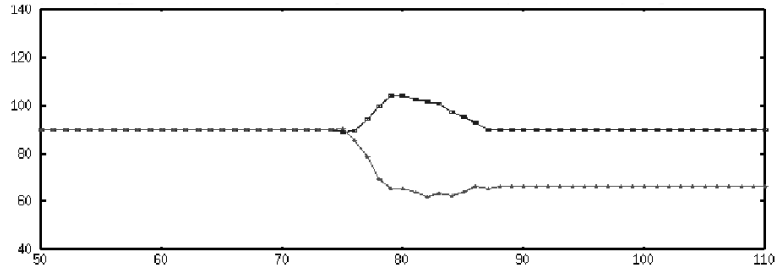


Figure 5.13: Comparison between layer-2 rotation angle ρ in both legs while the person is prostrating.

arm is depicted. Fig. 5.13 shows a comparison plot of the same angle in the two legs for a motion sequence in which the person is prostrating. While bending the knees the legs were slightly rotating to the outside in opposite directions. This fact is nicely visible in the plot.

The number and positions of the cameras are crucial for the quality of the visual hull. Typical reconstruction errors produced by shape-from-silhouette approaches are visibility artifacts (also known as phantom volumes), which in our case, may occur in the form of bulgy arms or legs. Despite this noise in the data, our approach still recovers the correct arm and leg poses. A camera looking at the scene from the top is advantageous but not required. Even with only four cameras recording from lateral viewpoints robust fitting is possible.

The data obtained with feature tracking and volume reconstruction fruitfully complement each other. The information on the correct head, hands and feet positions enables robust model fitting even in cases that are problematic for pure volume-based motion capture approaches [Cheung00, Bottino01]. For instance, if the arms are very close to the chest the feature tracking prevents them from getting stuck in the torso volume.

We have measured the execution speeds of the different algorithmic components using our reference prototype running on 3 Athlon 1.1 GHz PCs. The combined visual hull reconstruction, background subtraction, feature tracking and visual hull rendering runs at approximately 6-7 fps for a 64^3 voxel volume using two client computers and one server. Our measurements show that currently feature tracking consumes over 30% of total computation time. Furthermore, we experience a network overhead in our current implementation, since the frame rate of one client running independently can reach up to 19 fps. The performance of the model fitting strongly depends on the chosen parameters, such as the number of cylinder samples and angular search steps. On an average motion sequence it takes around 0.5 to 1 s to fit both layers of the body model to one time step. Pose determination for the layer-1 model only can be performed at acquisition frame rate. The estimation of the motion parameters on layer 2 is computationally more challenging. Higher frame rates can be achieved if less cylinder samples and less search steps are used. With the faster PCs available today significantly higher frame rates can be achieved.

Our approach is subject to a few limitations. The feature tracking in the online system and constraints in the model parameterization currently limit the range of movements which can be captured. The fitting method itself, however, allows arbitrary rotations of the human actor around the vertical body axis. Furthermore, it is feasible to extend the method to handle arbitrary body orientations by pose-dependent switching between stereo camera pairs. There are also some circumstances under which the quality of our color-based feature tracking will deteriorate. This may happen, if the color of the hands or feet is very similar to the color of the rest of the body. Furthermore, situations in which the motion of individual body parts overlap in the image plane are hard to distinguish. One possible solution to the latter problem would be the incorporation of a motion prediction scheme into the tracking method.

Despite these limitations our results show that our hybrid approach to marker-free human motion capture correctly estimates human body poses. The joint application of volume data and feature positions enables use to robustly determine pose parameters despite the noise in our measurements.

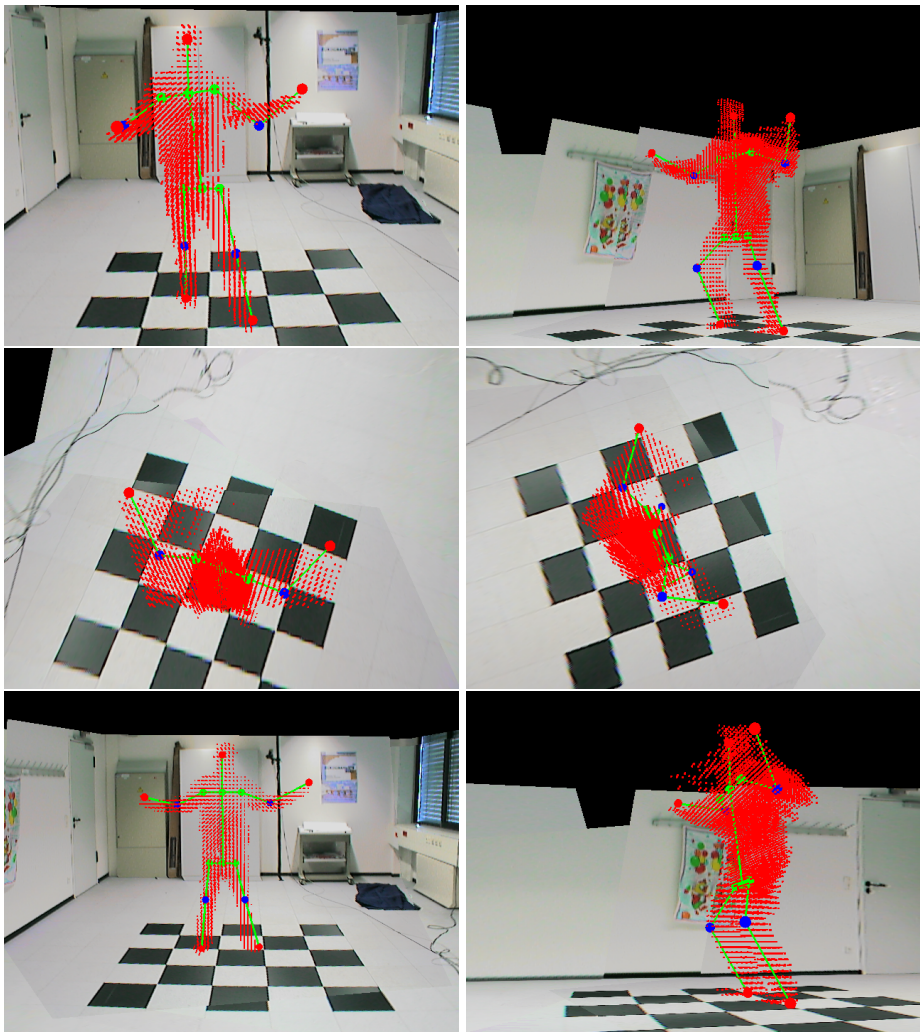


Figure 5.14: Skeleton fitted to visual hulls (rendered as small cubes) of a moving person. For these results, the person has been recorded from four camera views. In the middle row undersampling artifacts in the visual hull arising as bulgy arms can be seen. Our motion capture method nonetheless correctly recovers the body pose.

Chapter 6

Marker-free Body Model Estimation from Video

Chapter 5 presents a novel marker-free technique for estimating human motion parameters from multi-view video streams. In the same way as many related methods from the literature, this approach employs a kinematic human body model. While an algorithm to estimate motion parameters based on a known body model is a central component of each human motion capture system, a way to automatically infer this body model is equally important. It has been demonstrated that, if optical markings are employed, it is feasible to automatically estimate articulated body structures [Silaghi98]. Despite its relevance, the problem of fully-automatic marker-free skeleton estimation from video footage has hardly been considered up to today. Only a handful of approaches have been published so far that try to solve this problem. However, apart from only being applicable to specific prescribed motion sequences, they also fall short of the general case of arbitrary moving subjects.

In this chapter we introduce a novel method [de Aguiar04, Theobalt04d] that

- estimates the kinematic structure of the moving subject from multi-view video without optical markings in the scene;
- does with no significant a priori knowledge;
- is applicable to arbitrary moving subjects, including humans and animals.

The inputs to our algorithm are sequences of voxel volumes that are reconstructed from multi-view video streams by means of a shape-from-silhouette approach. At each time step the volumes are subdivided by fitting primitive shapes, also referred to as *approximators*, to the voxel data. We have developed the algorithmic frameworks for two types of shape primitives, ellipsoidal shells and

superquadrics. Exploiting the temporal dimension, we can identify correspondences between shape primitives over time and thus identify coherent rigid body parts. Knowing the motion of the rigid bodies over time, a complete kinematic skeleton model for the moving subject can be reconstructed. Optionally, we can obtain a first estimate of the motion parameters based on the derived body representation.

In this work we draw from ideas that have been developed in the fields of shape classification and shape approximation. Characterizing 3D point clouds by means of fitting primitive shapes is a common approach in 3D shape analysis (see [Loncaric98] for a survey) where it is typically applied to static data. In [Chevalier03], multiple superquadric shapes are used to decompose 3D point data into primitive sub-shapes. The same category of geometric primitives is commonly used in computer vision for object recognition, range map segmentation [Leonardis97], and analysis of medical data sets [Banégas01]. We extend these ideas by fitting primitive shapes to time-varying data and deriving kinematic information from their motion over time.

Most similar to the method presented in this chapter are the approaches by Cheung et al. [Cheung03] and by Kakadiaris et al. [Kakadiaris95]. In the former work a kinematic skeleton is estimated from a sequence of voxel volumes reconstructed by means of a shape-from silhouette approach. The person is required to perform a sequence of initialization moves with each limb separately. In the latter work a kinematic body model is reconstructed from multiple video streams that show the silhouette of the moving person from different camera positions. This method prescribes a sequence of moves with individual limbs, too.

In contrast we present a more flexible approach which requires only a minimum of a priori knowledge about the observed subject. It can be applied to moving humans, mechanical structures, and animals. Furthermore, it infers body models from any arbitrary motion sequence. We demonstrate the performance of our algorithm using both real and synthetic input data.

6.1 Overview

In Fig. 6.1 the algorithmic workflow of our method is illustrated. The system expects a voxel volume $V(t)$ for each time step t of video as input (Sect. 6.2). In step 1, the Shape Primitive Fitting step, each $V(t)$ is filled with either superquadrics or ellipsoids by means of a split and merge approach (Sect. 6.3). The result is a set of fitted approximators $U(t)$ and a list of associated voxel subsets $S(t)$ for each time instant. The correspondences between quadric primitives at different time instants are established by means of a dynamic programming method in step 2, the Shape Primitive Matching step (Sect. 6.4). The result of step 2 is a

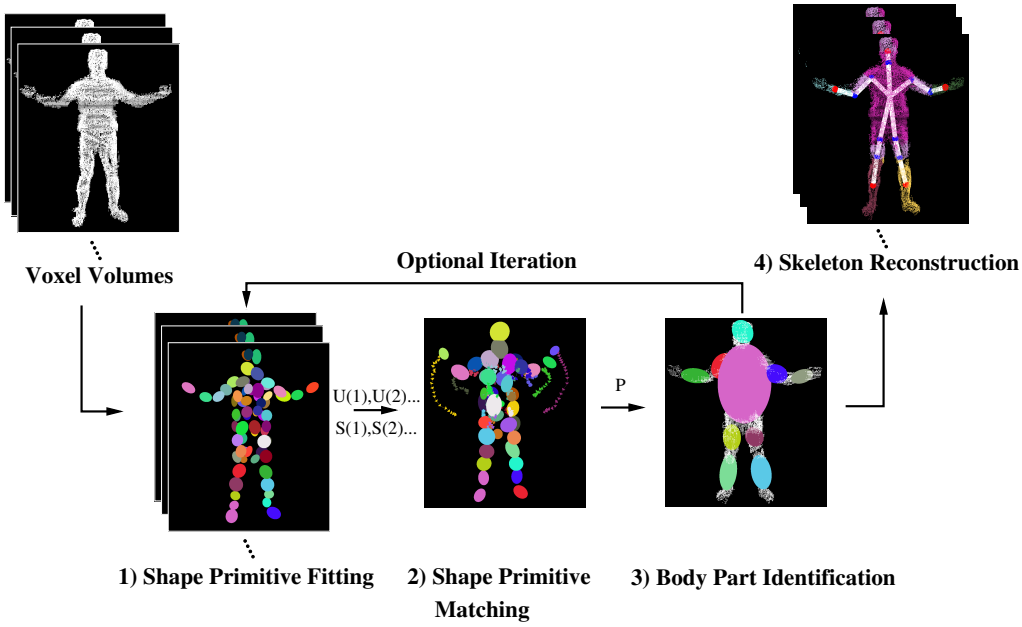


Figure 6.1: Visualization of the algorithmic workflow.

path for each primitive shape that describes its motion over time. All individual paths are subsumed in the path set P. By analyzing their motion, the ellipsoids or superquadrics are clustered into separate rigid bodies in step 3, the Body Part Identification step (Sect. 6.5). After step 3, the motion of each rigid body over time is known, and joint locations between neighboring bodies can be estimated in step 4, the Skeleton Reconstruction step (Sect. 6.6). This step also enables approximative estimation of body motion parameters based on the derived skeleton model. The final output is a complete kinematic skeleton model. Optionally, steps 1-3 may be iterated on subsets of the volume data (Sect. 6.7).

6.2 Input Data

We have tested the algorithm both on data acquired in the real world as well as on synthetic data generated with a 3D animation package.

The video footage acquired in the real world was recorded in our multi-view video studio using camera equipment-evolution 1 (Chap. 4). The eight synchronized FireWire^{TM} cameras were placed in a convergent setup around the center

of the scene. For these experiments we operated the cameras at a resolution of 320x240 pixels and at a frame rate of 15 fps. The cameras were metrically calibrated into a common coordinate system. We record several motion sequences of a test person who was performing simple gymnastic exercises.

We apply a space-carving approach to reconstruct high-quality time-varying voxel models of the moving person [Kutulakos00]. We first reconstruct a plain voxel-based visual hull at each time step of video by intersecting the reprojected silhouette cones (see Chap. 5 Sect. 5.5). This coarse voxel-model still exhibits artifacts which are due to the limitations of the visual hull approach, such as incorrectly reconstructed concavities. The space-carving method eliminates many of these artifacts by removing all those voxels from the visual hull which are not photo-consistent in all camera views. A voxel is photo-consistent in multiple camera views if the color values at the image plane locations to which a voxel reprojects in each camera are identical (within a tolerance interval). The sequence in which voxels are tested for photo-consistency has to take into account changes in voxel visibility through the carving process. It has been proven [Kutulakos00] that a correct order of consistency checks is achieved if a plane-sweeping approach is employed. Multiple plane sweeps along the three main axes of the voxel volume are typically necessary until the method terminates. Space carving converges, if no more voxels are carved out of the volume. In the end we obtain a set of surface voxels, also known as the photo-hull, for each time step of a motion sequence.

We want to demonstrate that the proposed algorithm can also infer the body configuration of moving subjects that are not human. Unfortunately, due to security reasons and ethical concerns, it turned out to be difficult to find animals which perform in front of our cameras. Thus, in order to complement the human motion data that we recorded in our multi-camera studio, we chose the safer option and created several synthetic data sets. The synthetic sequences were generated with 3D Studio Max^{TM} by placing animation skeletons into the surface meshes of a bird, a snowman and a monster. Animations with these models were created via key-framing. For each time frame of animation, a separate surface voxel set was exported.

6.3 Shape Primitive Fitting

We have tested two types of shape primitives, simple ellipsoids and superquadrics. While the former class of primitives can be fitted very efficiently to volume data, it lacks the generality of superquadrics which can approximate a much larger range of shapes more accurately. However, the additional flexibility of superquadrics

comes with the cost of a more time-consuming fitting procedure. After briefly describing the properties of either shape primitive individually, we will later refer to an ellipsoid and a superquadric as a shape primitive or approximator $\mathscr{U}$.

6.3.1 Ellipsoids

An ellipsoid is a closed surface defined as the solution to the implicit equation

$$F(x,y,z) = (\frac{x}{a_1})^2 + (\frac{y}{a_2})^2 + (\frac{z}{a_3})^2 = 1 \tag{6.1}$$

where a_1, a_2 and a_3 are scaling factors along the three coordinate axes. $F(x,y,z)$ enables a simple test for deciding if a point (x,y,z) lies inside ($F < 1$), on the surface of ($F = 1$), or outside ($F > 1$) the primitive shape. An ellipsoid in a general position is described by three additional rotation parameters (R_x, R_y, R_z) and three translation parameters (T_x, T_y, T_z) with respect to the world origin. Thus, in order to fit an ellipsoid $\mathscr{E}$ to a set of N 3D points (in our case surface voxel centers) such that its surface comes as close as possible to all points nine shape parameters $\mathscr{E} = [a_1, a_2, a_3, R_x, R_y, R_z, T_x, T_y, T_z]$ need to be determined. Using the following procedure we can robustly and quickly fit ellipsoids while avoiding a time-consuming numerical optimization. First, T_x, T_y, T_z are found as the 3D location of the voxel set's center of gravity. The six remaining parameters are found via moment analysis [Cheung00], i.e. the directions of the main axes of variation in the 3D voxel set are found as the eigenvectors of the point set's covariance matrix. The optimal radii a_1, a_2, a_3 along the main axes are found as $a_j = 2 \cdot \sqrt{\lambda_j}$, λ_j being the eigenvalue corresponding to eigenvector j [Banégas01]. The initial rotation parameters R_x, R_y, R_z are also derived from the directions of the eigenvectors.

This procedure computes an ellipsoidal fit very quickly, but it does not provide a direct measure of the fitting quality. Hence we calculate a fitting error (FE) D that gives a numerical estimate of how well the ellipsoid approximates the point data. The error function sums up the squared distance values $d(\mathscr{E}, x, y, z)$ of voxel centers to the ellipsoid surface:

$$D = \frac{\sqrt{a_1 a_2 a_3}}{N} \sum_{i=1}^{N} d(\mathscr{E}, x_i, y_i, z_i)^2 \tag{6.2}$$

$$\text{with } d(\mathscr{E}, x_i, y_i, z_i) = ||\overline{\mathbf{Op}_i}||_{rad} \cdot (F(x_i, y_i, z_i)^{\frac{1}{2}} - 1)$$

In Eq. 6.2, $\mathbf{O}$ is the center of the ellipsoid. $||\overline{\mathbf{Op}_i}||_{rad}$ is the radial Euclidean distance [Bardinet98] between the ith point in the data set $\mathbf{p}_i$ and the intersection point of the line segment $\mathbf{Op}_i$ with the ellipsoid surface. The penalty factor

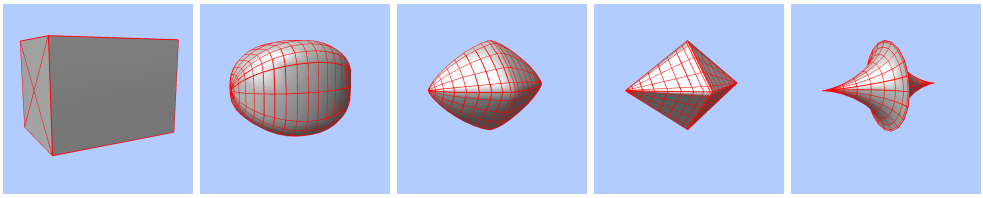


Figure 6.2: Different superquadric shapes obtained with different combinations of the roundness parameters ε_1 and ε_2.

$\sqrt{a_1 a_2 a_3}$ is included in order to prevent a shape primitive from growing excessively in one direction or uniformly in all directions.

6.3.2 Superquadrics

A superquadric is a closed curve defined as the solution of the implicit equation

$$F(x,y,z) = ((\frac{x}{a_1})^{\frac{2}{\varepsilon_2}} + (\frac{y}{a_2})^{\frac{2}{\varepsilon_2}})^{\frac{\varepsilon_2}{\varepsilon_1}} + (\frac{z}{a_3})^{\frac{2}{\varepsilon_1}} = 1 \tag{6.3}$$

In Eq. 6.3 a_1, a_2 and a_3 are the radii along the three main axes, and ε_1 and ε_2 are roundness parameters. The same inside-outside test based on F as for ellipsoids applies (Sect 6.3.1).

Depending on the roundness parameters, the shape of a superquadric shell mediates between circular and rectangular, enabling a variety of intermediate representations (see Fig. 6.2). Thus, it can approximate a large range of voxel set geometries at a high accuracy.

A superquadric in general position is thus described by 11 parameters $\mathcal{Q} = [a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, R_x, R_y, R_z, T_x, T_y, T_z]$. $(R_x, R_y, R_z, T_x, T_y, T_z)$ are the three rotation and translation parameters with respect to the world origin, a_1, a_2, a_3 are the radii along the major axes. Thus, in order to fit an approximating superquadric $\mathcal{Q}$ to a set of N 3D voxel centers, 11 shape parameters need to be determined. The optimal parameters are found by numerically minimizing an error function that measures the distance between the superquadric's surface and the volume elements.

The choice of a good error function is essential for the quality of the final fit. We have run experiments with several different distance measures and were most satisfied with the following one:

$$D = \frac{a_1 a_2 a_3}{N} \sum_{i=1}^{N} (F(x_i, y_i, z_i)^{\varepsilon_1} - 1)^2 \tag{6.4}$$

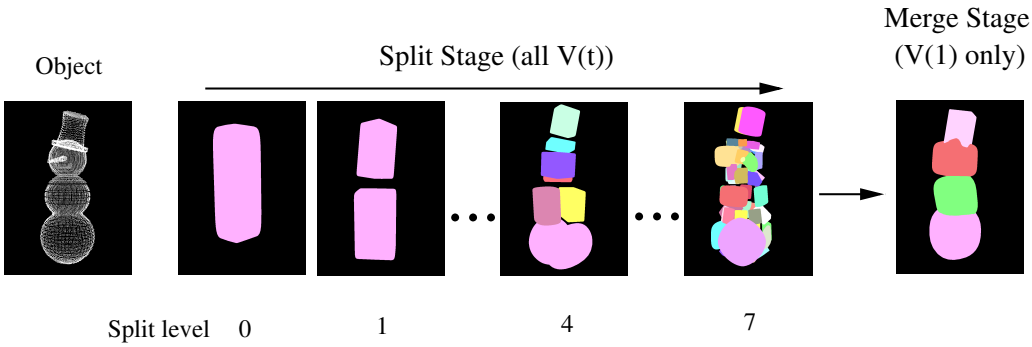


Figure 6.3: Illustration of the split and merge procedure for superquadric primitives using a synthetic data set of a snowman.

In Eq. 6.4 N is the number of voxels and $d(\mathscr{Q},x_i,y_i,z_i) = F(x_i,y_i,z_i)^{\varepsilon_1} - 1$ is an approximation to the distance of a volume element to the superquadric surface as proposed in [Leonardis97]. The factor $a_1a_2a_3$ is included in order to prevent the shape primitive from growing excessively in one direction or uniformly in all directions. In contrast to Eq. 6.2 this corresponds to a stronger penalization of excessively grown shapes.

We have evaluated several non-linear optimization schemes on test voxel sets to identify the most appropriate minimizer. We achieved the best results with the LBFGS-B method [Byrd95], which is a quasi-Newton algorithm that permits the specification of bound constraints on the parameters. Results with other numerical optimization schemes such as Amoeba (a downhill-simplex variant), Powell's method (a direction set method), and the often used Levenberg-Marquardt optimizer were significantly worse (see [Press02] for information on these methods). This is mainly due to the fact that these methods don't allow the specification of constraints in the parameters, and thus it may happen that irregular superquadrics with negative roundness parameters are produced. A good initial set of parameters to start the minimization with is found by fitting a regular ellipsoid to the voxel data (strictly speaking a regular ellipsoid is a superquadric with $\varepsilon_1 = \varepsilon_2 = 1$).

6.3.3 Split and Merge

Using the method described in Sect. 6.3 for each time step, we fill the voxel volumes with shape primitives such that their total number and fitting error are as small as possible. We achieve this by applying a hierarchical *split and merge* approach [Chevalier03]. The procedure starts with a split stage, approximating the whole voxel volume first with one $\mathscr{U}$, which is subdivided into two instances in case D is greater than some threshold (Fig. 6.3). The split stage recursively

processes each newly created approximator in the same way, thereby producing a hierarchical decomposition of the voxel set. The split stage is applied to each voxel volume $V(t)$ individually.

The merge stage follows the split stage and improves the fitting result by merging pairs of neighboring primitives into one. It is performed only for the voxel volume $V(1)$ of the first time step.

In the following the individual steps of both stages are detailed.

Split Stage

For each $V(t)$:

1 The whole set of 3D voxels $V(t)$ is approximated by one shape primitive $\mathscr{U}$.

2 If the fitting error D of $\mathscr{U}$ is less than some threshold T_{split}, the procedure stops. Otherwise, it proceeds to step 3.

3 The set of 3D voxels is split into two subsets S_1 and S_2 along the plane $\mathscr{P}$ orthogonal to the major axis of elongation of the voxel set (Note that $\mathscr{P}$ contains the centroid of the set).

4 S_1 and S_2 are approximated individually by one shape primitve each. For each subset, the procedure is repeated from step 2.

We obtain a set of primitives $U_{split}(t)$ and a set of corresponding voxel subsets $S_{split}(t)$ that approximate the voxel model $V(t)$. After a sufficient number of subdivisions (in our case typically 7), there is a high likelihood that all points in one voxel subset belong to the same rigid body of the tracked subject's kinematic skeleton. Nonetheless, it is still possible that more than one approximator is fitted to one rigid body (e.g. four ellipsoids to the upper arm), or that an ellipsoid was fitted to a position on the boundary between two adjacent rigid bodies (e.g. centered on the knee joint). In the latter case the voxel subset associated with the shape primitive would belong to two different kinematic elements.

Merge Stage

For $V(1)$ only:

1 For each subset of voxels $S_i \in S_{split}$, we determine the list $K_i = \{S_{n1}, .., S_{nk}\}$ of neighboring voxel subsets ($S_{n1}, .., S_{nk} \in S_{split}$).

2 For each possible pairing of the voxel set S_i and one neighboring voxel set $S_j \in K_i$, a merged voxel set M_j is created. A novel shape primitive is fitted to each M_j and a fitting error D_j is computed. From all paired primitives whose D_j is smaller than the sum of fitting errors of the approximators it was created from, the one with the lowest D_j is chosen to replace the two primitives it emerged from.

3 A new set of approximators is obtained. The procedure is repeated from step 1. It terminates when no further reduction of the fitting error is possible.

We perform the merging step only on the first voxel volume $V(1)$. If we were considering voxel volumes from different time steps independently and would merge approximators only due to structural criteria, it would not be possible to prevent erroneous merges across rigid body boundaries. The resulting set of shape primitives is the starting point for the matching step (Sect 6.4) which exploits the temporal dimension to prevent merging across boundaries of separate bodies.

The result of the split and merge process is a set of approximators $U(t)$ and a set of voxel subsets $S(t)$ for each $V(t)$.

6.4 Shape Primitive Matching

After subdividing each voxel volume using primitive shapes, a set of correspondences $C(t,t+1)$ between each pair of approximator sets $U(t)$ and $U(t+1)$ at subsequent time steps is computed. The set of correspondences describes for each shape primitive in $U(t)$ to which member of $U(t+1)$ it is related. In other words, the correspondences indicate from which 3D location at t to which position at $t+1$ a primitive shape moves.

The correspondence finding procedure processes each pair of sets $U(t)$ and $U(t+1)$ at subsequent time instants separately. It is important to note that the number of shape primitives in the sets $U(t)$ and $U(t+1)$ may differ. If we can reorganize the approximators such that their number at each time step is the same, the motion in space of each primitive from beginning to end of an input sequence can be estimated. We employ a two-stage procedure to establish the correspondences and to reorganize the approximators. This way we establish a bijective correspondence mapping between approximating shapes at subsequent time steps. Technically, the correspondences from t to $t+1$ are established by searching for correspondences from $t+1$ to t which are, in the end, inverted.

In the first stage, a correspondence for each individual shape primitive is established to an approximator at the preceding time instant by means of a dynamic programming approach [Sniedovich92]. The error function used in this optimization procedure is the Euclidean distance between the primitives' centers.

After the first stage, two cases of degenerate correspondences may occur that need to be corrected in a second stage in order to establish a bijective mapping.

The first case, the *unmatched shape primitive* (Fig. 6.4a), occurs if there exists an approximator $\mathscr{U}_1$ at time t to which no approximator from $t+1$ is connected. To solve this problem, the shape $\mathscr{U}_2 \in U(t+1)$ closest to $\mathscr{U}_1$ according to the Euclidean distance is selected. The voxel subset associated to $\mathscr{U}_2$ is split in two and two new approximators $\mathscr{U}_3$ and $\mathscr{U}_4$ are fitted to the newly created voxel subsets. $\mathscr{U}_3$ inherits the original correspondence to time t from $\mathscr{U}_2$, $\mathscr{U}_4$ establishes a new correspondence to $\mathscr{U}_1$.

The second case, the *multi-match* (Fig. 6.4b), arises if more than one shape primitive from $U(t+1)$ is assigned the same partner in $U(t)$. We solve this problem by merging all the approximators at $t+1$ which have been assigned to the same partner from t. This is achieved by merging all the associated voxel subsets and fitting a new shape primitive.

The two degenerate cases are corrected subsequently. After stage two of the correspondence finding, the correspondence directions are inverted. By this means, for each primitive in $U(t)$ exactly one partner from $U(t+1)$ is found.

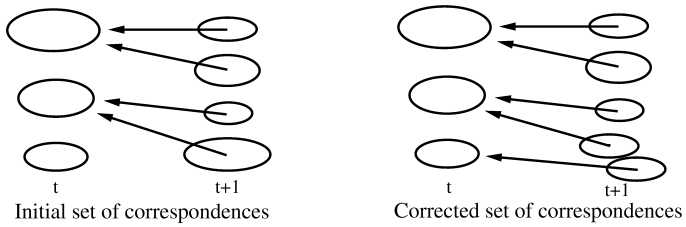
After all time steps have been processed in this way, each set of approximators contains the same number of shapes as the set $U(1)$. Note that in order to establish correct correspondences $C(t,t+1)$ the superquadric sets are modified as well. For each shape primitive in $U(1)$ a complete motion path for the whole sequence can be built by linking subsequent correspondences. The so-created set of paths P contains for each $\mathscr{Q}_i \in Q(1)$ a path P_i, P_i being an ordered set of 3D coordinates $P_i = \{(x_i(t), y_i(t), z_i(t)) \mid t \text{ valid time step}\}$ of the primitive shape's center at time t. Fig. 6.5a,b shows example paths of individual approximators.

6.5 Body Part Identification

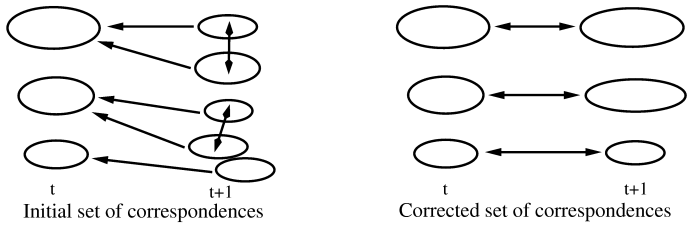
The paths of P provide all necessary information we need to identify separate rigid bodies in the kinematic skeleton of the moving subject. In case we are analyzing volume data of a human, this means that the paths enable us to identify, for example, the upper arm segment or the lower leg segment. Implicitly, we make the simplifying assumption that individual kinematic elements can be represented as rigid structures that do not undergo strong deformations.

In order to identify individual rigid bodies, we make use of the fact that the mutual Euclidean distance between any two points on the same body does not change while the skeleton is moving. Thus, if the mutual distance between the motion paths of two approximators over time is subject to significant variations, it is most likely that the two primitives do not lie inside the same rigid body.

This criterion gives us a procedure at hand which enables clustering individual



(a) Unmatched Shape Primitive



(b) Multi-match

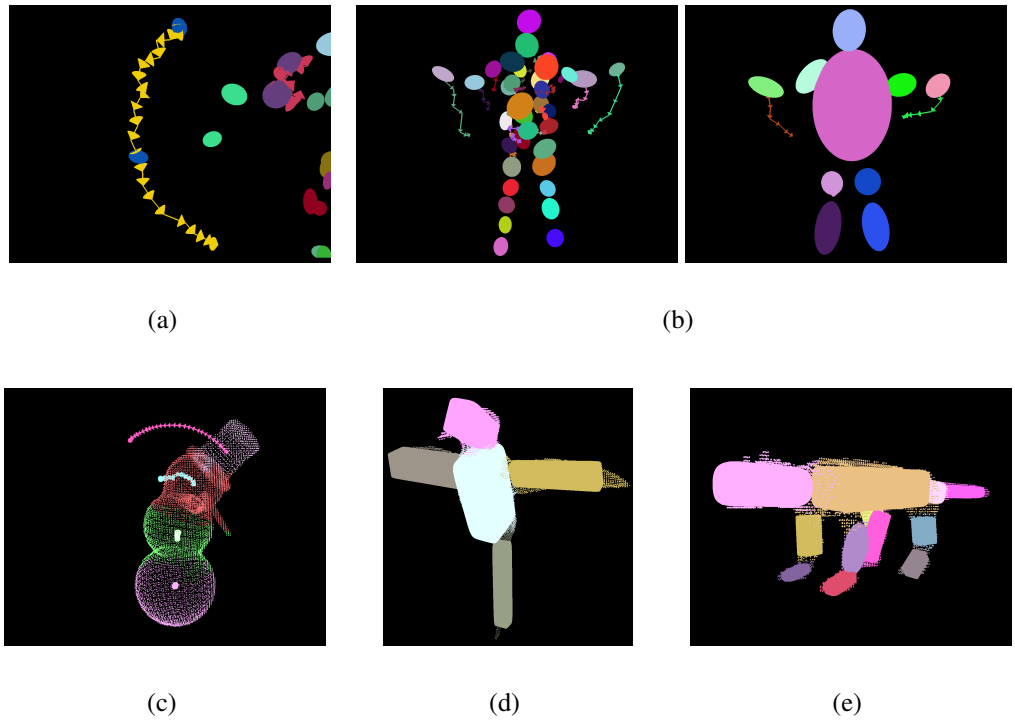
Figure 6.4: Handling of degenerate cases during correspondence finding.

shape primitives into separate kinematic elements of the whole body. We employ a voting-based test that analyzes the curve of Euclidean distances between the paths of the approximators over time. The value of the distance curve $d_{i,j}(t)$ between the paths of two primitives $\mathscr{U}_i \in U(1)$ and $\mathscr{U}_j \in U(1)$ at time t is defined as the Euclidean distance between their respective positions on the paths at t. In order to decide if $\mathscr{U}_i$ and $\mathscr{U}_j$ lie on the same rigid body we check for the presence of two features in the distance curves:

The first feature are those parts of the distance curve in which the absolute value of the first derivative is large. These parts indicate those time instants in which the individual rigid bodies possibly drift away from each other or move closer to each other. For each t at which $\mid d'_{i,j}(t) \mid > T_{deriv}$, T_{deriv} being a derivative threshold, a voting counter $vc(i,j)_{deriv}$ is increased by one.

The second feature arises at every time step for which the value of the distance curve differs by more than a threshold from the initial distance value $d_{i,j}(1)$. Thus, for each t with $\mid d_{i,j}(t) - d_{i,j}(1) \mid > T_{diff}$, T_{diff} being a difference threshold, a second voting counter $vc(i,j)_{diff}$ is increased by one.

The final vote $vc(i,j)$ is the sum of the two previously mentioned voting counters $vc(i,j) = vc(i,j)_{deriv} + vc(i,j)_{diff}$. If this final vote is larger than a threshold T_{vote}, the distance curve fails the test and the approximators are considered to



(a) (b)

(c) (d) (e)

Figure 6.5: (a) Rendered motion paths of two ellipsoidal approximators (radii reduced for better visibility) in the forearm and upper arm of a human test subject respectively (a). (b) Motion paths of ellipsoids (left) and detected rigid bodies (right) shown as larger ellipsoid in the human body. (c) Motion of superquadrics that were fitted to the snowman model. Rigid bodies detected in the synthetic bird (d) and monster (e) data sets.

reside on different rigid bodies.

To eliminate spurious peaks in a distance curve due to noise, a median filter is applied to it before applying the distance criterion. By means of our voting-based scheme and appropriate thresholds (found through experiments) it is possible to perform robust path comparison even in the presence of measurement noise. We apply the voting-based test to classify individual rigid bodies as follows:

1 A seed primitive $\mathscr{U}_{seed} \in U(1)$ is selected and a distance curve $d_{seed,k}$ with each superquadric $\mathscr{U}_k \in U(1) \setminus \{\mathscr{U}_{seed}\}$ is computed.

2 For each $\mathscr{U}_k$ the voting-based test is applied to $d_{seed,k}$, and $\mathscr{U}_k$ is classified as lying on the same rigid body if the test succeeds.

3 The procedure iterates by restarting from step 1 and selecting a new seed from all approximators that have not yet been assigned to a rigid body.

The seed $\mathscr{Q}_{seed}$ in the first iteration is the primitive nearest to the center of gravity (COG) of the voxel set $V(1)$. In the subsequent iterations, the selected seed is the primitive nearest to the COG of the body part that was found in the preceding iteration. This seed selection criterion is a heuristics which enables the construction of a hierarchy of rigid bodies in the moving character. The rigid body detected first is considered to be the root of the skeleton hierarchy. Each subsequently detected rigid body is considered to be on the next lower hierarchy level, and to be connected to the root. The whole classification procedure is recursively applied to each individual rigid body on the next lower hierarchy level, thereby further refining the set of detected body parts.

In case of a human subject this strategy leads to the identification of one rigid body for the torso and one for each arm, each leg and the head in the first iteration. Now the procedure is repeated for each limb which produces the final correct subdivision into body parts.

For each $V(t)$ it is now known which voxel subsets form a rigid body and how the rigid bodies move over time. Figs. 6.5b,d,e show rigid bodies that were found in some of our test data.

6.6 Skeleton Reconstruction

In the final step we use the detected rigid bodies and their motion to estimate the 3D locations of joints in the skeleton hierarchy. For estimation of the joint locations one has to agree on a reference time instant. For the pose that the subject struck in this time instant, the skeleton structure is estimated. Usually we regard the body model reconstructed for the first time instant as the reference model. The rigid body hierarchy, and thus the information which rigid bodies are connected, has already been determined in the Body Part Identification step (Sect. 6.5). We assume that all discovered joints provide three rotational degrees of freedom.

The goal is to identify a joint location for each pair of adjacent rigid bodies B_a and B_b that are connected. We have tested two different methods to achieve this goal.

The first method, *skeleton reconstruction 1* or SR1, estimates a joint location based on the boundary voxels between the voxel subsets that are associated to B_a and B_b. The boundary voxel set contains all those volume elements from B_a that have at least one neighbor from B_b, and all voxels from B_b that have at least one adjacent voxel from B_a. The joint position is at the center of gravity of the boundary voxel set.

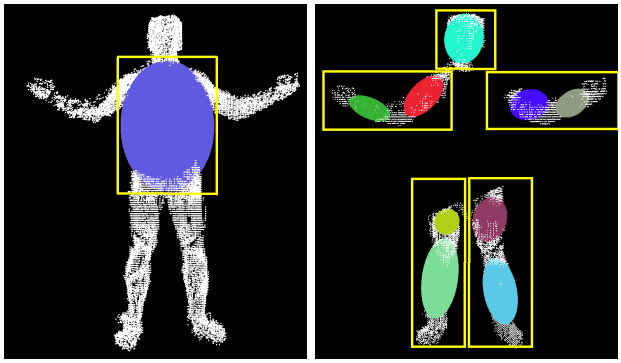


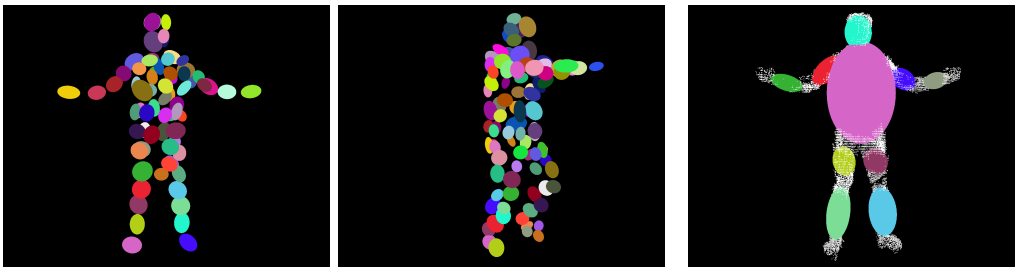
Figure 6.6: Optional iterative model estimation: After the first iteration the torso voxels (left) are identified and eliminated from each voxel set. The split and merge, the shape primitive matching, and the body part identification steps are then recursively applied to each remaining isolated voxel subset (right).

The second method, *skeleton reconstruction 2* or SR2, performs a simple collision test between approximators that have been fitted to B_a and B_b. Rays are shot along the three major axes of both neighboring primitives. The rays originate from the center of an approximator. An intersection test is performed between each ray (positive and negative direction) and the surface of the respective other shape primitive. From all intersecting rays the one with the smallest distance between ray origin and intersection point is found. The intersection point of this ray is considered to be the joint position.

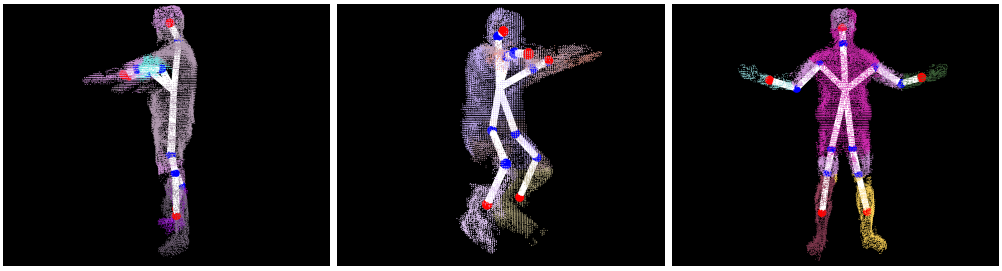
The primary goal of our approach is to reconstruct a kinematic skeleton model. Nonetheless, since we are able to build such a model for each time step of a motion sequence, approximate motion tracking of the moving subject is also feasible. The application of our joint localization scheme to each time step of video is no tracking in a strict sense since the skeleton models may differ. However, it is still possible to obtain a preliminary estimate of the motion parameters.

6.7 Results and Discussion

We have evaluated the performance of our system using synthetic and real data sets. The real input data were recorded in our multi-view video studio and show a person that performs simple gymnastic moves such as knee bends. Volume representations for each time step of video were obtained using space carving (Sect. 6.2). The volume data were carved out of blocks of 256^3 volume elements. In the case of the real input data, one volume consists of approximately 22000



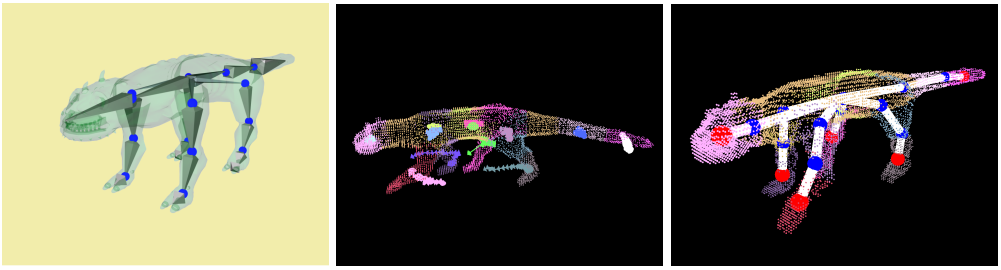
(a) (b)



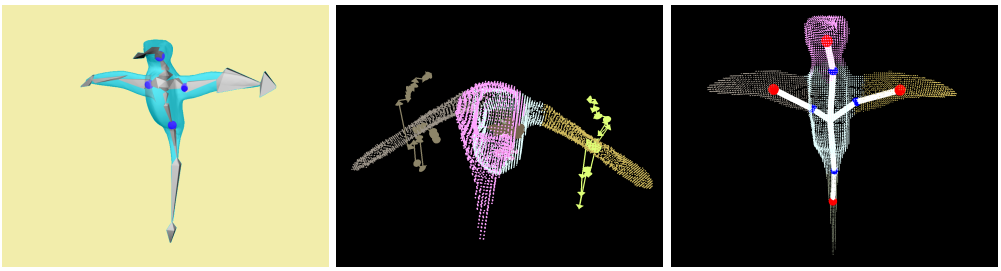
(c)

(d)

Figure 6.7: (a) Ellipsoids fitted to different body poses. (b) Discovered rigid bodies rendered as ellipsoidal shells inside the voxel volumes. (c) Skeletons estimated for different stances using SR1. Voxel sets belonging to different rigid bodies are drawn in different colors. (d) Input voxel set, discovered rigid bodies, and reconstructed skeleton if the person only moves the upper extremities. In this experiment superquadrics and SR2 were applied.



(a)



(b)

Figure 6.8: (a) Left to right: Monster with 3D Studio skeleton, animated joints are shown as spheres; motion of individual body parts; estimated skeleton, joints are shown as (blue) spheres, bones are shown in white. (b) Left to right: Bird with 3D Studio skeleton; estimated body parts and their motion; reconstructed skeleton.

voxels.

In Fig. 6.7a individual voxel volumes that have been approximated with ellipsoidal primitives are depicted. Fig. 6.7b shows individual rigid bodies that have been identified and to which ellipsoidal shells have been fitted for better visualization. In Fig. 6.7c different human skeletons are shown that have been reconstructed from a gymnastics sequence. Our method faithfully reconstructs body models for different body poses. The bone and joint layout comply with the skeleton of the real human.

In Fig. 6.7d some results are shown that we obtained after processing a motion sequence of around 40 frames in which the person only moves the upper extremities, i.e. the arms, and the head. For these tests we employed superquadric primitives. These images nicely illustrate the working principle of our approach. Since the lower extremities were not moving at all they could not be identified as

separate kinematic chains and were therefore considered to be part of the torso. Little inaccuracies in the detected locations of the elbow joints can be observed. This is mainly due to the fact that the sequence is very short and that the person wears comparably wide clothes. It is obvious that the algorithm can only discriminate two separate bodies if at any point there is a noticeable relative motion between them. In contrast to previous methods from the literature we can identify individual bodies, no matter at what point in time this relative motion was observed.

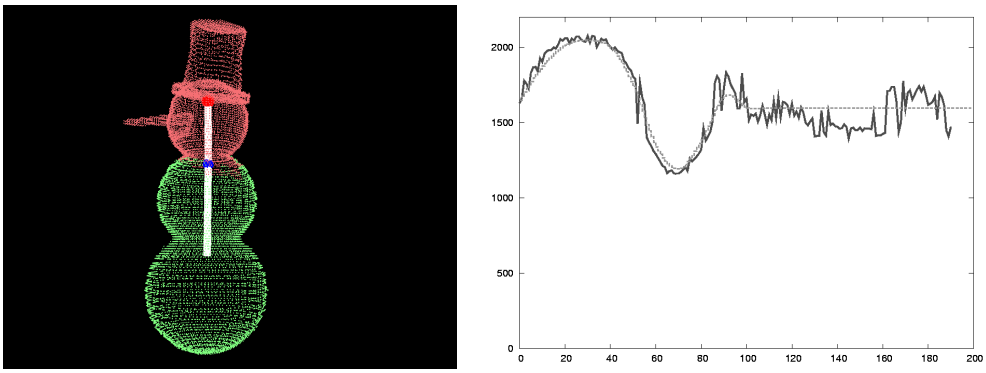
Our results demonstrate that our method performs well despite the presence of noise in the volume data. Although the space carving approach eliminates most of the typical visibility artifacts in shape-from-silhouette volumes, bulky arms and legs still occur sometimes.

We found out that an iterative implementation of our algorithm in which the steps 1-3 (see Fig. 6.1) are repeated, is beneficial if the noise level in the data is high. After each iteration, the largest rigid body is identified and, before the next iteration, all voxels belonging to this rigid body are eliminated from all volume data sets $V(t)$. Subsequently, steps 1-3 are applied in the same way to each newly found isolated voxel set. In the case of a human subject, this means that the first iteration identifies the torso segment, and in subsequent iterations, the algorithm proceeds with the arms, the legs and the head. The working principle of the iterative version of our algorithm is shown in Fig. 6.6.

The synthetic data sets we used were the moving snowman (about 8000 voxels per time step), the bird (about 11000 voxels per time step), and the monster (about 14000 voxels per time step). Animated voxel sequences with these models were created in 3D Studio Max™ using hand-crafted skeletons. One major advantage of the synthetic data is that the ground truth skeleton structure is available for comparison.

The bird sequence is a study of the wing beat, as well as the tail and the head motion of a bird in flight. We animated 4 joints of a kinematic skeleton, one at the neck, one at the tail and two at the roots of the wings. Fig. 6.8a shows the body structure that we estimated using superquadrics and SR1. The skeleton nicely complies with the ground truth kinematic model that we used for the animation.

Our most complex data set was the monster, a lizard-like four-legged creature. In total, we used 15 joints for animating its body, 2 in the tail, 3 in each leg and 1 at the neck. We created a walking animation in which the monster imitates the walking style of iguanas. While moving, the lizard's head as well as its tail slightly oscillate from left to right. The skeleton of the creature that we estimated is shown in Fig. 6.8b. The motion paths of individual rigid bodies over time are depicted as well. The kinematic structure of the head, the tail and the legs has been learned correctly. However, it was hard to identify the feet as separate rigid



(a) (b)

Figure 6.9: (a) Snowman skeleton. (b) Plot of reconstructed (dashed) against ground truth y-coordinate of one joint in the snowman for a sequence of 200 frames. On the abscissa the time is plotted, the ordinate is the y-coordinate in internal dimensionless size units (bounding box size of snowman=2000).

bodies since their motion relative to the lower leg was only very marginal.

The simplest synthetic data set was the snowman sequence. The snowman model was animated using one point of articulation at the neck. The derived skeleton and the correctly detected two body parts are shown in Fig. 6.9a. Fig. 6.9b shows a plot in which one coordinate of the neck joint position is plotted against the corresponding coordinate of the ground truth joint location, and skeleton reconstruction is performed for each time step individually. With the exception of some outliers, the coordinate difference is small (mostly below 2% with respect to the maximal bounding box side length, 5% in the worst case).

The timings for the individual system components and different algorithmic options are summarized in Table 6.1. We measured the runtimes on a Pentium IV 3GHz.

The time ranges represent the lower and upper bounds of average runtimes that we measured when analyzing our synthetic and real test data. For evaluating the split and merge performance we have exhaustively subdivided the volume up to level 7. Due to the simpler fitting method, the split and merge for ellipsoidal shapes is significantly faster than for superquadric shapes. The execution speed of the superquadric split and merge steps are dominated by the runtime of the LBFGS-B optimizer. Typically, in order to obtain a similar shape approximation quality as with superquadrics, many more subdivision levels with ellipsoidal approximators are necessary. Depending on the data set, the longer runtimes of superquadric fitting may be justified.

split using ellipsoids (7 levels)	2-4 s (per time step)
merge using ellipsoids	9-13 s (per time step)
split using superquadrics (LBFGS-B, 7 levels)	120-160 s (per time step)
merge using superquadrics (LBFGS-B)	250-1000 s (first time step)
Correspondence Finding	7-19 s (per time step)
Body part identification	3-6 s (per time step)
Skeleton reconstruction 1 (SR1)	0.3-0.5 s (per time step)
Skeleton reconstruction 2 (SR2)	0.3-0.4 s (per time step)

Table 6.1: Measured runtimes of individual system components.

The execution speeds of the two alternative methods we use to infer the skeleton structure after the body parts have been identified are almost identical. Nonetheless, we found that SR1 usually estimates joint positions more accurately than SR2. Furthermore, the performance of SR2 depends more strongly on how accurately shape primitives have been fitted to the volumes (in particular in terms of correct orientation). For SR1 this dependency is not as strong making it more robust against measurement noise.

The proposed body model estimation algorithm is subject to a couple of limitations. Even though we don't prescribe an initialization motion, two different adjacent rigid body segments can only be discriminated if at least once in a sequence a relative motion between them can be observed. We consider this a principal problem of a non-informed motion analysis approach and not a limitation that is specific to our method. Furthermore, we expect that the system's performance will deteriorate if voxels of individual rigid bodies merge frequently with the rest of the volume (e.g. if the arms are often kept tight to the torso).

Even though, our results show that our method is capable of inferring the kinematic structure of arbitrary moving subjects at a high accuracy. Our method achieves this without resorting to optical markings and without relying on a priori structural information about the observed subject.

Part II

Capturing Appearance and Motion - Free-Viewpoint Video

Chapter 7

Free-Viewpoint Video - Problem Statement and Preliminaries

The field of computer graphics has always been guided by the aim to develop algorithms that enable the photo-realistic rendition of real world scenes in a computer. One possibility to fool the eye of the beholder is to design geometric scene and computational lighting models from scratch that resemble the real thing as closely as possible. Another possibility that has gained much attention recently is to reconstruct computational models from video footage that was acquired in the real world. The ongoing technical advancement in computer and imaging sensor technology has rendered this novel paradigm of computer graphics feasible. If a visually pleasing computational scene model is to be derived from image data, geometry, appearance and reflectance models have to be automatically inferred.

This new video-based paradigm in Computer Graphics has paved the trail for an exciting novel field of research that aims at lifting the traditional two-dimensional medium video onto a novel three-dimensional immersive level. The general idea is to reconstruct three-dimensional video content from two-dimensional video data of a real world scene. This novel video format allows a viewer to look at a scene from a novel viewpoint that no physical camera has actually recorded.

The term 3D video subsumes many technological approaches that vary in the range of possible novel viewpoints they can generate, in the kind of display technology they employ, or the kind of scene model they reconstruct. Free-viewpoint video is one 3D video category in which the viewer is given the greatest freedom. There, he is allowed to change his viewpoint to an arbitrary novel position in virtual 3D space.

The number of possible applications for free viewpoint video technology is enormous. While in today's feature films the viewer is bound to follow the camera path plotted by the director, the viewer of a 3D video can create his own personal camera flight. TV broadcasts of sports events will gain a new dimension by providing sports enthusiasts with new forms of visualization. For instance, The motion of the basketball player jumping to the basket could be frozen and a virtual camera flight around the scene could be created. The motion of a track and field athlete could be shown from an arbitrary novel perspective that has not been seen by any physical camera.

Human actors are the central elements of motion pictures. Over millions of years the human eye has been trained to notice even the slightest inaccuracies in visual appearance and movements of virtual actors. It is thus a great challenge to create a visually convincing virtual copy of real world person within a 3D video.

In this part of the thesis we demonstrate how it is possible to reconstruct and render high quality free-viewpoint videos of human actors. We base our approach on the principals of marker-free optical motion capture that have been detailed in part I.

In Chap. 8 we describe the algorithmic components of a model-based system for the reconstruction and rendering of free-viewpoint videos of human actors. The inputs to our method are multi-view video streams that we recorded in our acquisition room (Chap. 4). We employ a marker-free model based optical human motion capture approach to estimate the motion of a person. The capturing method analyzes the overlap between the reprojected virtual model and the silhouettes of the person in each camera view in order to determine the optimal pose parameters at each time step of video. During playback of the 3D video the model is rendered in the sequence of acquired body poses and realistic surface textures are generated by blending and reprojecting the input video images onto the model. This way a realistic time-varying surface appearance of the virtual actor is achieved that captures even subtle details, such as wrinkles in clothing. While the dynamic 3D scene is rendered the viewer can interactively change its viewpoint to an arbitrary position in space.

In Chap. 9 we present an extended version of the original free-viewpoint video system in which we add a new processing step to the motion capture approach. While a purely silhouette-based capturing method can robustly estimate human motion on a large scale, slight pose inaccuracies, e.g. for the head, may still occur. We solve this problem by incorporating texture information into the pose estimation process. We compute corrective 3D flow fields from 2D optical flows that enable us to update the stance estimated in the silhouette step. This way we achieve a multi-view texture consistent posture of the body model for each time step.

The methods described in Chap. 4 and Chap. 9 enable the photo-realistic ren-

dition of real world scenes from arbitrary novel viewpoints. However, the visual appearance is only captured under the illumination conditions that prevailed in the studio while the scene was recorded. In Chap. 10 we present an extension of our original approach that allows us to not only capture the motion but also dynamic surface reflectance of a person from multi-view video streams. The reflectance model we estimate consists of a per-texture element parametric BRDF model and time-varying normal maps. With the so-created relightable free-viewpoint videos any virtual environment can be augmented with three-dimensional dynamic scene content that has been captured in the real world.

In the remainder of this chapter we will briefly review some work from the literature that is related to our free-viewpoint video approach.

7.1 Related Work

In the computer vision and computer graphics literature, many conceptually different approaches to scene reconstruction from image data and novel view synthesis have been proposed. They greatly differ in terms of functional and performance criteria, e.g. the range of novel viewpoints that can be generated, their run-times, or the employed type of image data. For this review, however, we find it more instructive to categorized the related work based on the employed general principle of scene reconstruction.

We begin with methods that take a purely image-based approach to scene reconstruction (Sect. 7.1.1). Thereafter we present methods that additionally reconstruct geometric scene models to generate novel views and compare them to the former class of methods (Sect. 7.1.2). After this, we briefly review approaches that aim at handling the whole pipeline ranging from acquisition to rendering in real-time (Sect. 7.1.3). Our research adds an additional dimension to the idea of 3D video by enabling us to reconstruct also surface reflectance models from multi-view video. We therefore also briefly review related work on image-based reflectance measurement (Sect. 7.1.4). The algorithmic solutions presented in part II of this thesis also capitalizes on the related work on marker-free video-based human motion capture which is discussed in Chap. 3

7.1.1 Purely Image-based Novel View Synthesis

Image-based rendering (IBR) techniques are a class of algorithms in computer graphics that do not employ explicit geometry models to render novel views but rather generate them from a collection of images [Kang00].

One type of IBR methods is based on the concept of the 7D plenoptic function [Adelson91] that describes the intensity of light rays passing through each

point in space, at every possible angle, and for every wavelength at every time. This function is far too complex to be reconstructed in its full complexity. Furthermore its storage requirements make it totally inappropriate for novel viewpoint rendering. Thus researchers have dropped some of the variables to reduce its complexity. In [McMillan95], the time and wavelength variable were dropped, and the obtained 5D function used for view synthesis. The simplest plenoptic function is a 2D panorama, where the viewpoint is fixed and only the direction can be changed [Szeliski97]. The light-field [Levoy96] and lumigraph [Gortler96] approaches make use of the fact that, if one stays out of the convex hull of an object, the 5D plenoptic function can be reduced to 4D. Typically, this lower-dimensional function is parameterized by two parallel planes of the object's bounding box. An alternative form of light field parameterization that also enables the realistic reconstruction of depth of field effects with a changing virtual camera focal plane is described in [Isaksen00]. In [Aliaga01] a light-field based approach that can create interactive walkthroughs of virtual environments is presented. It stitches together plenoptic functions that have been reconstructed from images of a moving panoramic camera. Important issues in all purely images-based approaches are the proper selection of image samples and the reconstruction from them [Chai00].

Another class of approaches makes use of simple implicit geometry information that is not directly available. In view interpolation, arbitrary novel views can be synthesized from reference images. However, the results will only be pleasing if the reference views are close to the novel virtual view. The method by Chen and Williams [Chen93] computes a dense optical flow field between reference images and from this creates intermediate views. A related approach is the view morphing algorithm presented in [Seitz96]. Intermediate views along the line linking to cameras views can be generated. Other approaches make use of the trifocal tensor [Hartley00] that mathematically describes the relation between three camera views to generate novel views from two reference images [Avidan97].

Some IBR methods make use of explicit 3D scene information, either in the form of per-pixel depth information or 3D coordinates. In 3D warping, per-pixel depth data are used to generate novel views of an image from nearby camera positions. To achieve this, the image data are back-projected to their correct 3D locations and then projected into the virtual camera view [McMillan97]. The problem with 3D warping is that due to insufficient visibility, occlusion and disocclusion artifacts arise if the novel view is too different from the reference view. A solution to this problem is provided by layered depth images which store not only one depth value per pixel but multiple depth values for each occluded surface [Shade98].

7.1.2 Novel View Synthesis via Image-based Geometry Reconstruction

The algorithms presented in this section exploit information on scene geometry that is automatically inferred from input video streams. This has two important advantages over purely image-based approaches. First, the number of recording cameras can be significantly reduced. Second, despite the lower number of recording imaging sensors, a large range of novel virtual viewpoints can be synthesized.

One prominent type of approaches employs stereo reconstruction from multiple video streams to estimate a 3D model of the recorded scene. In [Narayanan98], an approach is presented that can reconstruct dynamic 3D scene geometry using dense stereo reconstruction algorithm. For recording, a hemispherical dome of 51 cameras is used. In [Zitnick04] a novel system is presented that reconstructs dynamic 3D scenes using stereo. The results in their paper were generated with a prototype system that features a few cameras in a closely spaced semi-circular arrangement. The authors present a novel algorithm to reconstruct low-noise depth maps and address the problem of ghosting artifacts at the silhouette boundaries of objects in the scene foreground. Stereo-based algorithms have been very popular in telepresence applications. The geometry of the person at one side of the communication channel is reconstructed and transferred to the other end where it is rendered [Mulligan00]. In stereo-based approaches no a-priori knowledge about the scene geometry is required. Nonetheless, several other factors limit their applicability. In the first place, stereo methods can only robustly reconstruct diffuse objects. Secondly, novel viewpoints onto the scene only look plausible if they are close to the input camera views. In consequence, full virtual flyarounds are only feasible if a very high number of tightly packed cameras are used for recording.

A second very popular category of approaches are algorithms that reconstruct scene geometry from multi-view silhouette images, so-called shape-from-silhouette methods. The overall idea is to reproject silhouette cones from the camera positions into the scene and to intersect them. This way, the so-called visual hull [Laurentini94] of the object in the scene is obtained (see Chap. 5 Sect. 5.5). In [Moezzi97] a system is described that reconstructs voxel-based visual hulls of dynamic scenes from multi-view video. It is also possible to reconstruct textured polyhedral models from multi-view silhouette footage and render them in real-time from any arbitrary novel viewpoint [Matsuyama02, Matusik01]. In [Matusik00, Wuermlin02] an alternative approach to visual hull reconstruction is presented that does not explicitly estimate 3D geometry but reconstructs a novel output view from multiple video streams using only image-space constraints. The method presented in [Gross03] uses point primitives that are a generalization of 2D pixels into 3D for reconstructing dynamic scenes from video streams.

Shape-from silhouette approaches are subject to a couple of limitations. The

visual hull is only a coarse approximation to the true shape of an object. Concavities in the surface cannot be faithfully reconstructed by only employing silhouette information. Thus the renditions of the reconstructed scenes wil often look unnatural, in particular if texture information from the images is reprojected to incorrect geometry. Furthermore, if only few input cameras are available, the undersampling artifacts lend the shape-from-silhouette geometry an awkward facetted look.

Several methods have been proposed to overcome these limitations. In space-carving [Kutulakos00] a multi-view color consistency criterion is used to improve the visual hull geometry (see also Chap. 6 Sect. 6.2). It has also been shown that the employment of view-dependent opacity maps that control the transparency of the visual hull geometry can significantly improve the natural look of the approximate shape [Matusik02]. A hybrid method that improves the visual hull geometry by additionally considering stereo information is presented in [Li02].

An approach that renders smooth transitions between shape-from silhouette volumes available at discrete time steps is presented in [Vedula02]. The authors achieve this by computing 3D optical flows between subsequent volume models and rendering the views with a spatio-temporal ray-casting approach.

In contrast we propose in part II of this thesis a model-based approach to free-viewpoint video reconstruction. The commitment to a generic geometry model enables us to generate highly realistic novel viewpoint renditions of moving human actors although we only apply a handful of recording video cameras. Furthermore, the complete dynamic scene description consisting of the time-varying geometry and the dynamic surfaces textures can be stored in a very compact format.

7.1.3 Scene Recording and Novel Viewpoint Rendering in Real-time

Trying to handle the full pipeline from scene recording to interactive scene rendition in real-time is a very demanding undertaking. The time-constraint renders complicated reconstruction algorithms inappropriate. In consequence, compromises with respect to rendering quality and freedom of interaction with the content are inevitable. Although many ideas stem from the two previously reviewed categories, we find it instructive to consider real-time systems separately.

It has been shown in several papers that shape-from silhouette approaches, such as visual hull reconstruction, can perform in real-time [Matsuyama02, Matusik01, Li04b, Li04a]. However, all these approaches can only reconstruct the dynamic geometry of selected objects in the scene foreground. In recent years, the term 3D TV has emerged for systems that reconstruct and display 3D content of

entire scenes in real-time. Today, different researchers have different understandings of what 3D TV is supposed to achieve.

An approach which extends the existing 2D TV infrastructure with 3D information is described in the ATTEST project by the European Union [Redert02]. Here, the idea is to transmit per-pixel depth data in addition to the actual 2D TV images. On the receiver side, one can employ these data to render a stereoscopic view of the scene, which, in combination with an appropriate stereoscopic display device, provides a depth impression to the viewer [Fehn04].

In [Matusik04] a slightly different prototype of a 3D TV system is presented. It records a scene with multiple cameras, and transmits the video streams over a network to multiple projectors that illuminate a specially coated projection screen. The screen features a micro-structure that assures that the image projected by one projector is only reflected in a narrow angle in space. If the viewer laterally changes his viewpoint he always sees a 3D impression that is reconstructed from those cameras that are closest to his virtual viewpoint in the real scene. Currently, the viewer is allowed to move his head in a very limited range parallel to the screen since only this range is covered by the input cameras. Within these limits nice rendering results with a head-motion dependent depth parallax are obtained. A light-field algorithm is used for display.

So far, real-time approaches can, in terms of rendering quality and freedom of viewpoint change, not compete with most non-real-time algorithms.

7.1.4 Image-based Reflectance Estimation and Photometric Shape Reconstruction

For realistic rendering of the appearance of real world models under arbitrary lighting conditions, a mathematical model of the light interaction at the surface of the object is required. Typically, the physics of light interaction is described by means of a bi-directional reflectance distribution function or BRDF (see Chap. 2). For a detailed review on different models of light interaction we would like to refer the interested reader to [Rusinkiewicz00] or [Lensch04].

In image-based reflectance estimation, the camera serves as a measuring sensor that samples the radiance outgoing from an object's surface under different illumination situations. Many systems that follow this line of thinking have been proposed in the literature. Typically, a single point light source is used to illuminate an object of known 3D geometry. One common approach is to take HDR images of a curved object, yielding a different incident and outgoing directions per pixel and thus capturing a vast number of reflectance samples in parallel. Quite often the parameters of an analytic BRDF model are fitted to the measured data [Sato97, Lensch03] or a data-driven model [Matusik03]

is used. Reflectance measurements of scenes with more complex incident illumination can be derived by either a full-blown inverse global illumination approach [Yu99, Gibson01, Boivin01] or by representing the incident light field as an environment map and solving for the direct illumination component only [Yu98, Ramamoorthi01, Nishino01].

Instead of explicitly reconstructing a mathematical reflectance model it has also been tried to take an image-based approach to relighting. In [Hawkins04] a method to generate animatable and relightable face models from images taken with a special light stage is described. Using deformable geometry, the face is rendered under novel illumination by reconstructing from a large database of images that show the face under different incident illumination directions, different viewing directions and with different expressions. For our 3D video scenario, we prefer a more compact scene description based on parametric BRDFs that can be reconstructed in a fairly simple acquisition facility.

Reflection properties together with measured photometric data can also be used to derive geometric information of the original object [Zhang99]. Rushmeier et al. estimate diffuse albedo and normal map from photographs with varied incident light directions [Rushmeier97, Bernardini01]. A linear light source is employed by Gardner et al. [Gardner03] to estimate BRDF properties and surface normal. In [Georghiades03, Goldman04], reflectance and shape of static scenes are simultaneously refined using a single light source in each photograph.

Carceroni and Kutulakos present a volumetric method for simultaneous motion and reflectance capture for non-rigid objects [Carceroni01].

In contrast, in Chap. 10 we propose a model-based approach that captures shape, motion parameters and dynamic reflectance of the whole human body at high accuracy. In our approach we will approximate the incident illumination by multiple point light sources and estimate BRDF model parameters taking only direct illumination into account. We have also developed a photometric stereo method that reconstructs time-varying changes in surface geometry from reflectance samples.

Chapter 8

Model-based Free-Viewpoint Video of Human Actors

This chapter presents a novel model-based algorithm for reconstructing and rendering free-viewpoint videos of human actors. It applies synchronized multi-view video footage of an actor's performance to estimate motion parameters and to interactively re-render the actor's appearance from any viewpoint [Carranza03, Theobalt03b, Theobalt04b]. We achieve this by employing a model-based analysis-by-synthesis method.
The actor's silhouettes are extracted from synchronized video frames via background segmentation and then used to determine a sequence of poses for a 3D human body model. Prior to motion estimation, the body model is automatically adapted in shape and proportions to its real world counterpart. By employing multi-view texturing during rendering, time-dependent changes in the body surface are reproduced in high detail. The motion capture subsystem runs offline, is non-intrusive, yields robust motion parameter estimates, and can cope with a broad range of motion. The rendering subsystem runs at real-time frame rates using ubiquitous graphics hardware, yielding a highly naturalistic impression of the actor.

Our model-based free-viewpoint video approach has many advantages over 3D video approaches that explicitly extract the scene geometry from the video footage:

- Since the type of object in the scene is known in advance we can design a generic model and use a powerful marker-free motion capture approach to estimate its movements.

- We achieve highly convincing novel-viewpoint renderings even though free-viewpoint videos are only reconstructed with a handful of cameras.

- High-quality rendering is possible even with slightly inexact geometry.
- The temporal change in scene geometry can be parameterized by only 35 pose parameters. Thus, our free-viewpoint video format is ideal for transmission over bandwidth-limited network channels.
- The number of input cameras necessary is very low.
- Our model-based analysis-by-synthesis approach exploits graphics hardware, and thus solves a computer vision problem by means of computer graphics technology.
- The hierarchical structure of the motion estimation problem can be exploited to efficiently solve sub-problems in parallel.

8.1 Overview

In Fig. 8.1 the interplay of the algorithmic ingredients in our model-based free-viewpoint video system is shown. Inputs to our system are multi-view video streams that are recorded in our acquisition studio (Sect. 8.2). The video frames are postprocessed in order to segment the person in the foreground from the scene background by means of a color-based background subtraction (Chap. 2). We represent the 3D dynamic scene content with a generic human body model that can be adapted in shape and proportions to the dimensions of its real world counterpart (Sect. 8.3). For estimating the time-varying appearance of the actor in a scene we employ a model-based analysis-by-synthesis scheme. The principle clue that we use to fit the model to the scene content is the overlap between the image silhouettes and the silhouettes of the projected model in each camera view (Sect. 8.4). We transform this criterion into a numerical error function which is efficiently evaluated in graphics hardware. Using multiple camera views of the actor standing in an initialization pose, the geometry of the body model as well as as its skeleton dimensions are automatically customized by means of a numerical minimization in shape parameter space (Sect. 8.5). The shape parameters of the model remain fixed throughout the whole 3D video sequence. The central component of our analysis-by-synthesis scheme is a silhouette-based marker-free motion capture approach (Sect. 8.6). For each time step of video it performs an optimization search for an optimal set of pose parameters for the model. The energy function guiding this search is the previously mentioned silhouette-overlap. The hierarchical structure of the human body makes the pose determination problem a compartmentalized one, i.e. individual sub-problems can be solved independently from

each other. We profit from this fact and exploit this parallelism in both silhouette-match computation (Sect. 8.7.1) and pose parameter search (Sect. 8.7.2). Scene recording, model initialization and pose estimation run offline.

During playback of the reconstructed 3D video, the model is displayed in the sequence of captured body poses. We create a photo-realistic surface appearance of the model by projectively texturing it with the input video frames (Sect. 8.8). In order to generate an artifact-free surface appearance despite only approximate model geometry, we have developed a special texture blending (Sect. 8.8.1) and visibility pre-processing scheme (Sect. 8.8.2).

Our free-viewpoint video renderer provides the actual interface to the viewer. It enables him to freely navigate through 3D space while the dynamic scene content plays back in real-time. In Sect. 8.9 we demonstrate the very high visual quality that we can achieve with our algorithm for even as complex scenes as human ballet dance.

8.2 Input Data Acquisition

The video sequences used as inputs to our system are recorded in our multi-view camera studio (Chap. 4). To generate the results presented in this chapter we ap-

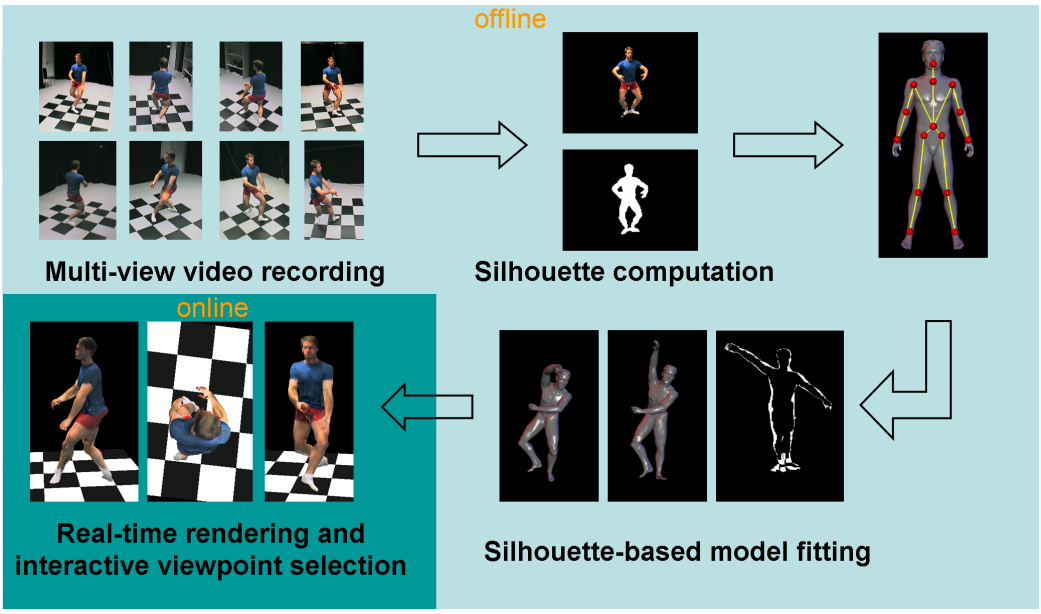


Figure 8.1: Illustration of the algorithmic workflow of our free-viewpoint video recording and rendering System.

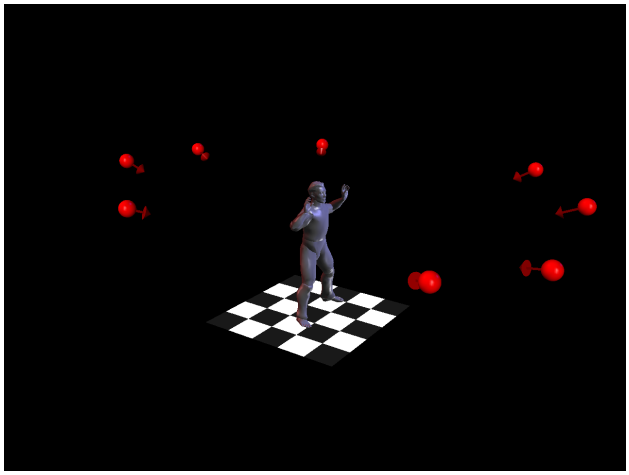


Figure 8.2: Illustration of one semi-circular camera arrangement that we used for our experiments. Red spheres denote camera positions, the arrows indicate their viewing directions.

plied camera setup - evolution I. For recording we placed our eight video cameras in a semi-circular arrangement around the center of the scene (Fig. 8.2). Hardware limitations allow us to to record frame-synchronized multi-view video footage at a frame rate of 15 fps and a resolution of 320x240 pixels only. At the higher frame rate of 640x480 pixels the I/O overhead limits the frame rate to 10 fps (see Chap. 4). All cameras are calibrated into a common coordinate system and the effects of first order lens distortion are eliminated. To establish multi-view color consistency, all cameras are relatively color-calibrated.

The inputs to both motion parameter estimation and model initialization are silhouette images of the person in the foreground. We calculate these silhouette images by applying our color-based background subtraction scheme. Isolated holes in the silhouettes can be filled via morphological dilate and erode operations [Jain95].

8.3 The Adaptable Human Body Model

While 3D object geometry can be represented in different ways, here, a triangle mesh representation is used because it offers a closed and detailed surface description, and, even more importantly, it can be rendered very fast on graphics hardware. Since the model must be able to perform the same complex motion as its real-world counterpart, it is composed of multiple rigid-body parts that are linked by a hierarchical kinematic chain (c.f. Chap. 2). The joints between segments are suitably parameterized to reflect the object's kinematic degrees of free-

dom. Besides object pose, also the dimensions of the separate body parts must be kept adaptable (within reasonable bounds) as to be able to match the model to the object's individual stature.

As geometry model, a publicly available VRML geometry model of a human body is used, Fig. 8.3a. The model consists of 16 rigid body segments, one for the upper and lower torso, neck, and head, and pairs for the upper arms, lower arms, hands, upper legs, lower legs and feet. In total, more than 21000 triangles make up the human body model. A hierarchical kinematic chain connects all body segments, resembling the anatomy of the human skeleton. 17 joints with a total of 35 joint parameters define the pose of the virtual character. Different joints in the body model provide different numbers of rotational degrees of freedom the same way as the corresponding joints in an anatomical skeleton do. Furthermore, we have employed a special parameterization of each limb that is particularly suited for application in our pose estimation process. For global positioning, the model provides three translational degrees of freedom which influence the position of the skeleton root. The root of the model is located at the pelvis. The kinematic chain is functionally separated in an upper body half and a lower body half. The initial joints of both kinematic sub-chains spatially coincide with the model root.

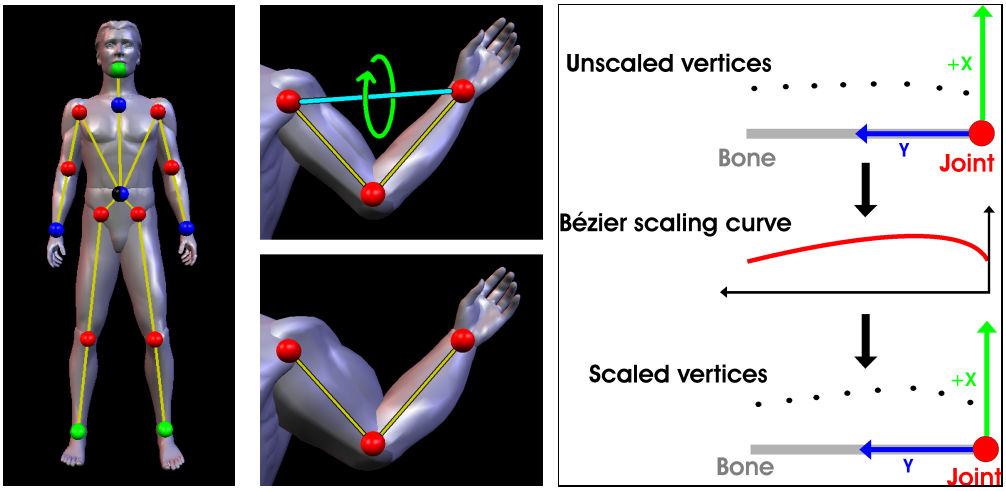
In Fig. 8.3a individual joints in the body model's kinematic chain are drawn and the respective joint color indicates if it is a 1-DOF hinge joint, a 3-DOF ball joint, or a joint being part of our custom limb parameterization. Each limb, i.e. complete arm or leg, is parameterized via four degrees of freedom. These are the position of the tip, i.e wrist or ankle, in local coordinates, and the rotation around an axis connecting root and tip (Fig. 8.3b). This limb parameterization was chosen because it is particularly well-suited for an efficient grid search of its parameter space which we describe in Sect. 8.6. The head and neck articulation is specified via a combination of a 3-DOF ball joint and a 1-DOF hinge joint. The wrist offers three degrees of freedom and the foot motion is limited to a 1-DOF hinge rotation around the ankle. In total, 35 pose parameters fully specify a body pose.

Starting with a generic body model, the initial geometry does not have the same proportions as the human which it currently is meant to represent. Thus, in addition to the pose parameters, the model provides anthropomorphic shape parameters that control the bone lengths as well as the structure of the triangle meshes defining the body surface.

Each of the 16 body segments features a scaling parameter that scales the bone as well as the surface mesh uniformly in all three coordinate directions (in the local coordinate frame of the segment). This parameter provides control over the bone length but does not give sufficient control of the mesh geometry.

In order to match the geometry more closely to the shape of the real human each segment features four one-dimensional Bézier curves $B_{+x}(u), B_{-x}(u), B_{+z}(u), B_{-z}(u)$, which are used to scale individual coordinates of

each vertex in the local triangle mesh. In the local coordinate system a bone is aligned with the y-axis. For each direction orthogonal to the bone direction, i.e. +x,-x and +z,-z one scaling curve is applied. A novel vertex coordinate in one of the orthogonal directions is obtained by multiplying its default position with the value of the appropriate scaling curve. The correct curve parameter u is obtained by transforming the local y-coordinate of the vertex to the range $[0, 1]$ with respect to the bone length (Fig. 8.3c). Each Bézier curve is defined via four control values, making it an additional 16 shape parameters for each body segment. This surface deformation scheme gives a good control over the segment geometry while only requiring comparably few parameters.



(a) (b) (c)

Figure 8.3: (a) Surface model and the underlying skeletal structure. Spheres indicate joints and the different parameterizations used; blue sphere - 3 DOF ball joint, green sphere - 1 DOF hinge joint, red spheres (two per limb) - 4 DOF. The black/blue sphere indicates the location of three joints, the root of the model and joints for the upper and lower half of the body. (b) The upper figure shows the parameterization of a limb, consisting of 3 DOF for the wrist position in local shoulder coordinates (shown in blue) and 1 DOF for rotation around the blue axis. The lower right figure demonstrates an exaggerated deformation of the arm that is achieved by appropriately tweaking the Bézier parameters. (c) Schematic illustration of local vertex coordinate scaling by means of a Bézier scaling curve for the local +x direction.

8.4 Silhouette Matching

The challenge in applying model-based analysis for free-viewpoint video reconstruction is to find a way how to automatically and robustly adapt the geometry model to the subject's appearance as it was recorded by the video cameras. Since the geometry model is suitably parameterized to alter, within anatomically plausible limits, its shape and pose, the problem consists of determining the parameter values that achieve the best match between the model and the video images. This task is regarded as an optimization problem.

Etienne de Silhouette, Louis XV.'s financial minister, realized that the outline of a man's head, while inexpensive to acquire, comprises enough characteristic information about the depicted subject to enable recognizing the person. To save money, he ordered silhouette drawings to be made of all civil servants, instead of oil paintings, as was customary before his time.

250 years later, silhouette renderings are cheap to display on modern graphics hardware. The subject's silhouettes, as seen from the different camera viewpoints, are used to match the model to the video images (an idea used in similar form in[Lensch01]): The model is rendered from all camera viewpoints, and the rendered images are thresholded to yield binary masks of the model's silhouettes. The rendered model silhouettes are then compared to the corresponding image silhouettes. As comparison measure, the number of silhouette pixels is determined that do not overlap. Conveniently, the exclusive-or (XOR) operation between the rendered model silhouette and the segmented video-image silhouette yields those pixels that are not overlapping. The sum of remaining pixels in all images is the mismatch score, with zero denoting an exact match.

This matching function can be evaluated very efficiently on graphics hardware (Fig. 8.4): Each input image silhouette is packed into one bitplane of a byte sized buffer. That buffer is transferred into the OpenGL stencil buffer, and successive drawings of the model compute the XOR in each bit plane. The stencil buffer stores 8-bit values which can be modified through a number of simple operations on a per-fragment basis. To compute an XOR in the stencil buffer, the model is rendered from each camera perspective into the bitplane that corresponds to this camera. The stencil buffer settings are chosen such that each fragment which passes the depth test is told to invert the bit at its corresponding pixel position. In order to prevent multiple fragments from inverting a single pixel more than once, the camera's projection matrix is modified so that all vertices project into z=0 plane. With the depth test properly set to reject all fragments with a z-value larger or equal to the value in the depth buffer, at most one inversion occurs per pixel. Once the XOR has been computed for all 8 cameras, the stencil buffer that contains in each of its bit-planes one binary XOR image (Fig. 8.5) is transferred back to the CPU. In software the total number of set bits is counted which is at the same

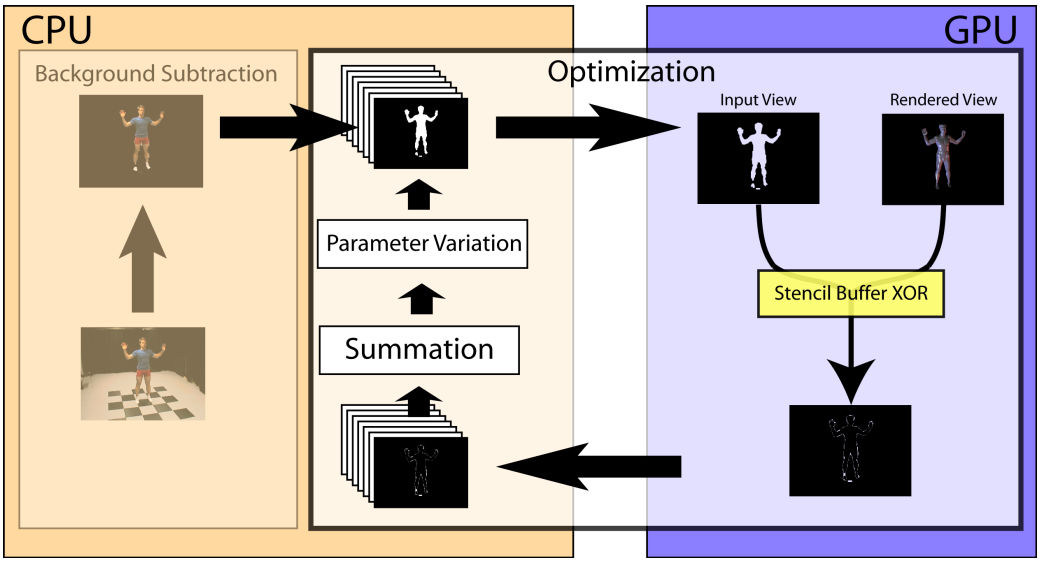
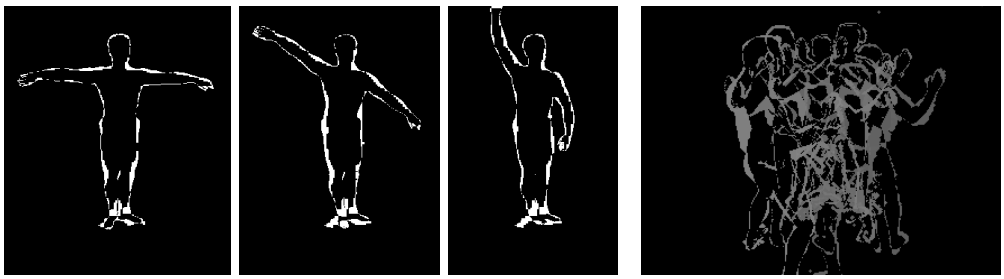


Figure 8.4: Hardware-based analysis-by-synthesis for free-viewpoint video: To match the geometry model to the multi-video recordings of the actor, the image foreground is segmented and binarized. The model is rendered from all camera viewpoints. The boolean XOR operation is executed between the foreground images and the corresponding model renderings, and the number of remaining pixels in all camera views serves as matching criterion. Model parameter values are varied via numerical optimization until the XOR result is minimal. The numerical minimization algorithm runs on the CPU while the energy function evaluation is implemented on the GPU.

time the numerical value of our energy function. An Nvidia GeForce3^{TM} graphics card performs more than 100 of such matching function evaluations per second. Currently, the main limiting factor is the overhead generated by the read-back from the graphics board. To adapt model parameter values such that the mismatch score becomes minimal, a standard numerical optimization algorithm, such as Powell's method [Press02], runs on the CPU. For each new set of model parameter values, the optimization routine invokes the matching function evaluation routine on the graphics card.

One valuable benefit of model-based analysis is the low-dimensional parameter space when compared to general reconstruction methods: The parameterized model provides only a few dozen degrees of freedom that need to be determined, which greatly reduces the number of potential local minima. Furthermore, many high-level constraints are implicitly incorporated, and additional constraints can be easily enforced by making sure that all parameter values stay within their anatomically plausible range during optimization. Finally, temporal coherence is

straight-forwardly maintained by allowing only some maximal rate of change in parameter value from one time step to the next.



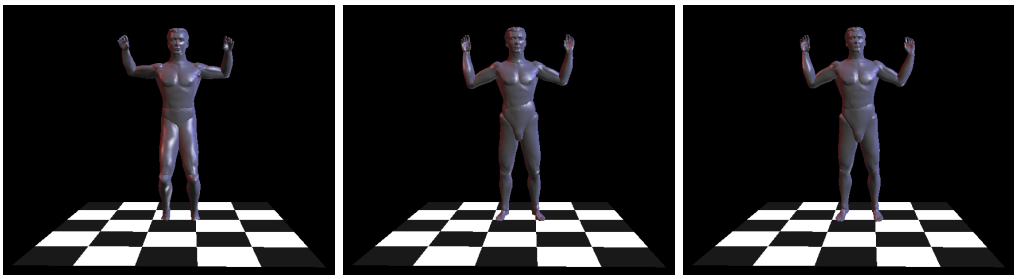
(a) (b)

Figure 8.5: (a) Binary XOR images for a single camera view at different times steps. In a single time step, the energy function sums up the numbers of set pixels in all input views. (b) The content of the stencil buffer after one evaluation of the silhouette-XOR energy function. Each bitplane contains one XOR image for one of the input cameras. Gray values have been enhanced in order to make the content better visible.

8.5 Model Initialization

To apply the silhouette-based model pose estimation algorithm to real-world multi-video footage, the generic geometry model must first be initialized, i.e. its proportions must be adapted to the subject in front of the cameras. To achieve this we run a numerical minimization in the scaling parameter space of the model using the silhouette XOR energy function. The model provides one scaling and 16 Bézier deformation parameters per body segment that control the shape and proportions of the model. This way, all segment surfaces can be deformed until they closely match the actor's stature.

During model initialization, the actor stands still for a brief moment in a predefined pose to have his silhouettes recorded from all cameras. The generic model is rendered for this known initialization pose, and without user intervention, the model proportions are automatically adapted to the individual's silhouettes. First, only the torso is considered. Its position and orientation is determined approximately by maximizing the overlap of the rendered model images with the segmented image silhouettes. Then the pose of arms, legs and head are recovered by rendering each limb in a number of orientations close to the initialization pose



(a) (b) (c)

Figure 8.6: (a) template model geometry; (b) model after 5 iterations of pose and scale refinements; (c) model after adapting the Bézier scaling parameters.

and selecting the best match as starting point for refined optimization. This step is identical to the optimization which we perform for pose determination (see Sect. 8.6). Following the model hierarchy, the optimization itself is split into several sub-optimizations in lower-dimensional parameters spaces. After the model has been coarsely adapted in this way, the uniform scaling parameters of all body segments are adjusted. For selected body segments (e.g. arm and leg segments) we found it advantageous to scale their dimension only in the bone direction, and to leave the control of the triangle mesh shape orthogonal to this direction to the Bézier parameters. The algorithm then alternates between optimizing joint parameters and segment scaling parameters until it has converged to the error function's minimum. Through experiments we have found that five to ten iterations are typically sufficient. An adaptive approach that terminates when the changes in error function become minor is also feasible. Now that body pose and proportions have been established, the Bézier control parameters of all body segments are optimized in order to fine-tune each segment's outline such that it complies with the recorded silhouettes. For the hands and the feet we do not optimize the Bézier parameters. To speed up the initialization process, deformed model is rendered by means of a vertex program on the GPU. This way, deformation parameters can be efficiently altered on-the-fly. In Fig. 8.6 the initial model shape, its shape after five iterations of pose and scene optimization, and its shape after Bézier scaling are shown. For numerical minimization we employ the direction set downhill optimization method by Powell (see also Sect. 8.6).

Obviously, an exact match between model outline and image silhouettes is not attainable since the parameterized model has far too few degrees of freedom. Thanks to advanced rendering techniques (Sect. 8.8) an exact match is neither needed, nor is it actually desired: Because the recorded person may wear rela-

tively loose, flexible clothes, the silhouette outlines can be expected to be inaccurate, anyway. By not being dependent on exact image silhouette information, model-based motion analysis is capable of robustly handling also non-rigid object surfaces.

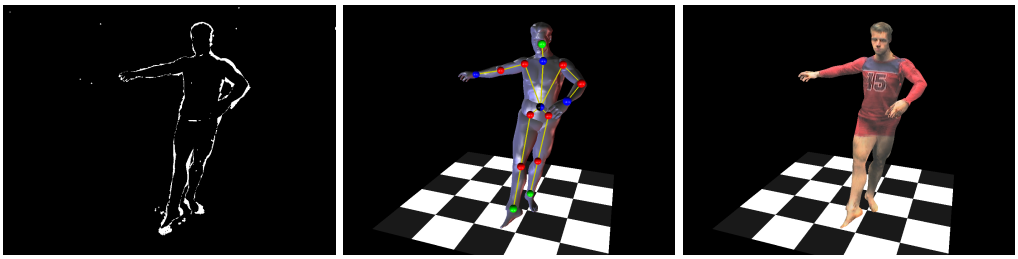
The initialization procedure takes only a few seconds, after which the segments' scaling parameter values and Bézier surface deformation values are known. These are kept fixed from now on. During motion capture, only the 35 joint parameters are optimized to follow the motion of the dancer.

8.6 Motion Parameter Estimation

The human body is capable of a very large range of complex motions. Given the relatively small set of 35 joint parameters, only a subset of all possible body poses can be reproduced. Fortunately, modeling the 17 most important joints of human anatomy suffices to capture gross motor skills faithfully and to realistically reproduce even such expressive movements as ballet dance.

The individualized geometry model automatically tracks the motion of the human dancer by optimizing the 35 joint parameters for each time step. The appropriate pose parameters are found via a silhouette-based marker-free motion

Figure 8.7: From eight image silhouettes per time step, model-based analysis automatically captures the complex motion of a ballet dance performance.



(a) (b) (c)

Figure 8.8: (a) Silhouette XOR; (b) body model; (c) textured body model from same camera view.

capture approach that does not expect the person to wear any specialized apparel. This is a necessary pre-condition for free-viewpoint video reconstruction, since only if motion is captured completely passively can the video imagery be used for texturing. The model silhouettes are matched to the segmented image silhouettes of the actor so that the model performs the same movements as the human in front of the cameras, Fig. 8.7 and Fig. 8.8.

At each time step an optimal stance of the model is found by performing a numerical minimization of the silhouette XOR energy functional in the space of pose parameters.

The numerical optimization of the multi-dimensional, non-convex matching functional can potentially result in sub-optimal fitting results. A straightforward approach would be to apply any standard numerical minimization method to optimize all pose-related degrees of freedom in the model simultaneously. This simple strategy, however, exhibits some of the fundamental pitfalls that make global optimization infeasible. In the global case, the energy function reveals many erroneous local minima. Fast movements between consecutive time frames are almost impossible to track since it may happen that no overlap between the model and the image silhouette occurs that guides the optimizer towards the correct solution. A different problem arises if one limb moves very close to the torso. In this case, it is quite common for global minimization method to find a local minimum in which the limb penetrates the torso.

One of the major issues in marker-free pose estimation is therefore the question of how to constrain the search space of possible pose parameters. Many different ways to formulate these constraints have been presented in the literature (see Chap. 3). Here, we present a method that enables us to use a standard direction set minimization scheme to robustly estimate pose parameters. We effectively constrain the search space by exploiting structural knowledge about the

human body, knowledge about feasible body poses, temporal coherence in motion data and a grid sampling preprocessing step.

To efficiently avoid local minima, the model parameters are not all optimized simultaneously. Instead, the model's hierarchical structure is exploited. Model parameter estimation is performed in descending order with respect to the individual segments' impact on silhouette appearance and their position along the model's kinematic chain, Fig 8.9. First, position and orientation of the torso is varied to find its 3D location. Next, arms, legs and head are considered. Finally, hands and feet are regarded.

Temporal coherence is exploited by initializing the optimization for one body part with the pose parameters found in the previous time step. Optionally, a simple linear prediction based on the two preceding parameter sets is feasible.

Due to the limb parameterization described in Sect. 8.3, fitting an arm or leg is a four-dimensional optimization problem. In order to cope with fast body motion that can easily mislead the optimization search, we precede the numerical minimization step with a regular grid search. The grid search samples the four-dimensional parameter space at regularly-spaced values and checks each corresponding limb pose for being a *valid pose*. Using the arm as an example, a valid pose is defined by two criteria. Firstly, the wrist and the elbow must project into the image silhouettes in every camera view. Secondly, the elbow and the wrist must lie outside a bounding box defined around the torso segment of the model. For all valid poses found, the error function is evaluated, and the pose that exhibits the minimal error is used as starting point for a direction set downhill minimization. The result of this numerical minimization specifies the final limb configuration. The parameter range from which the grid search draws sample values is adaptively changed based on the difference in pose parameters of the two preceding time steps. The grid sampling step can be computed at virtually no cost and

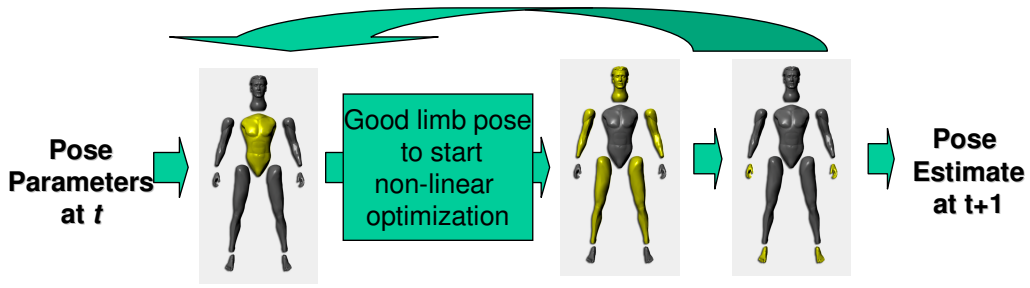


Figure 8.9: For motion capture, the body parts are matched to the images in hierarchical order: the torso first, then arms, legs and head, finally hands and feet. Local minima are avoided by a limited regular grid search for some parameters prior to optimization initialization.

significantly increases the convergence speed of the numerical minimizer.

Optionally, the whole pose estimation pipeline can be iterated on for single time step. The numerical minimizer we employ throughout our free-viewpoint video system is Powell's downhill minimization procedure [Press02]. It is both used for motion capture and model initialization. As a direction set method it always pertains a number of candidate descend directions in parameter space. The optimal descend in one direction is computed using Brent's line search method. Our algorithm shows that with the appropriate combination of constraints even such a standard minimizer can perform as well in human body tracking as a more complicated statistical optimization schemes, such as condensation [Deutscher00]. Our method does not require the estimation of statistical process models, and is general enough to be used with a large range of differently structured body models.

8.7 Accelerating Motion Capture

We have demonstrated in the preceding section that by following the hierarchical structure of the human skeleton, the pose determination problem for the whole body can be decomposed into multiple smaller problems on kinematic sub-chains. It is important to note that several of these sub-problems can be solved independently from each other. We can exploit this compartmentalized nature of the problem to speed up two time-critical algorithmic components of the motion capture system, the evaluation of the energy function and the optimization search for the pose parameters [Theobalt03b]. Firstly, the energy function evaluation speed can be significantly increased by only considering sub-windows in the image plane. Secondly, the rendering overhead for energy function evaluation can be reduced by selectively rendering only those body parts that are currently optimized. Finally, the motion capture algorithm lends itself to a parallel implementation that jointly uses several CPUs and GPUs.

8.7.1 Accelerated Silhouette Matching

Variable Window Size

The partitioning of the motion parameter estimation process into a number of sub-optimizations of specific body parts implies that only the motion of one body part is considered at a time. The energy function evaluation can thus be restricted to that region of the image plane in which the body part projects. This body part may take up only a small portion of the overall silhouette. An arm or leg is much smaller than a person's torso, and a hand or foot is much smaller than an

Figure 8.10: Global energy function (center) and smaller sub-windows (128x128 pixels) used to optimize the arm positions (shown for one camera view only).

arm or leg. Therefore, there is no need to transfer the entire silhouette image to the GPU. For each body part in the hierarchy and each camera view, a window with a fixed size is selected. When that body part's pose is being estimated, only these sub-windows of the input silhouettes are transferred to and from the GPU. In each camera view, the window is centered at the projected position of the body part's center of mass from the preceding time step (Fig. 8.10). Consequently, the window locations, and thus the silhouette data sent to the GPU, do not change for a given time instant.

A reduced window size allows for rapid evaluation of the energy function for small body parts. As such, the choice of window sizes is critical to both performance and accuracy. The projected size of a body part can vary greatly depending on position. Obviously, in some extreme cases (for example the person holds his face directly in front of a certain camera), a body part may exceed the window size in a specific camera. However, it is worth noting that with our camera setup and reasonable window sizes, even if the projection of a body part exceeds the window of a certain camera, it will certainly fall entirely within the window of several other cameras, and thus its pose still be reliably estimated. Regardless of this fact, we choose our window size very conservatively (128x128 pixels for arms and legs, 64x64 pixels for head, hands and feet).

While optimal pose parameters for the torso are searched the pose of the whole body model is influenced. The model silhouette varies over a large region in image space making it impossible to identify a sub-window of significantly reduced dimensions. However, we have found that the torso and its linked body parts are large enough to be reliably tracked with silhouette images at half the input resolution, i.e. 160x120 pixels.

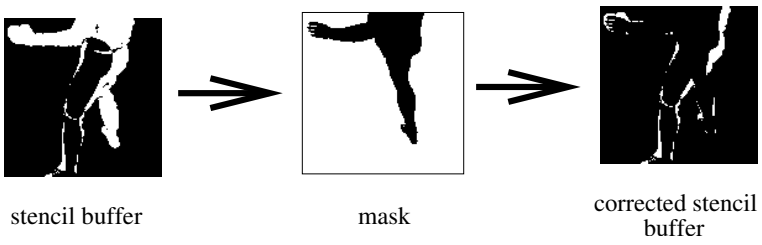
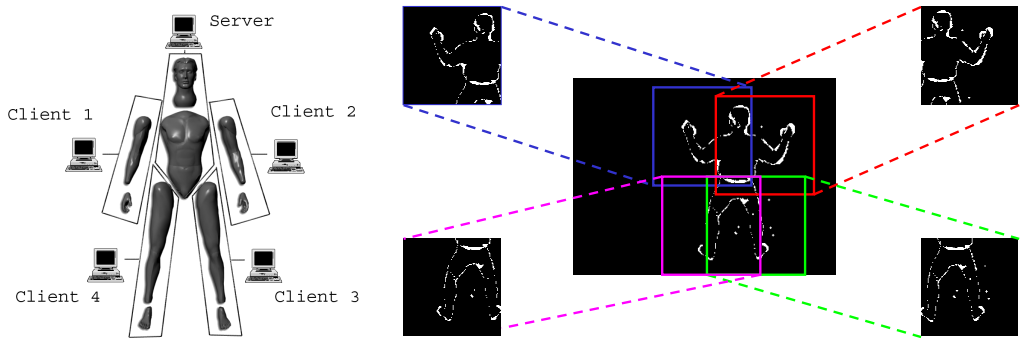


Figure 8.11: Body part pre-rendering: During the motion parameter estimation of a kinematic sub-chain, only the geometry of this sub-chain is rendered. To correct for errors in the XOR energy function, a mask is pre-computed before the optimization starts. A bitwise AND between the mask and the stencil buffer is computed to get the final value of the energy function. The same process applies to all bit-planes of the stencil buffer, i.e. all camera views.

Body Part Pre-rendering

The energy function evaluation rate can be further sped up by reducing the number of geometric primitives that need to be rendered in one iteration. During optimization of a limb, for example, the pose parameters of all the other body parts are not modified, hence their projection into all the camera views does not change. The energy function evaluation speed can therefore be greatly improved by only rendering the geometry of those body parts that are currently optimized. The problem with this approach is that it adds a wrong contribution to the XOR energy function. During computation of the XOR in the stencil buffer the bits are set in those regions where there are pixels from the image silhouettes, but where no body part projects to, since it was excluded from rendering. To eliminate this erroneous contribution to the energy function, an additional pre-rendering step needs to be performed which creates (for each camera view) a mask that corrects the error function on the CPU. This mask contains a 0-bit for each pixel to which a body part projects that does not change during optimization, and a 1-bit for all other pixels. The masks are generated by setting the stencil buffer configuration appropriately and rendering the model without the body parts that are currently optimized from each camera view. The energy function error is corrected on the CPU by computing a pixel-wise AND between the stencil buffer bit-planes and all camera masks before counting the set bits. Fig. 8.11 illustrates the modified error function evaluation for one of the camera views contributing to its final value.



(a) (b)

Figure 8.12: (a) Illustration of which components of the distributed model-based motion analysis system are responsible for the pose of which body part. (b) After torso position and orientation have been determined, the pose of each extremity is estimated on a separate PC.

8.7.2 Parallel Pose Estimation

The compartmentalized nature of the problem suggests that a distributed computation approach is feasible. The optimization of a specific body part is primarily dependent on the optimization of body parts which are higher in the hierarchy and relatively independent of any other body parts. For example, the correct position for the left arm is unaffected by the position of the right arm and legs.

While the enhanced version of the energy function minimizes the amount of information travelling across the GPU bus, running a distributed computation model effectively increases the bandwidth of a single "virtual"bus. In our distributed implementation, motion parameter estimation is concurrently executed on five computers (Fig. 8.12). A single computer, designated as the server, is responsible for estimating the position of the torso and head, while each of the four clients' task is to estimate the position of one limb and attached hand or foot. The computers are connected over standard 100 MBit/s network connections and communicate with a very basic protocol over TCP/IP.

Motion parameter estimation at each time step begins at the server which packs the input silhouettes into a single buffer and then transfers this information to the clients. The server optimizes the position and rotation of the torso and then sends the resulting model pose to the clients. At this point, each client begins estimating the motion parameters for its respective limb and extremity (Fig. 8.12), while the server estimates the pose parameters of the head. In this way 29 out of the 35 parameters are estimated concurrently over 5 GPUs. Once each client completed

its pose estimation, its results are transferred back to the server. The server, after receiving all results, initiates a further iteration of the complete parallel motion capture pipeline in order to refine the estimate, or it proceeds to the next time step.

Certainly, other models for distributed computation exist. It would also be feasible to employ several GPUs even during pose estimation of a single body part. However, we chose our way of implementation because of its proper balance of speed, simplicity, and hardware requirements. Introducing the additional complexity of several computers per body part would provide relatively minor speed improvements in comparison to the speed improvements of using a single computer versus five.

The motion capture results that we obtain with the distributed system are as accurate as the ones obtained with the single PC implementation. With some rare exceptions, the estimates obtained for each limb or extremity are completely independent of that of other limbs. This can be explained by the fact that, for a large majority of poses, the limbs are distinct from each other in at least one camera view. The situation one would expect to be problematic for distributed motion parameter estimation, namely where there is no distinction between two limbs in any camera view, is a fundamental problem for any silhouette based method. Fortunately, such poses (for example a person in the fetal position) are quite uncommon.

8.8 Rendering

A high-quality 3D geometry model is now available that closely matches the dynamic object in the scene over the entire length of the sequence. To display the object photo-realistically, the recorded video images are used for texturing the model surface.

Since time-varying video data is available, model texture doesn't have to be static. Time-varying cloth folds and creases, shadows and facial expressions are faithfully reproduced, lending a very natural, dynamic appearance to the rendered object. At 264 MBytes/sec AGP standard transfer bandwidth, the available eight images of 225 KBytes each are uploaded to the graphics card at rates faster than the available 15 frames per second in the input video data.

Modern graphics hardware supports projective texturing to apply the images as texture to the triangle-mesh model. To attain optimal rendering quality, however, the video textures need to be processed offline prior to real-time rendering: Since the available texture consists of multiple images taken from different viewpoints, the images need to be appropriately blended in order to appear as one consistent object surface texture.

(a) $\alpha = 0$ (b) $\alpha = 3$ (c) $\alpha = 15$

Figure 8.13: Texturing results for different values of the control factor α.

Also, local visibility must be taken into account, and any adverse effects due to the inevitable small differences between model geometry and the true 3D object surface must be countered effectively.

8.8.1 Blending

Most surface areas of the model are seen from more than one camera. If the model geometry corresponded exactly to that of the recorded object, all camera views could be weighted according to their proximity to the desired viewing direction and blended without loss of detail. However, the generic geometry model has been adapted to the recorded actor using only a comparatively small number of available parameters. Furthermore, the subject is being modeled as consisting of rigid body elements, which may be only a coarse approximation, e.g. if the person wears loose clothes. As a result, the model surface locally deviates from the true geometry. Blending multiple image projections from different angles thus causes blurred texture, and texture interpolation introduces inconsistencies when the viewpoint is moving.

If surface reflectance can be assumed to be approximately Lambertian, view-dependent reflection effects play no significant role, and high-quality, detailed model texture can still be obtained by blending the video images cleverly. Let θ_i denote the angle between a vertex normal and the optical axis of camera i. By emphasizing for each vertex individually the camera view with the smallest angle θ_i, i.e. the camera that views the vertex most head-on, a consistent, detail-preserving texture is obtained. A visually convincing weight assignment has been found to be

$$\omega_i = \frac{1}{(1 + \max_j(1/\theta_j) - 1/\theta_i)^\alpha} \tag{8.1}$$

where the weights ω_i are additionally normalized to sum up to unity. The parameter α determines the influence of vertex orientation with respect to camera viewing direction and the impact of the most head-on camera view per vertex (Fig 8.13). Singularities are avoided by clamping the value of $1/\theta_i$ to a maximal value.

Although it is valid to assume that most types of apparel have purely Lambertian reflectance, in some cases the reproduction of view-dependent appearance effects may be wanted. To serves this purpose, our method provides the possibility to compute view-dependent rescaling factors, ρ_i, for each vertex on-the-fly while the scene is rendered:

$$\rho_i = \frac{1}{\phi_i} \tag{8.2}$$

where ϕ_i is the angle between the direction to the outgoing camera and the direction to input camera i.

8.8.2 Visibility

Projective texturing on graphics hardware has the disadvantage that occlusion is not taken into account, so hidden surfaces get also textured. The z-buffer test, however, allows determining for every time step which object regions are visible from each camera.

Due to the use of a parameterized geometry model, the silhouette outlines in the images do not correspond exactly to the outline of the model. When projecting video images onto the model, a texture seam belonging to some frontal body segment may fall onto another body segment farther back, Fig. 8.14a. To avoid such artifacts, *extended soft shadowing* is applied: For each camera, all object regions of zero visibility are determined not only from the camera's actual position, but also from several slightly displaced virtual camera positions (Fig. 8.14b). Each vertex is tested whether it is visible from all camera positions, actual as well as virtual. A triangle is textured by a camera image only if all of its three vertices are completely visible from that camera.

While too generously segmented silhouettes do not affect rendering quality, too small outlines can cause annoying untextured regions. To counter such rendering artifacts, all image silhouettes are expanded by a couple of pixels prior to rendering. Using a morphological filter operation, the object outlines of all video images are dilated to copy the silhouette boundary pixel color values to adjacent background pixel positions (Fig. 8.15).

(a) (b)

Figure 8.14: (a) Small differences between object silhouette and model outline cause erroneous texture projections. (b) By projecting each video camera's image onto the model also from slightly displaced camera positions, regions of dubious visibility are determined (purple) which are then not textured by the camera.

8.8.3 Real-time Free-Viewpoint Rendering

With the processed video textures, the dynamic model is rendered interactively from any arbitrary viewpoint. Prior to display, the geometry model as well as the video cameras' calibration data is transferred to the graphics card. During rendering, the user's viewpoint information, the model's updated pose parameter values, the current video images, as well as the visibility and blending coefficients v_i, ω_i for all vertices and cameras i are continuously transferred to the graphics card.

The color of each rendered pixel $c(j)$ is determined by blending all l video

Figure 8.15: Morphologically dilated segmented input video frames that are used for projective texturing.

Figure 8.16: Capturing a smile: Texture detail is preserved. Block artifacts are due to the limited camera resolution.

images I_i according to

$$c(j) \quad = \quad \sum_{i=1}^{l} v_i(j) * \rho_i(j) * \omega_i(j) * I_i(j) \tag{8.3}$$

where $\omega_i(j)$ denotes the blending weight of camera i, $\rho_i(j)$ is the optional view-dependent rescaling factor, and $v_i(j) = \{0,1\}$ is the local visibility. During texture pre-processing, the weight products $v_i(j)\rho_i(j)\omega_i(j)$ have been normalized to ensure energy conservation. Expression (8.3) is evaluated for each fragment by a fragment program on the graphics board. The rasterization engine interpolates the blending values from the triangle vertices.

8.9 Results

Our free-viewpoint video reconstruction and rendering approach has been tested on a variety of test scenes, ranging from simple walking motion over karate performances to complex and expressive ballet dance. The sequences are between 100 and 400 frames long and were recorded from eight camera perspectives.

Ballet dance performances are ideal test cases as they exhibit rapid, complex motion. The motion capture subsystem demonstrates that it is capable of robustly following human motion involving fast arm motion, complex twisted poses of the extremities, and full body turns (Fig. 8.21). Certainly, there are extreme body poses such as the fetal position that cannot be reliably tracked due to insufficient visibility. To our knowledge, no non-intrusive system has demonstrated that it is able to track such extreme positions. In combination with our texture generation approach convincing novel viewpoint renditions cab be generated, as it is also shown in Figs. 8.18 and 8.19. Subtle surface details, such as wrinkles in clothing, are nicely reproduced in the renderings. In Fig. 8.20 snapshots from a freeze-and-rotate sequence, in which the body motion is stopped and the camera flies around the scene, are depicted. In Fig. 8.17, the original input images are compared to our rendering results from the same camera perspective. This comparison shows

Figure 8.17: For comparison, segmented input images (small) and the resulting rendered views corresponding to the same perspective (large) are depicted.

that the original appearance of the dancer is nicely reproduced even though the body geometry does not fully match. Different facial expressions of the actor are also faithfully reproduced (Fig. 8.16).

The free-viewpoint renderer can easily replay dynamic 3D scenes at the original captured frame rate of 15 fps. The maximal possible frame rate is significantly higher. Standard TV frame rate of 30 fps can easily be attained even with a GeForce^{TM} 3 GPU.

For measuring the execution speeds of individual algorithmic components of our reconstruction method we have 5 PCs at our disposition, each of which features an Intel XEON 1.8 GHz CPU, 512 MB RAM, and a graphics board with an Nvidia GeForce^{TM} 3 GPU.

First, we have a look at the execution speed of the motion capture subsystem. In our current implementation, model fitting time is dependent on the speed of the actor's motions. For slow motions, the limb parameter grid search can be confined to a narrow search space, and during subsequent downhill optimization a minimum is found very quickly.

We have measured the time needed to estimate one body pose with the single PC implementation (no parallelism) on two different input sequences. In sequence A (dancer wears blue shirt, Fig. 8.18) the motion is a lot slower than in sequence B (dancer wears red "15" shirt, Fig. 8.17) where the dancer shows very expressive twists and turns. For set A the minimum fitting time is 1.46s with an average fitting time of 6.81s per time step. For set B the lowest fitting time is 3.46s with an average of 11.73s. Due to the efficient implementation of the energy function in graphics hardware, approximately 100 energy function evaluations can be computed per second.

The exploitation of the compartmentalized nature of the pose estimation prob-

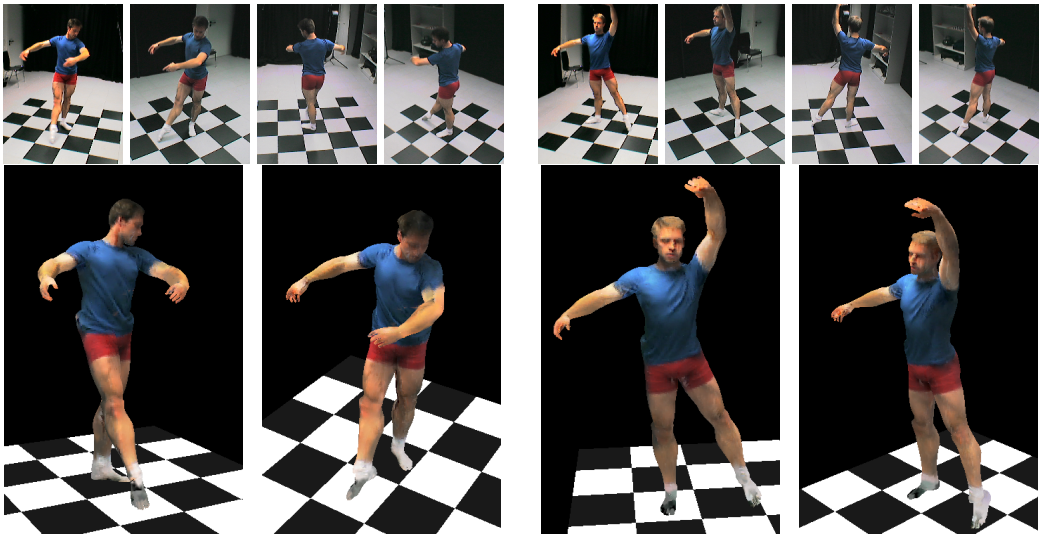


Figure 8.18: Novel viewpoints are realistically synthesized. Two distinct time instants are shown on the left and right with input images above and novel views below.

lem during energy function evaluation and optimization search (Sect. 8.7) leads to dramatic speed-ups. To illustrate the speed gains we have run tests with one very simple walking scene (Seq. C) and another very fast and expressive ballet dance sequence (Seq. D, Fig. 8.21).

In Tab. 8.1 the impact of a variable rendering window size on the evaluation speed of the silhouette XOR energy function is illustrated. As expected, the XOR energy function can be computed significantly faster if a smaller window size is used. It is also evident that as the window dimensions are reduced, rendering the model geometry becomes the major bottleneck. The method labeled XORPR uses the XOR energy function in combination with pre-rendering of unchanging

Window Size	XOR	XORPR
320x240 (full)	95.9	95.5
160x120 (half-res)	131.1	131.2
128x128 (arm)	133.7	433.1
64x64 (head)	144.9	855.4

XOR - original method
XORPR - XOR with pre-rendering

Table 8.1: Energy function evaluations per second for different stencil window sizes on a single computer.

	Seq. C	Seq. D
XOR	7.98	14.1
Single Client	3.30	10.1
Distributed	1.16	1.76

XOR - original method with single computer
Single Client - XORPR with single computer
Distributed - XORPR with 5 computers

Table 8.2: Average per-frame fitting times in seconds for different algorithmic alternatives.

body parts. One can observe that the XORPR method performs overproportionally better than pure XOR (without pre-rendering) especially for smaller window sizes. This is explained by the fact that during the optimization of the arms (128x128 window) or the head (64x64 window) most of the model geometry is excluded from rendering.

Table 8.2 shows the average fitting time for one time step of each sequence using the standard single PC implementation with standard full-frame XOR evaluation, a single PC implementation with enhanced energy function evaluation (XORPR), and a distributed implementation with enhanced energy function evaluation (XORPR). While the proposed methods to exploit parallelism already lead to a significant speed-up if only one computer is used for motion estimation, the parallel implementation leads to even faster fitting times. With the next generation of GPUs pose estimation at interactive frame rates will be feasible.

The method presented in this chapter is subject to a few limitations. First, the motion estimation process is done offline, making the system unsuitable for live

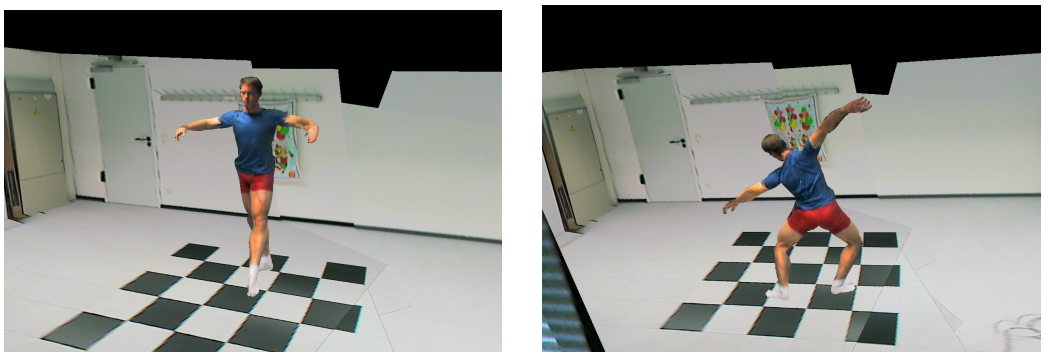


Figure 8.19: Free-viewpoint video of the dancer rendered into a virtual model of the acquisition room.

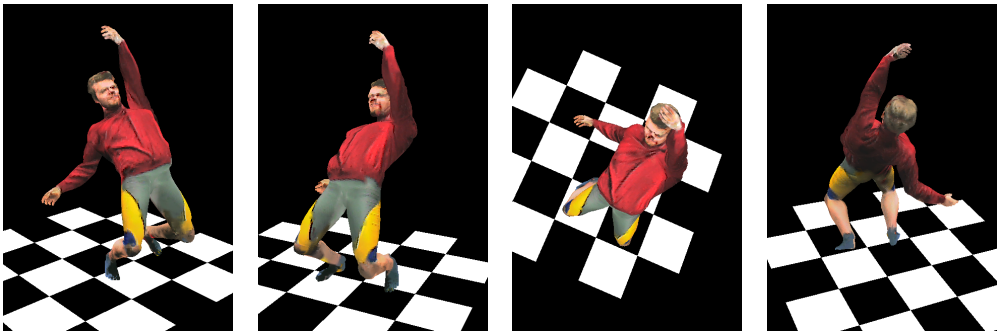


Figure 8.20: Conventional video systems cannot offer moving viewpoints of scenes frozen in time. However, with our free-viewpoint video system *freeze-and-rotate* camera shots of instable body poses are possible (bottom row).

broadcast applications. However, it is foreseeable that the ongoing performance advancement of graphics hardware will make this feasible in a few years time. The appearance of the actor cannot be faithfully represented if he wears very loose apparel. In the system presented in this chapter silhouette information is the only clue used for pose determination. However, the image data contain much more features, such as texture information which can be used to even further improve the accuracy of the free-viewpoint video reconstruction (Chap. 9)

A further limitation is that we can currently only reproduce the appearance of the actor under the illumination conditions that prevailed at the time of recording. For photo-realistic insertion into a novel virtual environment, however, the model has to be realistically rendered under new incident illumination. We address the problem of reconstructing relightable free-viewpoint videos in Chap. 10.

Even though our approach exhibits these limitations, our results show that our method enables high-quality reconstruction of free-viewpoint videos. Convincing novel viewpoint renditions of human actors can be generated in real-time on off-the-shelf graphics hardware. We achieve such a high rendering quality although we record the real-world scene only with a handful of cameras.

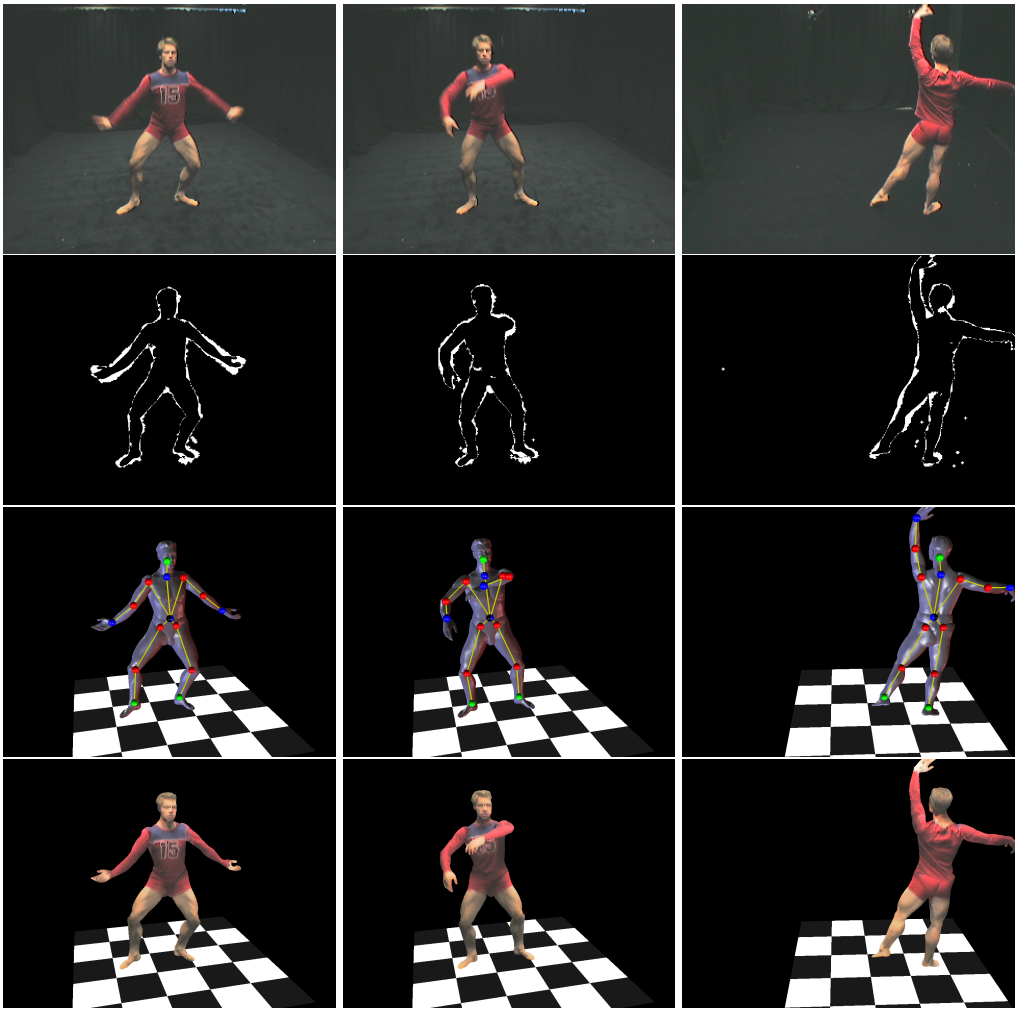


Figure 8.21: Top row: input camera views; second row: corresponding silhouette XOR; third row: fitted mode; fourth row: textured body model in the free-viewpoint video renderer.

Chapter 9

Enhanced 3D Video Reconstruction Using Texture Information

In Chap. 8 we have described the nuts and bolts of a system to reconstruct and render free-viewpoint videos of human actors. The central algorithm employed to reconstruct dynamic scene descriptions is a marker-free human motion capture method. So far, only silhouette information has been exploited to estimate pose parameters of the model from multi-view video streams. While silhouette-based analysis robustly captures large-scale movements, in some cases the body pose cannot be resolved unambiguously from object silhouettes alone. Especially small-scale motions may be unresolvable because at limited image resolution, and small protruding features, such as nose or ears, may not be discernible in the silhouettes. In a free-viewpoint video system inaccuracies in recovered body pose inevitably lead to rendering artifacts when the 3D videos are displayed. To resolve ambiguities as well as to refine silhouette-estimated model pose, we enhance our original analysis-by-synthesis approach such that the object's surface texture is considered in addition [Theobalt03a].

Since a-priori object texture may be hard to acquire, instead, the difference between predicted object appearance and the actual image recordings is used to reconstruct a correction vector field for the model. The corrective 3D motion field is reconstructed from 2D optical flows between predicted model appearance and the captured appearance of the actor in all camera views. We present a method for extracting hierarchical rigid body transformations from this motion field and show that it is used best in conjunction with, and not in place of, silhouette-based motion tracking. We demonstrate that this new hybrid method improves motion parameter estimation within our model-based free-viewpoint video approach, and

that it has a positive impact on the rendering quality of the reconstructed 3D video sequences.

The chapter continues with an overview of our method in Sect. 9.1. Thereafter, in Sect. 9.2 we describe a procedure to reconstruct 3D motion fields from 2D optical flow. In Sect. 9.3 we employ this motion field reconstruction method to develop a predictor-corrector scheme for enhanced motion capture. The chapter concludes with a presentation of results and a discussion of the method in Sect. 9.4.

9.1 Overview

In Fig. 9.1, an overview of the novel augmented algorithmic pipeline for the model-based free-viewpoint video approach is shown (see Fig. 8.1 in Chap. 8). The multi-view video acquisition procedure, the projective texture generation algorithm as well as the real-time free-viewpoint video rendering method remain unmodified. The algorithmic enhancement is the novel hybrid motion capture algorithm that we employ in our model-based analysis-by-synthesis approach.

Our novel hybrid motion estimation algorithm is a two-step predictor-corrector scheme. Considering an arbitrary time step $t+1$, the augmented motion capture algorithm works as follows: Starting with a set of 35 body pose parameters P_t that were found to be optimal for time step t, the system first computes an estimate of the pose parameters $P'_{sil,t+1}$ at time $t+1$ by employing the silhouette-based motion estimation scheme (see Chap. 8 Sect. 8.6). In a second step, estimate $P'_{sil,t+1}$ is augmented by computing a 3D corrective motion field from optical flows. The model that is standing in pose $P'_{sil,t+1}$ and that is textured with the video images from time t is rendered into all camera views. The images of the back-projected model form a prediction of the person's appearance at $t+1$. The optical flows are computed for each pair of back-projected model view and corresponding segmented video frame at time $t+1$. From camera calibration, the camera matrix of each recording imaging sensor is known. Since, in addition, the geometric structure of the body model is available, for each model vertex corrective flow vectors in 3D can be computed from the corrective 2D optical flows in all camera views. The end-point of each motion vector is the position at which the respective vertex should be in order for the whole model to be in a stance that is photo-consistent with all camera views. This information has to be translated into pose update parameters for the model's joints that bring the model into the photo-consistent configuration. We compute the differential pose update, $P_{\text{diff},t+1}$, in a least-squares sense and apply it to the model after $P'_{sil,t+1}$ in order to obtain the final pose estimate P_{t+1} for time $t+1$. The final pose parameter estimate serves as a starting point for the pose determination in the next time step.

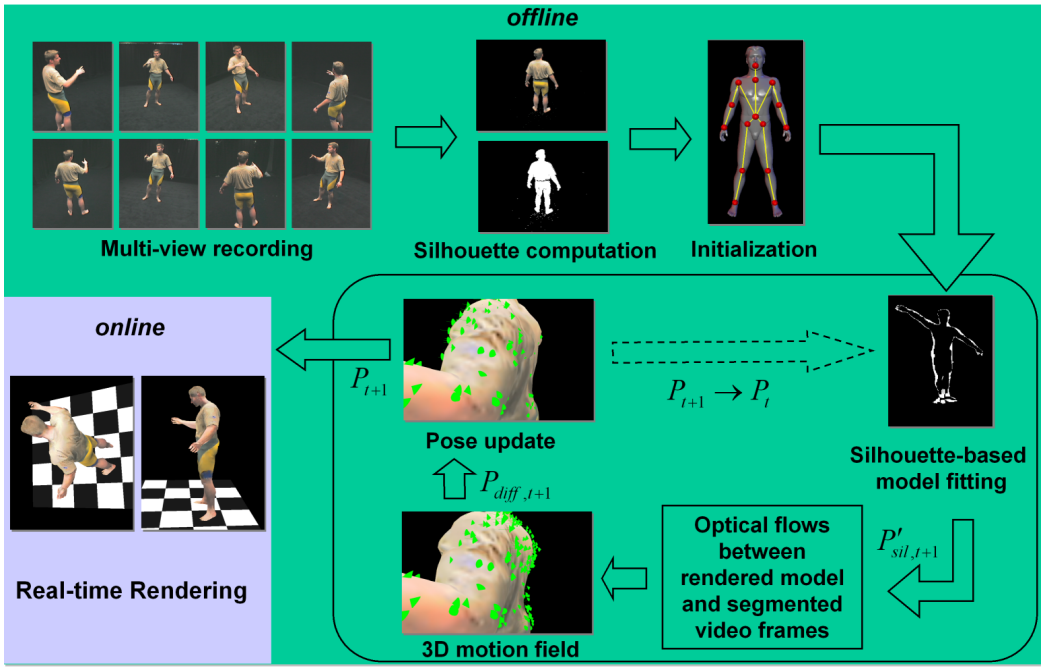


Figure 9.1: Overview of the algorithmic workflow of the free-viewpoint video system with augmented motion capture sub-system.

9.2 Reconstructing a 3D Motion Field from 2D Optical Flow

The optical flow observed by a camera (Chap. 2) is a 2D-projection of a 3D motion field in the real world. The goal of motion capture is the recovery of the parameters of three-dimensional motion. A reconstructed 3D motion field from optical flows in multiple camera views can be used to compute these parameters. The reconstruction of the 3D motion field, also know as the *scene flow*, from the 2D optical flows is possible using a technique described in [Vedula99].

If correspondences in the image plane are known, i.e. it is known to which image coordinates 3D points project in each camera view, the scene flow can be reconstructed by solving a linear system of equations. In our free-viewpoint video approach, the correspondences are known for each vertex because we have an explicit body model, and the projection matrices $\mathbf{P}_i$ for each recording camera i have been determined via calibration. The projection matrix $\mathbf{P}_i$ describes the relationship between a 3D position of a vertex and its projection into the image

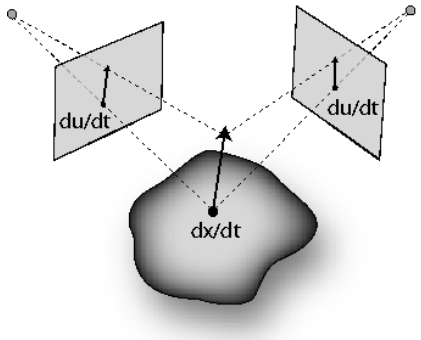


Figure 9.2: 3D motion (scene flow) of a surface point and the corresponding observed optical flows in two camera views.

plane of the camera i, $u_i = (u_i, v_i)^T$.

The differential relationship between the vertex $\mathbf{x}$ with coordinates $(x,y,z)^T$ and u_i is described by the 2×3 Jacobian matrix $J_i = \frac{\partial u_i}{\partial \mathbf{x}_i}$:

$$o_i = \frac{du_i}{dt} = J_i \frac{d\mathbf{x}}{dt} \tag{9.1}$$

In other words, the Jacobian describes the relationship between a small change in 3D position of a vertex, and the change of its projected image in camera i. The term $\frac{du_i}{dt}$ is the optical flow o_i observed in camera i, $\frac{d\mathbf{x}}{dt}$ is the corresponding scene flow of the vertex (Fig. 9.2). Having a mathematical camera model, the Jacobian can be computed analytically (see [Vedula99]).

If a vertex is visible from at least two camera views, an equation system of the form $\mathbf{B}\frac{d\mathbf{x}}{dt} = \mathbf{U}$ can be formulated, whose solution is the scene flow of the vertex. The matrix $\mathbf{B}$ and the vector $\mathbf{U}$ evaluate to:

$$\mathbf{B} = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} & \frac{\partial u_1}{\partial z} \\ \frac{\partial v_1}{\partial x} & \frac{\partial v_1}{\partial y} & \frac{\partial v_1}{\partial z} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \frac{\partial u_N}{\partial x} & \frac{\partial u_N}{\partial y} & \frac{\partial u_N}{\partial z} \\ \frac{\partial v_N}{\partial x} & \frac{\partial v_N}{\partial y} & \frac{\partial v_N}{\partial z} \end{bmatrix}, \mathbf{U} = \begin{bmatrix} \frac{\partial u_1}{\partial t} \\ \frac{\partial v_1}{\partial t} \\ \cdot \\ \cdot \\ \frac{\partial u_N}{\partial t} \\ \frac{\partial v_N}{\partial t} \end{bmatrix} \tag{9.2}$$

N is the number of camera views. A least-squares solution to this equation system can be found via singular value decomposition (SVD) [Press02].

9.3 Texture-enhanced Silhouette-based Motion Capture

For acquiring a three-dimensional representation of the human actor, we employ a model-based analysis-by-synthesis approach (see Chap. 8). The shape-adaptable generic human body model we employ consists of 16 body segments that are connected via a kinematic skeleton featuring 17 interconnecting joints. After the model has been customized in shape and proportion to the appearance of its real-world counterpart, in our original approach, we have employed a silhouette-based marker-free motion capture algorithm to make the model attain the same pose as the actor does at each time step of video. The accuracy at which body poses are captured directly influences the visual quality of the rendered free-viewpoint videos. If the model's geometry is not correctly aligned with the person in the real world, our texture generation algorithm (Chap. 8 Sect. 8.8) projects input video frames onto incorrect geometry. This, in turn, leads to ghosting artifacts in the final renderings.

Our silhouette-based motion capture approach faithfully captures even fast and complex body poses. However, slight inaccuracies in the measured poses may exist, which are mainly due to the limited image resolution and the lack of salient shape features on some body parts. The texture information which is available at no additional processing cost helps to correct these pose inaccuracies. In the first step of a predictor-corrector scheme, a set of pose parameters is computed by means of the original silhouette-base pose determination method. In a second step, a corrective 3D motion field is computed by comparing the predicted model appearance to the real video footage (Sect. 9.3.1). From this motion field, corrective pose parameters are computed that are used to update the silhouette-fitted model pose (Sect. 9.3.2).

9.3.1 A Predictor-Corrector Scheme for Hybrid Pose Estimation

Many assumptions in optical flow algorithms, such as the brightness constancy assumption, and the assumption that the visibility does not change in subsequent images, often break down if the motion in the scene is very fast. In rapidly moving scenes, illumination changes may cause strong differences in the appearance of identical surface elements in two subsequent video frames, and occlusions or disocclusions of parts of a scene are very likely to occur. We concede that a purely motion-field based tracking system is suitable for a slowly moving subject only. However, by jointly analyzing optical flow and silhouette information in a predictor-corrector algorithm, it becomes possible to bypass some of the limita-

tions of optical flow reconstruction and capture complex, fast motions of the body. A motion field describes the motion of a scene between two time instants. In contrast, our *corrective motion field* describes the corrective motion between a first pose estimate at one time instant obtained via silhouette based tracking and the correct pose for the same time instant if texture information is taken into account. These motions are small translations and rotations which properly align the model with the input video footage.

Let $I_{j,t}$ be the j-th input camera view at time t, and P_t be the model pose at time t. Then our predictor corrector scheme consists of the following steps:

- With P_t as the starting point, use silhouette fitting to compute $P'_{sil,t+1}$, which is an estimated pose for time $t+1$.
- Generate $I'_{j,t+1}$ by rendering the model from camera j in pose $P'_{sil,t+1}$ with texture from time t.
- **Computation of corrective motion field** D: For each model vertex
  - Determine the projection of the vertex into each camera's image plane.
  - Determine vertex visibility in all cameras by comparing the projected z-coordinate to the OpenGL z-buffer value.
  - If a vertex is visible from camera j, compute the optical flow between images $I'_{j,t+1}$ and $I_{j,t+1}$ by means of the hierarchical Lucas-Kanade approach (see Chap. 2 Sect.2.3.2).
  - If a vertex is visible in at least three camera views (more robust reconstruction than with minimum number of two views), compute a 3D corrective motion vector via the method described in Sect. 9.2.
- Update $P'_{sil,t+1}$ to conform with motion field to yield P_{t+1}.

The computed corrective 3D motion field D describes vertex position updates that correct slight inaccuracies in the result of the silhouette step.

9.3.2 Differential Pose Update from 3D Motion Fields

The corrective motion field D can be used to compute differential pose parameter updates for each limb of the body model. For the root, which is located in the torso segment, three differential rotation and three differential translation parameters are computed. All other joints are purely rotational. This includes 3-DOF rotations for the shoulders, hips, and neck, and a 1-DOF rotation for the elbows and knees. Wrist and ankle joints are currently not considered.

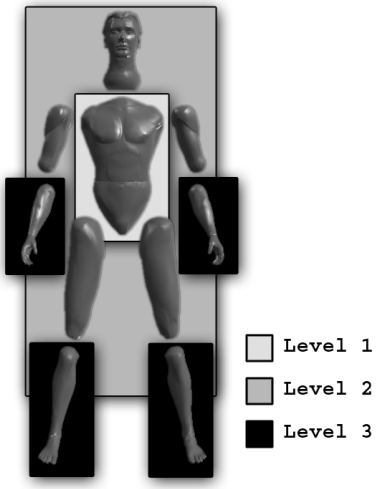


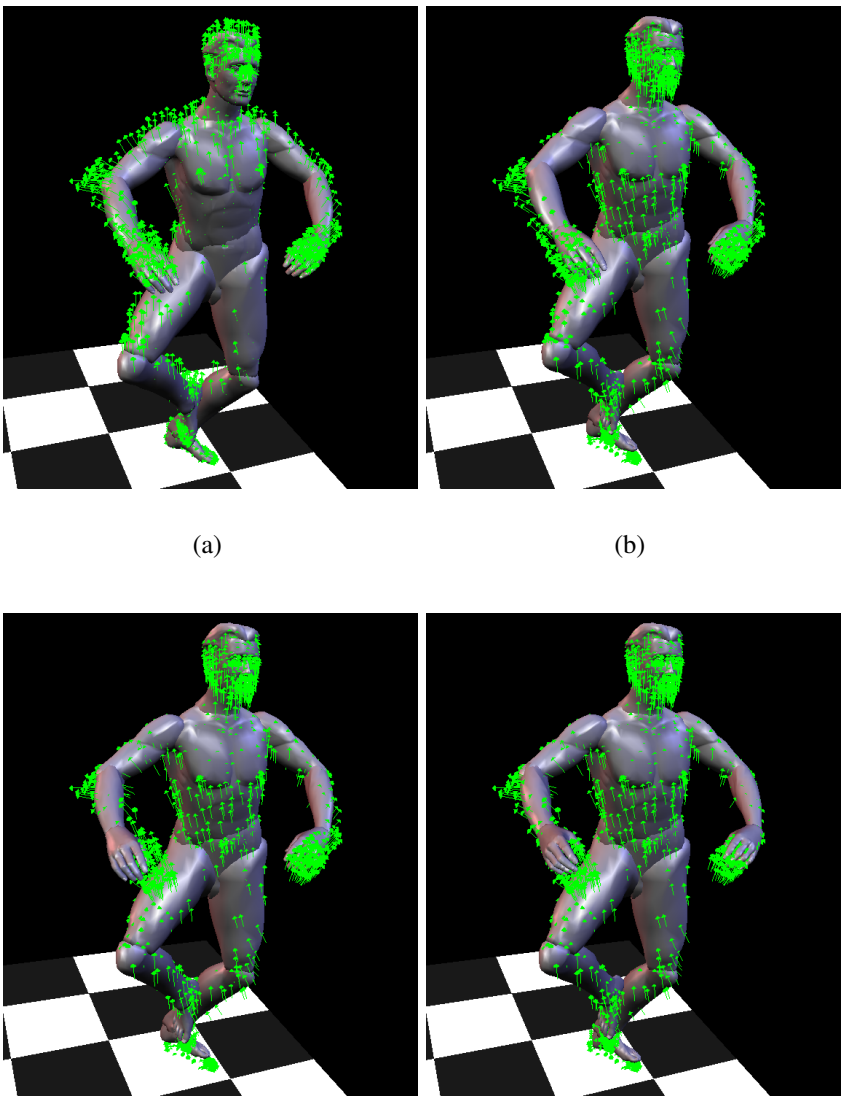
Figure 9.3: Body model showing the separate hierarchy levels.

By adding each vector in D to the current 3D position of its corresponding vertex, a set of goal positions is defined for each model vertex. The goal is to find the set of differential joint parameters of the body model that optimally align the vertices with these positions. The idea is to compute the differential pose parameter updates for every joint only from the goal positions of the vertices of the attached body segment, e.g. using the upper arm goal positions to find the shoulder parameters.

Both our artificial body model and the real human body are hierarchical kinematic chains. We estimate optimal differential pose parameters for one level of the model's hierarchy at a time, proceeding from top to bottom (level 1 being the highest level, see Fig. 9.3). After the pose updates for all body parts one level are found, the model's pose is updated accordingly and the method proceeds with the next lower hierarchy level.

So far we have not answered the question how the corrective rigid body transformation is computed for one individual body segment. Finding a pose update for a joint corresponds to finding a coordinate system transformation between two point sets, a problem know as the *absolute orientation problem* in photogrammetry [Horn86]. For each joint, one point set consists of the current 3D vertex positions of the attached body segment. The second point set defines the goal locations for each vertex in 3D space.

Horn [Horn87] describes a closed form solution to the absolute orientation problem, henceforth referred to as the registration method. In his work, Horn



(a) (b)

(c) (d)

Figure 9.4: (a) Silhouette-only model pose with 3D motion field (little arrows). (b) The model after correction on the first hierarchy level, the second (c) and then the third level (d). The lengths of the motion field vectors are exaggerated.

uses quaternions to parameterize rotations. All transformations are computed with respect to the centers of gravity of both point sets. Let $\mathbf{x}_{1,i}$ and $\mathbf{x}_{2,i}$, $i = \{1,\ldots,n\}$ be corresponding points from two point sets. Then the least-squares solution the absolute orientation problem are the rotation $\mathbf{R}$ and translation $\mathbf{c}$ that minimize the error function

$$\sum_{i}^{n} \parallel \mathbf{x}_{2,i} - \mathbf{R}\mathbf{x}_{1,i} - \mathbf{c} \parallel^2 \tag{9.3}$$

It has been shown that the optimal translation $\mathbf{c}$ is defined by the difference between the centroid of set 2 and the rotated centroid of set 1. To find the optimal rotation, the coordinates of the points in both point sets are defined relative to their center of gravity, respectively. It can be shown that the optimal rotation in the sense of (9.3) can be found by maximizing

$$\sum_{i}^{n} \mathbf{x}_{2,i}\mathbf{R}\mathbf{x}_{1,i} \tag{9.4}$$

The maximal solution to (9.4) can efficiently be computed in closed-form using a quaternion parameterization $\mathbf{q}$ of the rotation (Chap. 2). Using quaternions, the sum (9.4) can be transformed into the form

$$\mathbf{q}^T\mathbf{N}\mathbf{q} \tag{9.5}$$

The matrix $\mathbf{N}$ contains entries that are only made up of products of coordinates of corresponding points in the two point sets that need to be registered. Formally, it is computed from the entries of the scaled covariance matrix $\mathbf{M}$. Let $X_1 = \{(x_{1,i}, y_{1,i}, z_{1,i}) \mid i = 1,\ldots,n\}$ and $X_2 = \{(x_{2,i}, y_{2,i}, z_{2,i}) \mid i = 1,\ldots,n\}$ be two point sets, then $\mathbf{M}$ is defined as follows:

$$\mathbf{M} = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \tag{9.6}$$

with

$$S_{xy} = \sum_{i=1}^{n} x_{1,i} y_{2,i} \tag{9.7}$$

The entries in $\mathbf{N}$ are built via arithmetic operations on elements of $\mathbf{M}$:

$$\mathbf{N} = \begin{bmatrix} N_1 & N_2 & N_3 & N_4 \end{bmatrix}$$

$$N_1 = \begin{bmatrix} (S_{xx}+S_{yy}+S_{zz}) \\ S_{yz}-S_{zy} \\ S_{zx}-S_{xz} \\ S_{xy}-S_{yx} \end{bmatrix}, \qquad N_2 = \begin{bmatrix} S_{yz}-S_{zy} \\ (S_{xx}+S_{yy}+S_{zz}) \\ S_{xy}+S_{yx} \\ S_{zx}+S_{xz} \end{bmatrix},$$

$$N_3 = \begin{bmatrix} S_{zx}-S_{xz} \\ S_{xy}+S_{yx} \\ (-S_{xx}+S_{yy}-S_{zz}) \\ S_{yz}+S_{zy} \end{bmatrix}, \qquad N_4 = \begin{bmatrix} S_{xy}-S_{yx} \\ S_{zx}+S_{xz} \\ S_{yz}+S_{zy} \\ (-S_{xx}-S_{yy}+S_{zz}) \end{bmatrix}$$

The rotation that maximizes the sum (9.5) is derived from the eigenvector that corresponds to the largest eigenvalue of the symmetric 4×4-matrix $\mathbf{N}$. The final solution quaternion $\mathbf{q}$ is a unit vector in the same direction as this eigenvector.

We apply the registration method to compute differential pose updates as follows: The adjustment starts at hierarchy level 1 with the root of the model. To find the corrective model update of the root joint, a differential rotation and translation is computed using the start and end positions of the vertices in the torso segment that have been computed from D.

On the second level of the hierarchy, only differential rotation parameters for 3-DOF shoulder, hip, and head joints need to be computed.

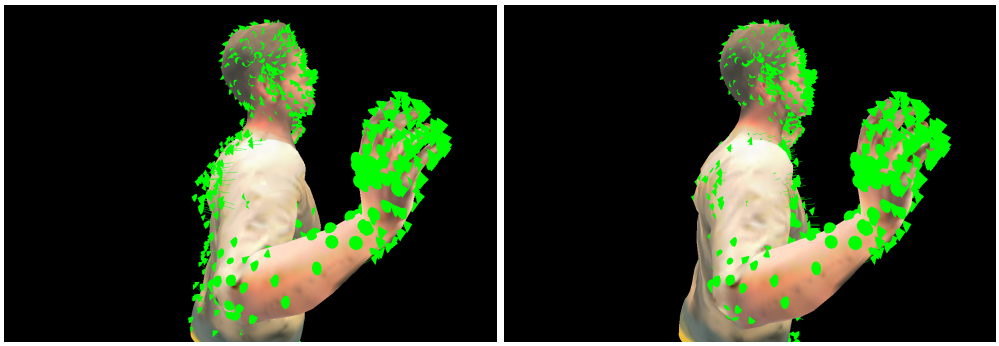
On hierarchy level 3, there are four 1-DOF joints (the elbows and the knees). The body model is designed in such a way that the rotation axis for each of these joints coincides with the x-axis of the local coordinate system. The optimal rotations are found using the same procedure as on hierarchy level 2. The 1-DOF constraint is incorporated by projecting the start and goal vertex positions into the local yz-planes.

It is important to note that for corrective pose update we do not employ the 4-DOF limb parameterization (Chap. 8 Sect. 8.3) as it is used during the silhouette-fitting step. The pose update scheme we employ here considers the shoulder/hip and elbow/knee joints to be on different levels of hierarchy.

In Fig. 9.4 the step-by-step pose correction on different hierarchy levels of the model is illustrated.

9.4 Results and Discussion

The performance of our augmented free-viewpoint video approach has been tested on two multi-view video sequences that were recorded with 8 cameras at a res-



(a) (b)

Figure 9.5: Body model with corrective motion field (green arrows) before (a) and after (b) pose update.

olution of 320x240 pixels and a frame rate of 15 fps. The sequences show simple gestures that exhibit a large amount of head motion which cannot be accurately recovered using only the silhouette step. All test PCs feature 1.8 GHz Pentium^{TM}IV Xeon CPUs with 512 MB of main memory and are equipped with Nvidia GeForce3^{TM} GPUs. For the different sub-components of the motion capture algorithm we obtained the following timing results:

On our two test sequences, the silhouette matching takes between 3s and 5s for a single time step if a single PCs is used. If the parallel implementation with five PCs is applied, per-frame fitting times significantly below one second are achieved.

The run time of the texture-based pose enhancement algorithm is dominated by the time needed to compute the optical flows in all camera views. The hierarchical Lucas Kanade optical flow algorithm takes, on average, 45s for the processing of one set of 8 input views if four levels of an image pyramid and a 20x20 Gaussian window are used. These numbers apply if the algorithm is configured to compute one scene flow vector for each vertex of the model geometry, i.e. one 2D optical flow vector in each camera view that sees a vertex. Speed-ups are gained by reducing the number of image pyramid levels and the size of the Gaussian neighborhood. For only one level in the pyramid and a 10x10-neighborhood, the optical flows in 8 camera views can be computed in about 8s, but the quality of the computed flow field is potentially reduced.

A further acceleration is achieved by computing the scene flows only for a subset of the model's vertices. However, since our focus lies on producing the maximal possible visual quality, we run the scene flow computation at the highest

level of detail. The reconstruction of a three-dimensional corrective motion field from eight 2D optical flow fields takes, on average, 0.34s.

The results we obtain when applying our augmented free-viewpoint video pipeline to both test sequences show that the motion field update step can noticeably improve the quality of the reconstructed 3D videos. In Fig. 9.5a the computed corrective motion field is shown and it is illustrated how the body pose is updated according to it. The two pairs of images in Fig. 9.7 show the textured and untextured body model side-by-side. The top pair shows the result that is obtained with pure silhouette-based motion capture, the bottom pair of images shows the result with the enhanced algorithm. It is clear that the improved visual quality of the textured model, notably in the face, is due to the more accurate body pose.

Fig. 9.6 shows three comparisons between novel viewpoint renderings that were created from 3D videos that were reconstructed with and without activated motion field correction respectively. As we expected, the most obvious improvements are visible in the face and on the torso. The silhouette step often cannot

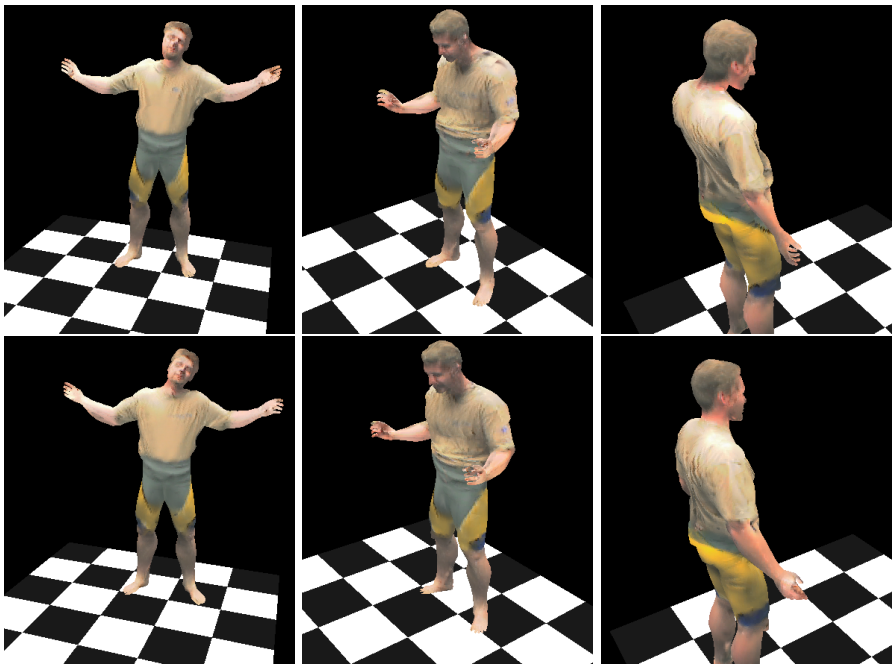


Figure 9.6: The top images show screen-shots of 3D videos that were reconstructed without motion field correction. The bottom images show renditions from the same virtual camera perspective if the texture-based pose enhancement has been applied during reconstruction. Improvements in rendering quality, in particular on the head and the torso, are clearly visible.

	Difference in Avg.	Max. Difference
sequence 1	0.33 dB	0.81 dB
sequence 2	0.35 dB	0.93 dB

Table 9.1: Differences in PSNR measurements between free-viewpoint videos that were reconstructed with and without motion field correction.

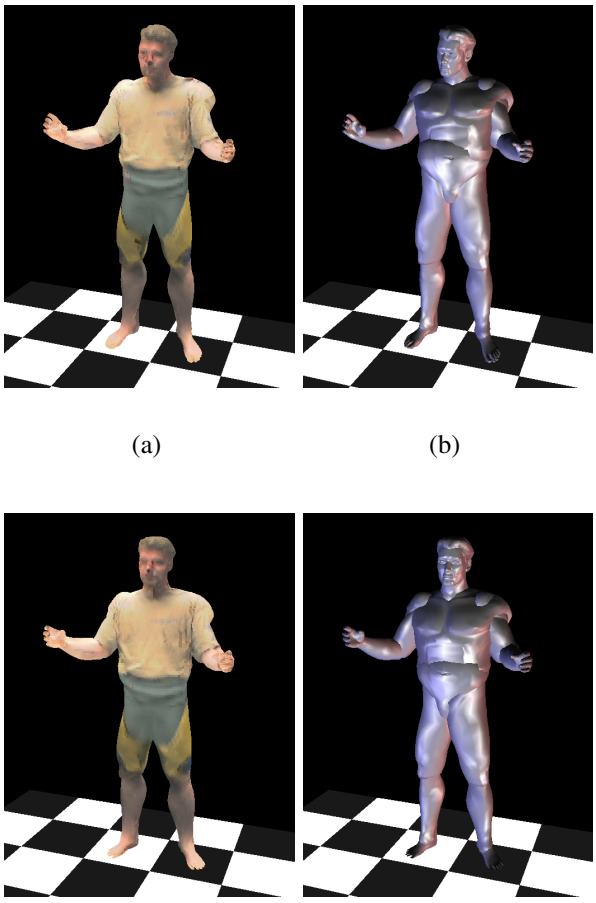
exactly recover the head orientation. Texture information enables our system to automatically correct these errors. Slight changes in torso orientation are also discovered more robustly if the motion field correction step is applied.

In order to validate the visual improvements inflicted by the motion field step quantitatively we employ a quality measure widely used in research on video encoding. For each time step of video we compute the peak signal-to-noise-ratio (PSNR) [Bhaskaran99] in the luminance channel between the 3D video rendered from the input camera perspectives and the segmented recorded input views. On both test sequences, the PSNR is computed for the 3D videos with and without the corrective motion field step.

The difference in the average PSNR between the corrected and uncorrected free-viewpoint videos as well as the maximal observed difference for one single time step of video are summarized in Tab. 9.1. The difference in the average PSNR over all video frames is a measure of reconstruction quality. A positive difference characterizes an improvement of rendering quality with respect to the original video frames. We obtained positive differences between the average PSNRs for both sequences. For one single time step of video the improvements can even be more significant as it is expressed in the values for the maximal observed PSNR difference.

It is interesting to observe that, after only small differences at the beginning, the PSNR differences are larger towards the end of both sequences. This confirms our original assumption that the correction step improves model fitting over time by reducing the impact of propagated pose errors.

In conclusion, we have presented a novel augmented motion capture algorithm for 3D video reconstruction that jointly employs silhouette and texture data for pose determination. It enables us to reconstruct free-viewpoint videos of human actors at very high precision. Subtle pose inaccuracies in the purely silhouette-fitted body poses, which may lead to texturing artifacts in the final renderings, are robustly resolved.



(a) (b)

(c) (d)

Figure 9.7: Top row: Textured (a) and untextured (b) body model in a pose that has been estimated via silhouette-fitting only. Bottom row: Textured (c) and untextured (d) body model if the motion field correction has been applied during reconstruction. The improvement in rendering quality due to an improved pose estimate is clearly visible.

Chapter 10

Joint Motion and Reflectance Capture: Relightable 3D Video

While the ability to realistically display dynamic events from novel viewpoints (Chaps. 8 and 9) has already a number of intriguing applications, the next step is to use such real world-captured objects for augmenting virtual scenes. To import a real-world object into surroundings different from the recording environment, however, its appearance must be adapted to the new illumination situation. Otherwise, the object will have an artificial, "pasted-in" look. To do so, the bi-directional reflectance distribution function (BRDF) must be known for all object surface points. Data-driven [Debevec00, Matusik03] as well as model-based [Marschner98, Lensch03] methods have been proposed to recover and represent the BRDF of real-world materials (see also Chap. 7 Sect. 7.1.4). Unfortunately, these methods cannot be directly applied to dynamic objects exhibiting time-varying surface geometry and constantly changing local illumination.

We present an augmented version of our free-viewpoint video approach that simultaneously captures the time-varying scene geometry as well as the BRDF parameters on the body model of a moving actor [Theobalt05]. As input to our algorithm we require only a handful of calibrated and synchronized video recordings. The algorithm automatically returns subject-adapted 3D geometry, animation parameters, diffuse texture, per-texel BRDF model parameter values, as well as time-varying surface normals. PC graphics hardware-assisted rendering then allows us to photo-realistically visualize recorded people at interactive frame rates in changing lighting conditions and from arbitrary perspective. We present results for several subjects wearing different clothes made of non-diffusely reflecting fabrics, and we show how to augment virtual environments with real world-recorded people.

The following algorithmic contributions are introduced by our novel aug-

mented free-viewpoint video pipeline:

- An algorithm to warp-correct input video images in order to guarantee multi-view photo-consistency in conjunction with inexact object geometry,
- Dynamic reflectometry, i.e.
  - per-texel, per-time step BRDF parameter estimation from multi-view video footage,
  - reconstruction of time-varying normal maps to capture small, variable detail of surface geometry (e.g., wrinkles in clothing), and
- the integration of the recording facilities, the motion capture method, the reflectance estimation approach and the renderer into one working system.

10.1 Overview

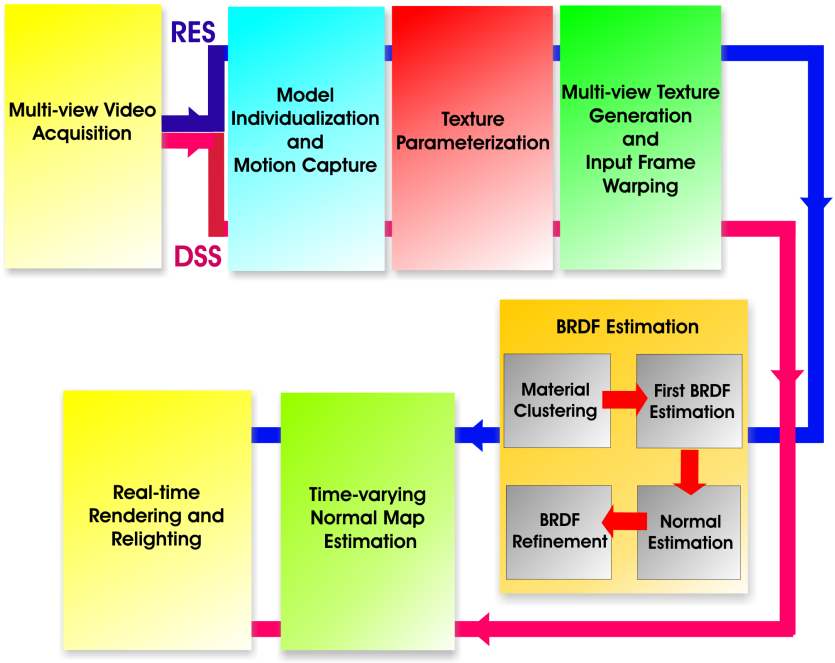


Figure 10.1: Algorithmic workflow joining the individual components of our method.

Fig. 10.1 illustrates the algorithmic workflow within our system for relightable free-viewpoint video reconstruction and rendering. For every person for whom

we reconstruct a relightable 3D video we record two types of multi-view video sequences from eight synchronized video cameras (Sect. 10.2). The reflectance estimation sequence (RES) is used to estimate surface reflectance properties. Arbitrary human motion is captured in the dynamic motion sequences (DSS), and these sequences are later visualized and relit. In both types of sequence, the person wears identical clothes. The respective data paths for both input sequences are shown in Fig. 10.1. Our generic body model is adapted to match the shape and proportions of the recorded person (Chap. 8 Sect. 8.5). Subsequently, human pose parameters are computed for all time frames in both RES and DSS by means our silhouette-based marker-free motion capture approach (Chap. 8 Sect. 8.5). To store all per-surface element data needed during reflectance estimation in texture space, we make use of a texture atlas as surface parameterization of the body model (Sect. 10.3.1). Multi-view video (MVV) textures are generated by transforming each input video image into the texture domain. To correct for photo-inconsistencies due to inexact body geometry, the input images can be warp-corrected prior to MVV texture generation (Sect. 10.3.2). From the RES video data, BRDF model parameter values are estimated for each surface element (texel) of the geometry model individually (Sect. 10.4.1). The recovered local reflectance properties then allow us to estimate the time-varying surface normal field in the DSS sequences (Sect. 10.4.2). The moving body model, its spatially-varying reflectance, and the time-varying normal field enable us to interactively render and instantaneously relight the DSS sequences from arbitrary viewpoint and illumination direction (Sects. 10.5 and 10.6).

10.2 Acquisition

As input to our system, we record multi-view video (MVV) sequences in our studio (Chap. 4). Since we estimate both motion and reflectance properties, we have strict requirements concerning the spatial, temporal, and color resolution of our imaging devices. Only recently, suitable production-line video cameras have become available that meet our requirements. Facing these technological demands, we have decided to upgrade our previously employed IEEE1394 camera setup to a novel setup with more advanced imaging devices. The technical details of evolution II of our camera system are given in Chap. 4 Sect. 4.2.2 but its main features shall be repeated here. The novel setup consists of eight Imperx^{TM}MDC-1004 cameras that feature a 1004x1004 CCD sensor with linear 12 bits-per-pixel resolution. We run the cameras in single-chip mode which means that they provide a sustained frame rate of 25 fps. For the sequences recorded in the course of this project we have placed the cameras in a semi-circular arrangement around the center of the scene (Fig. 10.2). The camera arrangement has to be suitable

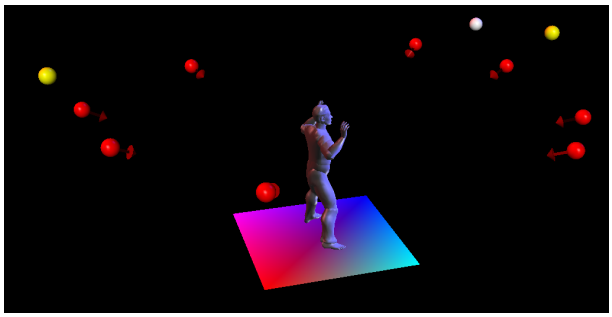


Figure 10.2: Illustration of camera arrangement (red), lighting setup 1 (white) and lighting setup 2 (white+yellow).

both for motion and reflectance estimation. Our semi-circular arrangement is a compromise we found to be suitable for both of these purposes. The cameras are calibrated, and radial and tangential lens distortions are corrected up to second order.

The lighting conditions in our studio are fully controllable. No exterior light can enter the recording area, and the influence of indirect illumination is minimized by covering up all the walls and the studio floor with black cloth and carpet. Two different lighting setups are used. Lighting setup 1 (LS1) illuminates the scene with only one K5600^{TM}Jokerbug 400 spot light. In lighting setup 2 (LS2), additional light sources on the ceiling are used in order to illuminate the set more evenly. In our simulations we approximate the contribution of the single spot light with one point light source and the illumination from the ceiling light with two additional point light sources, Fig.10.2. Light source positions, intensities and color response of the cameras are calibrated offline. Color consistency between multiple camera views is established (Chap. 4 Sect. 4.4.2).

We successively record three MVV sequences for each person and each type of apparel. A short sequence of the scene background recorded with illumination setup LS2 later facilitates color-based background subtraction of the motion sequences. The second sequence, referred to as the reflectance estimation sequence (RES), serves as input to the BRDF estimation algorithm. While BRDF parameter value estimation works best if the scene is illuminated by only one light source (LS1), robust motion capture is practically impossible if large parts of the subject are in shadow. To resolve the conflict, the RES is acquired in single-shot mode. The person strikes the initialization pose, Fig.10.3, and turns between shots by approximately 5° until having completed a full 360°-circle. At each orientation step, a set of eight images is captured for lighting setup 1, and a second set of images is recorded for setup 2, Fig.10.3. The first set is used for BRDF esti-

Figure 10.3: Three pairs of RES images recorded in lighting setup 2 (left) and lighting setup 1 (right).

mation, the second set for recovering body pose. Prior to reflectance estimation we fit our geometry model to each body pose in the RES. For each point on the model's surface, the RES contains as many different appearance samples as there are images depicting the respective point. Over time, the surface element normal points in various directions, and we obtain a large number of reflection samples for our large-scale moving object. While surface normal orientation varies freely, our static camera and lighting setup allows for only a limited number of half vector directions $\vec{h} = \hat{l} \cdot \hat{v}_j/2$, i.e., angular separations between spot light $\hat{l}$ and camera directions $\hat{v}_j$. By placing the cameras non-symmetrically with respect to the spot light, we gather samples for up to eight different light-to-camera angles, which we found sufficient to robustly fit our isotropic BRDF models (Sect. 10.4.1).

Finally, the dynamic scene sequences (DSS) capture the motion sequences from which the actual relightable 3D videos are generated. The scene is now illuminated using lighting setup 2. From the DSS we also reconstruct a time-varying surface normal field (Sect. 10.4.2).

10.3 Texture Generation

Our silhouette-based model fitting approach (Chap. 8) provides us with an appropriately rescaled human body model as well as a set of pose parameters for each time step of the RES and every DSS. Using a model-based approach, we can find a static surface parameterization that allows us to convert the input video images into textures (Sect. 10.3.1). Although our rescaled model is an accurate approximation to the true body geometry of the actor, small discrepancies between the projected model outline and the person's appearance in the video frames still exist. In order to prevent multi-view inconsistencies when the video frames are transformed into texture maps, we resample and align the input streams using a novel warp-correction technique (Sect. 10.3.2).

10.3.1 Texture Parameterization

Each body segment is parameterized separately over a planar rectangular domain using patches of minimal distortion. For the geometry of one body segment our parameterization method works as follows: First a seed triangle is selected from the mesh that initiates a novel patch. Triangles neighboring to the triangles currently contained in the patch are added to it until all remaining insertion candidates show too strong a deviation from a plane fitted to the patch in a least-squares sense. If no further triangle can be added the patch is projected orthographically onto the least-squares plane thereby generating its layout in the 2D texture domain. The procedure continues with a novel see triangle or terminates if all of them have been assigned to a patch. The planar patch layouts for each of the sixteen body segments are finally assembled into one texture atlas for the complete model (Fig. 10.4). This way, we obtain a pose-independent bijective 3D-to-2D mapping between a surface element and a texel in the texture domain. All data related to surface elements (normals, light vectors, visibility etc.) can now be conveniently stored as textures. Throughout our experiments, we use 1024x1024-texel texture maps.

The graphics hardware is used to transform each video camera image into the texture domain. For each video time step, eight so-called multi-view video textures (MVV textures) are created.

(a) (b)

Figure 10.4: (a) Color-coded body segments. (b) Corresponding patch layout in the texture domain obtained with our surface parameterization approach.

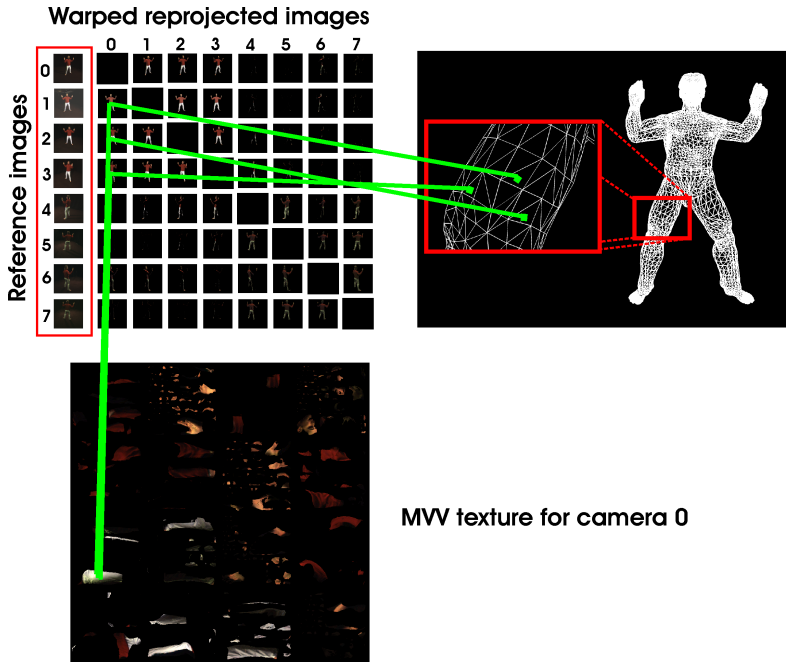


Figure 10.5: MVV texture generation for camera 0: The color information for each surface point on the body model is not looked up in the original input video frame recorded from camera 0. Instead the texel color is taken from an image that has been obtained by reprojecting the model that has been textured with camera image 0 into the camera view that sees the surface point most head-on.

10.3.2 Image-based Warp-Correction

Although the body model initialization procedure yields a good approximation to the person's true geometry, small inaccuracies between the real human and its digital counterpart are inevitable. Due to these geometry inaccuracies, pixels from different input views may get mapped to the same texel position in different MVV textures, even though they do not correspond to the same surface element of the true body geometry. This, in turn, can lead to errors during reflectance estimation.

One common strategy to enhance photo-consistency is to deform the geometry until an overall photo-consistency measure is maximized. Geometry deformation-based optimization, however, tends to give unstable results, e.g., due to sudden visibility changes.

We take an alternative approach. Instead of moving surface elements to their correct locations in 3D, we move the image pixels within the 2D input image planes until they all become photo-consistent given the available geometry.

The following example illustrates our modified MVV texture generation scheme (Fig.10.5):

Let's assume we want to assemble an MVV texture from the video image $I_y(t)$ seen by camera Y at time t. For texel K in the MVV texture, we find out which camera sees it best by searching for the minimal deviation between camera viewing vectors and the surface element normal. If the camera that sees the surface point best is Y, the texel color is taken from $I_Y(t)$. In case camera $X \neq Y$ sees the point best and it is not occluded, we regard the video image $I_x(t)$ as the reference image. The model at time t is projectively textured with $I_Y(t)$ and rendered into camera view X. The image of the reprojected textured model is warped such that it is optimally aligned with the reference image. The color of K is taken from the warped image. This way, all texel color values stem from the same physical camera image. The texel color, however, is always taken from a version of that camera image that has been brought into optimal registration with the camera view that sees the corresponding surface element most head-on.

Warped images are precomputed for all possible combinations of X and Y. In our case this corresponds to 56 warping computations for each time step. To establish per-pixel correspondences, the warping operation itself is based on the optical flow between the reference image and the image of the reprojected textured model, Fig. 10.6. For one pair of a reference image I_R and a reprojected image of the textured model I_M the warping operation works as follows: A regular 2D triangle mesh T with n vertices $\{v_1, \ldots, v_n\}$ is superimposed on I_M. The optical flow between the reference image and the reprojected model image is computed by means of the Lucas-Kanade technique (Chap. 2 Sect. 2.3.2). The so-created flow field describes a displacement for each pixel in I_M that brings it into optimal overlap with its corresponding pixel in I_R. From the per-pixel displacements we compute a globally consistent warping for I_M that brings it into photo-consistent registration with respect to I_R. In order to do this for each vertex v_i in T a 2D displacement vector $\vec{r}_i$ is estimated performing a weighted average on all flow vectors in a rectangular pixel neighborhood around the position of v_i. The triangle mesh is then deformed to globally adapt to the per-vertex displacements by means of a Laplace interpolation (see e.g. [Farin99, Lipman04]). The new mesh configuration approximately satisfies the displacement constraints and also preserves a smooth geometry. Formally, the deformation of the mesh is found by solving the Laplace equation

$$\mathbf{L}\mathbf{x} = 0 \tag{10.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ are the vertex positions and the $n \times n$-Matrix $\mathbf{L}$ is the discrete

Laplace operator [Meyer02] with

$$\mathbf{L}_{ij} = \begin{cases} 4 & \text{if } i \text{ inner vertex and } i = j, \\ -1 & \text{if } i \text{ inner vertex and } j \text{ in its 4-neighborhood,} \\ 0 & \text{else.} \end{cases} \tag{10.2}$$

The matrix $\mathbf{L}$ is singular, and we hence need to add suitable boundary conditions to Eq. 10.1 in order to solve it. We reformulate the problem as

$$\min \left(\begin{pmatrix} \mathbf{L} \\ \mathbf{K} \end{pmatrix} \mathbf{x} - \begin{pmatrix} 0 \\ \mathbf{d} \end{pmatrix} \right)^2 \tag{10.3}$$

This equation is solved in each of the 2 image plane coordinate directions separately. The $n \times n$ matrix $\mathbf{K}$ and $\mathbf{d} \in \mathbb{R}^n$ impose the interpolation conditions which will be satisfied in least-squares sense. $\mathbf{K}$ is a diagonal matrix with

$$\mathbf{K}_{ij} = \begin{cases} w_i & \text{if a displacement is specified for } i, \\ w_i & \text{if } i \text{ is a boundary vertex,} \\ 0 & \text{else.} \end{cases} \tag{10.4}$$

The elements of $\mathbf{d}$ are

$$\mathbf{d}_i = \begin{cases} w_i \cdot (x_{ik} + \vec{r}_{ik}) & \text{if a displacement is specified for } i, \\ w_i \cdot x_{ik} & \text{if } i \text{ is a boundary vertex,} \\ 0 & \text{else.} \end{cases} \tag{10.5}$$

The values w_i are constraint weights, x_{ik} is the k-th position coordinate of vertex i before deformation, and $\vec{r}_{ik}$ is the k-th coordinate of the displacement for vertex i. The least-squares solution to Eq. 10.3 corresponds to the solution of the linear system

$$\begin{pmatrix} \mathbf{L} \\ \mathbf{K} \end{pmatrix}^T \begin{pmatrix} \mathbf{L} \\ \mathbf{K} \end{pmatrix} \mathbf{x} = (\mathbf{L}^2 + \mathbf{K}^2)\mathbf{x} = \begin{pmatrix} \mathbf{L} \\ \mathbf{K} \end{pmatrix}^T \mathbf{d}. \tag{10.6}$$

Appropriate weights for the displacement constraints are straightforwardly found through experiments.

Finally, the warped reprojected image, $I_{M,warped}$, is created on the GPU by rendering the deformed mesh into a floating point buffer using texture and texture coordinates from the unwarped image I_M. Sometimes better results are obtained by recursively applying the warping procedure. Typically, after three iterations a convergence is achieved.

The warping-based MVV texture assembly is an optional step that is activated if geometry inaccuracies are apparent. The difference images shown in Figs. 10.11a,b proof that the image-based warp-correction yields a better

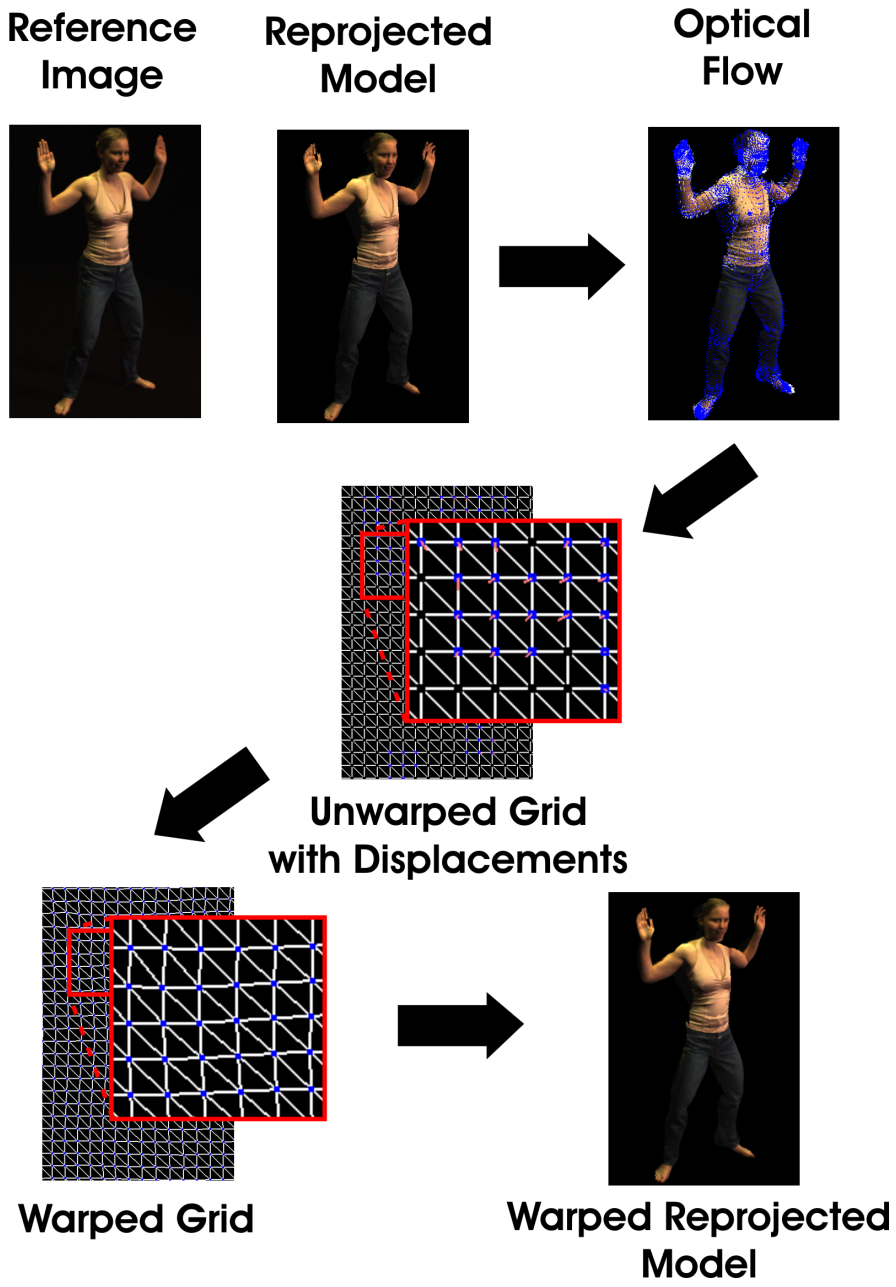


Figure 10.6: Illustration of the individual warp-correction steps for one pair of reference image and reprojected model (textured with one input image only). First, the optical flow between reference and reprojected image is computed. Thereafter, per-vertex displacements for the mesh's vertices are computed and it is globally warped to align with the constraints. The final warped reprojected image of the model is generated on the GPU.

registration between the model and the image data without having to resolve to an error-prone deformation of the model geometry. One might argue that optical flow is based on the assumption that all surfaces in the scene are diffuse. For reflectance estimation, though, we deliberately generate specular highlights in the images. Our experiments show that the method nonetheless produces good results since in most input frames the diffuse reflectance is predominant.

10.4 Dynamic Reflectometry

Our reflectance estimation approach consists of two steps. In the first step we determine BRDF parameter values per texel from the reflectance estimation sequence. An iterative estimation process enables us to handle geometry inconsistencies between the real object and the much smoother human body model. In the second step we compute even time-varying normal maps per frame to capture surface detail such as wrinkles in clothing whose shape and extend depend on the current pose of the person. The underlying technique is similar to [Lensch03] which we have extended in order to cope with multiple light sources, time-varying data, and inter-frame consistency.

10.4.1 BRDF Estimation

We estimate a set of spatially-varying BRDFs for each person and each outfit from the respective reflectance estimation sequence (RES) explained in Sect. 10.2. The pose parameters for the RES have been determined beforehand. The goal is to estimate a separate parametric reflectance model for each surface element that is able to faithfully reproduce the appearance in each camera view and at each time step of the multi-video sequence. For each surface element, the BRDF representation consists of an individual diffuse color component that is specific to the surface point, and a set of specular parameters that are shared by all surface points belonging to the same material. Our framework is flexible enough to incorporate any parametric reflectance model. However, in the majority of our experiments we employ the parametric BRDF model proposed by Phong [Phong75]. We have also tested our method with the model proposed by Lafortune [Lafortune97], using two specular lobes (see also Chap. 2 Sect. 2.1.2).

In general, our estimation of BRDF parameters and later the estimation of the time-varying normals is based on minimizing for each surface point $\vec{x}$ the error $E(\vec{x}, \rho(\vec{x}))$ between the current model $\rho(\vec{x})$ and the measurements for this point from all cameras c at all time steps t:

$$E(\vec{x},\rho(\vec{x})) = \sum_{t}^{N}\sum_{c}^{8}\kappa_c(t)\Bigg(S_c(t) - [\sum_{j}^{J}\lambda_j(t)(f_r(\hat{l}(t),\hat{v}_c(t),\rho(\vec{x})) \cdot I_j(\hat{n}(t)\cdot\hat{l}(t)))]\Bigg)^2. \tag{10.7}$$

The term is evaluated separately in the red, green and blue color channel. $S_c(t)$ denotes the measured color samples at $\vec{x}$ from camera c, and I_j denotes the intensity of light source j. For BRDF estimation the number of light sources equals one (lighting setup 1). More light sources are used when the same energy functional is employed during time-varying normal estimation (Sect. 10.4.2). The hemispherical viewing directions $\hat{v}_c(t)$ and light source directions $\hat{l}_j(t)$ are expressed in the point's local coordinate frame based on the surface normal $\hat{n}(t)$. Visibility of the surface point with respect to each camera is given by $\kappa_c(t)$ and with respect to the light sources by λ_j, both being either 0 or 1. f_r finally evaluates the BRDF. All information that is relevant for one texel thus can be grouped into an implicit data structure we called *dynamic texel* or *dyxel*:

$$\begin{aligned} Dyx(\vec{x},t) = &[S_1(t),\ldots,S_8(t),\hat{v}_1(t),\ldots,\hat{v}_8(t),\\ &\hat{n}(t),\hat{l}(t),\kappa_1(t),\ldots,\kappa_8(t),\lambda_1(t),\ldots,\lambda_J(t)]. \end{aligned}$$

Using a non-linear optimization this formula in principle could be used to determine a full BRDF and the surface normal at the same time. However, we applied an iterative approach and carefully designed the reflectance estimation sequence to obtain a much more stable optimization. For example we use only a single light source during the RES. The subsequent steps of our iterative BRDF estimation scheme are *material clustering, first BRDF estimation, normal estimation* and *refined BRDF estimation*, as depicted in Fig. 10.7.

Instead of determining the specular part of the BRDF per pixel we assume that there is only very little variation of the specular part within the same material, e.g. skin, hair or the different fabrics. By combining the measurements of multiple surface points exhibiting the same material we increase the number of samples and more importantly the variation in viewing and lighting directions in order to obtain a more faithful specular estimate. The *clustering* step determines to what material a surface element, i.e., each texel in the texture atlas, belongs. The number of materials is determined a priori. We employ a straightforward color-based clustering approach that considers the raw texel color values. The clustering output is a material texture map in which each texel is assigned a material label, Fig. 10.7.

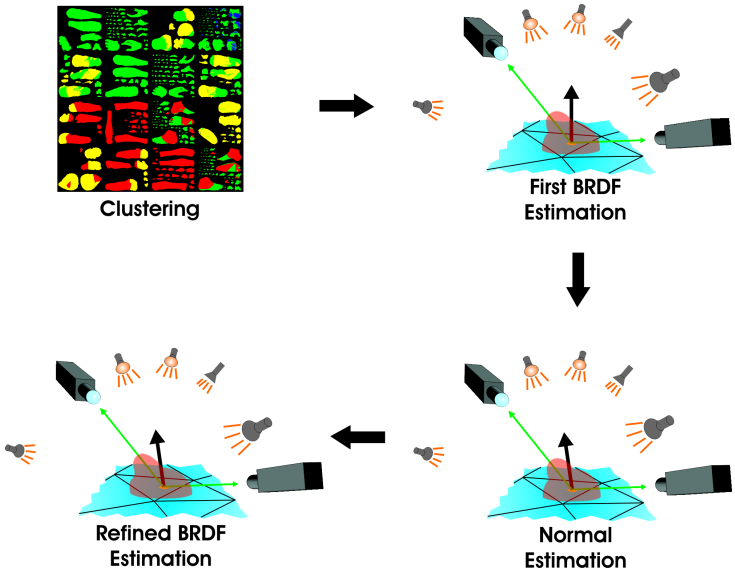


Figure 10.7: Subsequent steps to derive per-texel BRDF.

During the *first BRDF estimation*, an optimal set of per-texel BRDF parameters is determined while the normals are taken from the default geometry. The estimation itself consists of a non-linear minimization of Eq. 10.7 in the BRDF parameters. For optimization, we make use of a Levenberg-Marquardt minimization scheme [Press02] in the same manner as [Lensch03]. First, we find an optimal set of parameter values for each material cluster of texels. To quantify the estimation error per material cluster, we sum the error term in Eq.10.7 for all surface elements that belong to the cluster. Given the average BRDF for each material, we can render the model by applying only average specular reflectances. By subtracting this specular component from each sample, we generate new dyxels that contain purely diffuse reflectance samples. Using these purely diffuse samples, an individual diffuse component is estimated for each surface element (texel) by minimizing Eq.10.7 over the diffuse color parameter. The output of the first BRDF estimation is then a set of spatially-varying BRDF parameters ρ_{first}.

The default normals of the human body model cannot represent subtle details in surface geometry, such as wrinkles in clothing. In a *normal estimation* step, we make use of the first set of estimated BRDF parameters ρ_{first} in order to reconstruct a refined normal field via photometric stereo. In order to make this reconstruction tractable, we implicitly assume that the local normal directions do not change while the person is rotating in place. We found that normal estimation robustness is improved if the error function (Eq. 10.7) is extended into

$$E_{normal}(\vec{x},\rho(\vec{x})) = \alpha E(\vec{x},\rho(\vec{x})) + \beta\Delta(\hat{n})^{\gamma}. \tag{10.8}$$

The additional term $\Delta(\hat{n})$ penalizes angular deviation from the default normal of the body model. The terms α and β are weighting factors summing to one, the exponent γ controls the penalty impact. Appropriate values are found through experiments. Normal estimation robustness is further improved if only those color samples in a dyxel are used that come from the two best camera views. For each texel, the modified error function is now minimized by varying the local normal direction $\hat{n}$.

The refined normal field is used for a *second BRDF estimation*. The same computations as for the first BRDF estimation, ρ_{first}, are performed, but now with the more accurate normal field. By this means, we obtain the final set of per-texel BRDF parameters ρ_{final}.

The results are stored in parameter texture maps. For the Phong model, we obtain one texture map containing the per-texel diffuse component, and two texture maps that store the per-material specular colors and exponents. In case of the Lafortune model, the number of specular parameter maps depends on the number of specular lobes.

10.4.2 Time-varying Normal Map Estimation

The BRDF reconstructed in the previous step enables us to relight any dynamic scene in which the person wears the same apparel as in the respective RES. To generate a visually compelling rendition, however, we found that we need not only accurate reflectance, but also a representation of the small surface geometry details that appear and disappear while a person is moving. We are able to capture these geometry details by estimating a time-varying surface normal field for each DSS via photometric stereo.

Motion parameters for the DSS are found by means of our silhouette-based tracking approach. The video frames show the scene illuminated by lighting setup 2. During the estimation, we approximate the incident illumination with three point light sources,

The Time-varying normal direction is estimated for each surface point individually. We assume that the transverse motion of the cloth on the body is negligible, and, in consequence, that over time an MVV texel always corresponds to the exact same cloth surface point. The estimation procedure is a non-linear minimization of the regularized energy function, Eq. 10.8, in the normal direction. During optimization, the BRDF parameters for each surface element are taken from the parameter textures estimated from the corresponding RES.

In order to robustly perform photometric stereo and to minimize the influence of measurement noise, a sufficient number of samples has to be collected for each surface point. To serve this purpose, we assume that changes in local normal direction within a short window in time can be neglected. This way, all samples

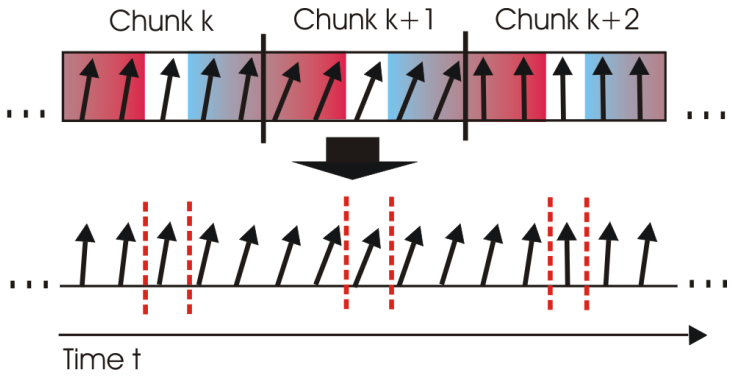


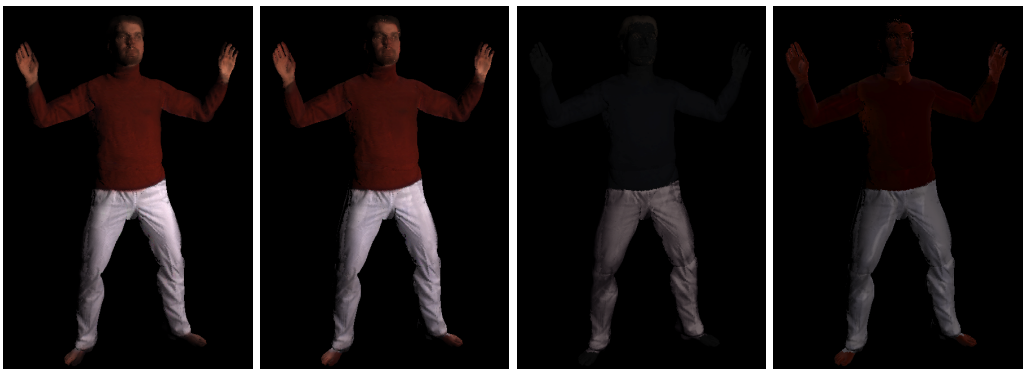
Figure 10.8: 2D illustration of robust time-varying normal map estimation. Top: The sequence subdivided into short chunks. For each chunk, one best-matching normal is derived per texel which is assigned to the chunk's center time step (white). Intermediate time steps are interpolated (bottom).

for a surface point that are taken from a chunk of subsequent time steps in the input footage can be applied to infer a single normal direction. The input sequence of length N is therefore split into C subsequent chunks of odd length d, the last chunk being allowed a different length. Typically, the chunk size is $d = 5$ time steps. For every point $\vec{x}$ on the body surface we fit an optimal normal $\hat{n}$ to each chunk of video individually. After the time-varying normals have been estimated at this coarse scale, the normal directions between subsequent chunks are interpolated via spherical linear interpolation, Fig.10.8.

This way, a normal field is generated that represents a compromise between smoothness in the temporal domain and local normal accuracy. It faithfully models subtle details in surface structure, and it exhibits no normal discontinuities at chunk boundaries that would appear as flickering in the final renditions of the 3D video. The results we obtain with this approach confirm that it is permissible to assume that during a sufficiently small time period the local normal direction does not change dramatically. A comparison to the video footage shows that we are able to capture even the subtle wrinkles that are due to limb bending, Fig. 10.12a,b,c.

10.5 Rendering

The output of our approach is a relightable dynamic object description that consists of the animated geometry and the material properties. The geometry is comprised of the 3D body model mesh, the underlying skeleton and the joints' motion parameters. The material properties consist of the time-independent BRDF



(a) (b)

Figure 10.9: (a) Person rendered with Phong (left) and Lafortune (right) model while being illuminated by one light source. (b) Only specular component rendered for Phong (left) and Lafortune (right) model under the same lighting conditions.

textures and the dynamic normal maps. The number of BRDF data parameters depends on the employed reflectance model. In the case of Phong we store a floating point diffuse component, a specular component, and a specular exponent for all color channels in each texel. The normal maps are represented as vectors in the tangent space of the triangles, where (0,0,1) represents an unaltered triangle normal (see Fig. 10.12a).

For the rendering we extend the human animation system in [Carranza03]. After reading the customized human character model and preparing the static BRDF textures, real-time rendering can commence. For each time step, we now read the pose parameters from the stream and apply the respective rigid transformations to the body model. The player also loads the fitting normal map. The final outlook is now determined by the shader programs, which use similar techniques as in [Fernando04] to perform per-fragment lighting computations with BRDF textures and normal maps. On a 3.0 GHz Pentium 4 and an Nvidia GeForce^{TM} 6800 graphics board, we achieve 25 fps sustained rendering frame rate at 1024×1024-pixel resolution while illuminating the scene with three moving light sources.

To better demonstrate relighting effects while articulated body motion is performed in the scene, we have decided to illuminate the 3D video with point lights, so that the viewer can see the light source positions and corresponding shadows on the floor. Since we use high-level Cg shaders [Mark03], our system can be switched to different parametric reflectance models with low effort. We currently can demonstrate Phong and Lafortune model implementations. A comparison be-

Figure 10.10: Dynamic reclothing.

tween renderings with a two-lobe Lafortune model and the Phong model is shown in Fig. 10.9. The results shown in Figs. 10.10, 10.11c,d and 10.12 have been generated using Phong reflectance.

Figs.10.12a show that wrinkles in the apparel are faithfully identified and represented in the normal maps. Under varying illumination, the wrinkles are realistically rendered, Figs.10.12b. Figs.10.12c show rendered images of the trousers at three consecutive time steps, illustrating the dynamic nature of the normal maps employed by the renderer. Small rendering artifacts are noticeable that are due to texture resampling.

10.6 Results and Discussion

For validation, we have five different sequences of a male and a female subject available. Each sequence is between 50 and 250 frames long. Unfortunately, ground truth BRDF data and normal maps are not at our disposal. Thus, we assess the estimation accuracy in both cases by means of visual comparison to the actual video footage. We found that our method is capable of nicely reproducing the appearance of the actor in the video frames.

Our BRDF estimation approach captures surface reflectance characteristics of different materials simultaneously, as seen in the renderings of Figs.10.12d,e. The animated male and female models are accurately relit for illumination conditions very different from the recording setup. The approach reliably discriminates between diffuse and specular reflectance. The realistically reproduced specular reflection of the trousers of the male model is shown in the accompanying video.

Once we have estimated the BRDF for one type of clothing, we can also use the surface appearance description to change the apparel of a person even for motion sequences in which the person was originally dressed differently. Fig.10.10 depicts an example of dynamic reclothing.

The entire estimation process including motion capture and reflectance estimation takes approximately three minutes per time step. Optional input frame warping takes around 10 seconds for one pair of reference image and reprojected image. We assess the multi-view warping quality by comparing the image differences between reference views and reprojected model views before and after the warp. Typically, we achieve an average reduction in absolute image difference in the range of 6% over a whole sequence (Fig. 10.11a,b). The local registration improvements in single image pairs lead to a global improvement in multi-view texture-to-model consistency. In Fig. 10.11c,d the texture registration improvement due to the warp-correction step is demonstrated. However, in some rare cases local deteriorations in the final texture can be observed despite an improvement on the global level. The decision if the warp-correction is applied is thus left to the user.

Our method is subject to a couple of limitations: First, our method is based on the assumption that interreflections on the body surface can be neglected. In the RES, interreflections potentially play a role between the wrinkles in clothing. To prevent this effect from degrading the estimation accuracy, we have taken care to minimize the number of wrinkles in the RES.

Another limitation of our approach is that visual quality deteriorates if the fabric shifts substantially across the body. Furthermore, we cannot account for loose apparel whose surface can deviate almost arbitrarily from the body model.

For some body poses, rendering artifacts due to undersampling may occur. Especially the lower side of the arms sometimes can not be seen by any of the cameras and thus the true normal directions cannot be inferred. Additional appropriately positioned imaging sensors would solve this problem.

Finally, we intend to employ a single-skin surface model instead of our current segmented one in the future. With the current body representation, occlusions of parts of the surface geometry in the RES complicate the reflectance and normal estimation processes. If a surface point on the model is never seen by any camera, we cannot reconstruct its reflectance. In that case, we interpolate missing parts in the BRDF textures from neighboring regions. However, discontinuities in the texture when frequently occluded surface patches suddenly appear may still be visible. Alternatively, recording the person in more than one body pose can solve that problem already during acquisition. Moreover, if the face geometry of the template model is too different from the shape of the real actor's face, blurring artifacts occur in the final rendering. One possibility to solve this would be to precede the reflectance estimation with a face model reconstruction from high-resolution images of the head. We'd like to emphasize that all limitations inflicted by our specific body geometry are not principal limitations of our method.

Our results demonstrate that we have developed an effective novel method

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

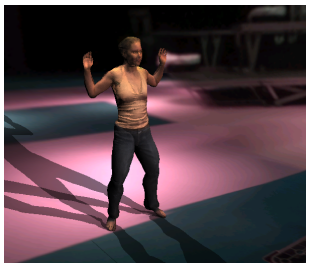
Figure 10.11: (a),(b): Absolute difference image between reference view and reprojected textured model before (a) and after (b) warp-correction. The darker a pixel, the lower the difference. It is clearly visible that, after warp-correction, the match of reprojected model and reference is much better. Larger gray areas that are still visible stem from those parts of the body that have not been seen by the one camera used for texturing the model. They are not due to erroneous registration.
(c),(d): Magnification of the lower leg of the rendered person. (c) Result without warp-correction prior to reflectance estimation - ghosting due to misalignments along the stripes of the trousers are visible. (d) Result with warp-correction - ghosting artifacts have been significantly reduced due to better multi-view consistency. Block artifacts are due to limited texture resolution.

for simultaneous capture of dynamic scene geometry, per-texel BRDFs and time-varying normal maps from multi-view video. The acquired scene description enables realistic real-time rendition of 3D videos under arbitrary novel lighting conditions. This way, we can convincingly implant virtual people into arbitrary novel surroundings. Joint motion and reflectance capture can be applied not only to humans but to any dynamic object whose motion is described by a kinematic chain and for which a suitably parameterized geometry model is available. For BRDF parameter recovery, the proposed algorithm currently assumes that the subject is illuminated by one point light source. While this setup has been chosen to maximize observed reflection variations, the approach can be extended towards more general illumination configurations captured, e.g., via HDR environment maps. To overcome the fixed relationship between light and camera direction, alternatively, a number of spotlights may be applied that are switched on and off during acquisition to illuminate the person sequentially from different directions.

(a)

(b)

(c)



(d)

(e)

Figure 10.12: (a) Color-coded normal map in local coordinates (left) and corresponding input video frame (right). The default normal in the tangent frame is the vector (0,0,1) which translates into a purely blue pixel in the local normal map. Normals deviating from the default one, e.g. due to wrinkles, appear in a different color. (b) Wrinkles on T-shirt rendered under different illumination conditions. (c) Rendered time-varying wrinkles in pants. (d) Single pose relit with different light positions. (e) Person rendered from different viewpoints and illuminations (colored dots: light source positions, colors are light source colors).

Part III

High-Speed Motion Estimation - Exploring the Limits of Photo Camera Technology

Chapter 11

Capturing High-Speed Scenes for Immersive 3D Media

In parts I and II of this thesis, we have described algorithmic solutions to the problems of full-body human motion analysis and free-viewpoint video reconstruction from image data. We have demonstrated that our algorithms can serve as building blocks in the development of novel immersive 3D media, which offer more to a viewer than a plain 2D depiction of a real world scene. All the methods presented so far have employed standard video cameras to acquire the input image material. However, there exists a limit to the speed in a moving scene above which no faithful reconstruction is feasible with standard imaging equipment.

Visual perception of very rapid events, e.g. the motion of a ball in tennis or baseball, or the motion of a bullet, is far beyond the capabilities of the human visual system and standard video cameras. Nonetheless, many application areas exist, in which exact knowledge about such rapid motion is of high importance. As an example, a sports coach who exactly knows how an athlete hit a ball with a racket can decide much more effectively on how the motion cycle can be improved. But also the sports enthusiast at home can profit from novel forms of visualization that become possible after an exact 3D reconstruction of the rapid dynamic scene has been performed. The previous two examples are just two out of a variety of application areas which demonstrate that systems and algorithms for accurate capture of rapid motion are also an important component that contributes to the creation of the next generation of visual media.

The problem of developing methods for capturing images of high-speed real world scenes for analysis of the underlying dynamics has attracted the attention of researchers already decades ago. Back in 1878, Eadweard Muybridge conducted his famous experiments to create serial images of fast motion [Muybridge87]. A setup of twelve cameras was used to capture different stages of a galloping horse.

One of the photographs indeed showed the horse with all of its hooves off the ground, corroborating the hypothesis that had led to these experiments. In the 1930's, Harold E. Edgerton at MIT perfected the use of stroboscope photography to create multi-exposure images of high-speed motion, see for instance [Bruce94]. However, the acquisition process is usually constrained to actions taking place in a very limited spatial domain for which decent illumination conditions can be set up easily. Today high frame rate video cameras are available that enable recording at several hundred frames per second. Nonetheless, the practical application of these cameras is very cumbersome due to technical limitations and high costs. Recording with a high frame rate imaging device produces an immense amount of data which renders the acquisition of video clips of more than a few seconds of duration virtually impossible. The application of these specialized video devices is further complicated by the very short exposure times. Usable image data can only be recorded under very intensive scene illumination. This lighting constraint makes recordings on a larger set very challenging, if not impossible.

In this part of the thesis, we present our answer to the question if it is possible to optically estimate motion data of large-scale high-speed scenes without having to resort to specialized high frame rate cameras. In Chap. 12, we present a novel principle for capturing motion data in such a type of scene [Theobalt04a]. It is based on low-cost commodity still cameras and the principle of multi-exposure photography. In addition to keeping the costs low, the application of off-the-shelf digital cameras enables us to take advantage of their high-resolution imaging sensors. Furthermore, the data handling overhead for processing still images is marginal compared to the overhead necessary for the huge data streams obtained through high-speed video recordings. Our novel algorithmic framework is a general recipe that can be applied to a variety of scenes. However, for demonstrating its applicability and accuracy we have applied it to capture important motion data during a baseball pitch. Besides the popularity of baseball, there are several reasons for our choice. First of all, the underlying motion is very fast and extends over a large area of space: the speed of a pitched baseball can reach 80 mph and above, and the distance from the pitcher mound to the home base is 60.5 feet (18.44 meters). In addition, there are many different motion parameters that we would like to measure simultaneously for a variety of pitches:

- 3D positions along the trajectory of the flying ball;
- initial flight parameters of the ball: norm and direction of initial velocity, rotation axis, spin frequency;
- Poses of the pitcher's hand before, at, and after releasing the ball.

Finally, it is possible to use a physically based model to analyze the consistency of the acquired data: if the ball's initial parameters and flight positions are recon-

structed with high accuracy, they should match the results from a physically based model that predicts the flight trajectory of a spinning ball traveling through air. In summary, both the pitching and flight of a baseball turn out to be a challenging and adequate type of motion for our motion capture approach alike.

In the remainder of this chapter, we will give some background that aids in understanding the next chapter. We begin with a quick look at general optical acquisition principles for high velocity scenes in Sect. 11.1.1. Thereafter, in Sect. 11.1.2 we briefly review approaches from the literature for automatic video-based analysis of sports events . We then describe different algorithmic approaches from the literature for capturing and anlyzing hand motion in Sect. 11.1.3. The chapter concludes with a primer on the technical elements of baseball and the physics of a ball flying through the air in Sect. 11.1.4.

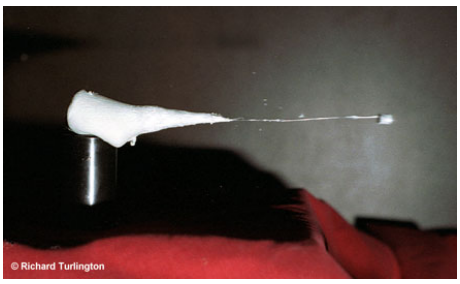
11.1 Background

11.1.1 High-speed Imaging and the Principle of Multi-Exposure Photography

Optical acquisition of very rapid events is a very challenging problem. However, image and video data of high-speed scenes are important sources of information in many areas of application since they enable to analyze the kinematic and dynamic nature of a scene. One example is the analysis of car accidents where detailed information about the deformation of the car's body and the strain on the occupants during the impact is eventually used to improve the car's overall security. A second example is the biomechanical analysis of athletic motion. A detailed insight into an athlete's movements or the motion of a a ball that he is playing can assist the coach in making the right suggestions for improvements. While it is already far beyond the capabilities of the human eye to perceive such fast events, also the technical limits of standard photo and video cameras are exceeded.

Today, high-speed video cameras are commercially available that are capable of recording high-speed motion events at frame rates of up to 10000 fps (e.g. [wei]). This high temporal sampling rate comes at the cost of a very short exposure time for each video frame. At a frame rate of 10000 fps the the shutter of the camera can at best be open for 0.1 ms per frame. Unfortunately, this theoretical value is never reached in reality since in each frame a service time interval for reading out the image sensor needs to be allocated. For capturing useable images, it is thus of significant importance that the recorded scene is illuminated by a very intensive light source. Typically, appropriate illumination conditions can only be created in a very confined volume in space, a fact which makes recordings on larger sets a difficult undertaking. Although a longer integration makes the

(a) (b)



(c)

Figure 11.1: (a),(b): Two (non-subsequent) frames of a high-speed video depicting a motorcycle crash [wei]. (c): Open flash photograph by Richard Burlington showing a bullet that penetrates a marshmallow.

generation of appropriate lighting conditions easier, it also leads to motion blur that may smear out important details in the image plane. Hence, it is often a very tedious task to find camera settings that lead to a decent compromise between image brightness and blurriness. A few example frames recorded with a high frame rate video camera are shown in Fig. 11.1a,b.

In some cases it is not necessary to have a sequence of images documenting the whole course of action, but it is sufficient to have an image of one particular time instant. In this case, one can take advantage of high-speed photography which allows capturing one single image of a rapidly moving scene. To this end, a photo camera and a light source have to be triggered at the exact same moment in time. A common technique used in high-speed photography is the so-called *open flash* [hiw]. When taking an open flash photograph, the environment lighting is completely dimmed and the shutter of the camera is kept open for a long period. At the exact moment in time a high intensity short-impulse flash light illuminates

the scene. Typically, the flash is triggered by a light barrier or by a sound detector. In Fig. 11.1c an example of an open flash photograph is shown.

A high-speed photograph cannot document the temporal evolution of dynamic event, but another photographic technique can achieve this. In the 1930's, Harold E. Edgerton at MIT brought the use of the electronic flash as an artificial light source for still image photography to perfection [Bruce94]. Edgerton enhanced the original electronic flash into a device called stroboscope that can emit many short light pulses at a high frequency. He demonstrated that this novel light source enables capturing several snapshots of a very rapid event in one single image frame. The idea is to use a photo camera which is set to a very long exposure time and illuminate the scene with the stroboscope. This way, a so-called multi-exposure image is created that shows several time instants of the moving scene superimposed in a single frame. Example photographs taken by Harold Edgerton are shown in Fig. 11.2. In addition to the artistic value that many people attribute to these pictures, they also contain a large amount of information about the motion in the scene.

Unfortunately, up to now multi-exposure photographs have hardly been considered as input data for a motion capture method. The algorithm presented in Chap. 12 bridges this gap.

11.1.2 Image-based Analysis and Interpretation of Sports Events

Today, novel high-speed video technology in combination with ever faster computing hardware makes detailed video- or image-based analysis of sports events feasible [Wang03]. For athletes and their coaches, as well as for the sports enthusiast watching a sports broadcast on TV, this novel technology means a great benefit. For the coaches it is highly interesting to obtain technical support in tactical analysis, and the analysis of the athlete's movements. Furthermore, optical

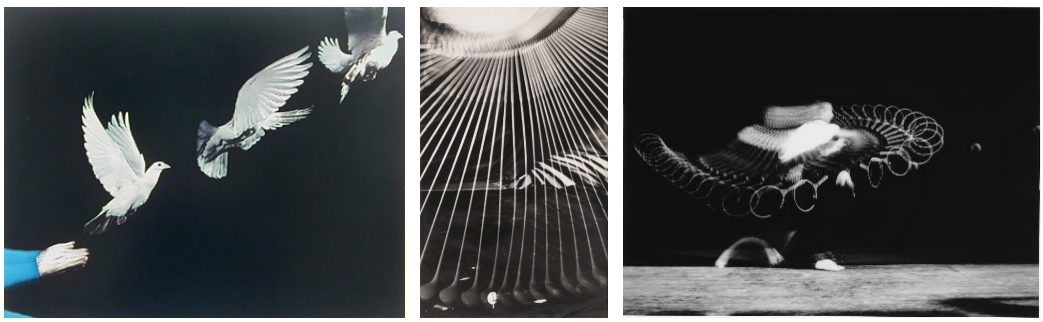


Figure 11.2: Multi-exposure photographs taken by Harold Edgerton.

motion analysis of sports events enables the delivery of entirely new forms of visualization to the viewer at home via TV.

Each sports discipline possesses its characteristic athletic elements, and thus individual tracking solutions need to be developed for each of them. Some of the world's most popular sports are ball games, and hence it is no wonder that the problem of tracking the path of a ball from video images has gained a lot of attention in the research community. Different ball games differ in the way how the ball is played, what size and shape the ball has, and how rapidly and how far the ball typically moves. In consequence, there is not a single multi-purpose tracking scheme for ball motion, but specific solutions need to be developed for each case individually.

One of the most popular sports worldwide is soccer, and there has been a great ambition from the computer vision community to develop ball identification and tracking solutions. In [D'Orazio02] a novel method for identifying the soccer ball in images is presented. It uses a modified version of the Circular Hough Transform (CHT) [Ballard81] to find the position of the ball in the image plane. The authors have designed an adaptive kernel for their matching filter which enables the identification of the ball even if it is partially occluded.

In [Yu03] multiple candidate regions in the image plane are tracked and the decision which of them corresponds to the ball is taken based on a trajectory analysis. This way, the authors try to circumvent classical problems that make ball identification in soccer videos hard, such as the frequent occlusions or the merging with other objects. The algorithm first estimates the center ellipse of the soccer field and the goal mouth. Ball region candidates are identified by looking at salient features such as color, roundness, or the distance to other objects in the image plane. A set of candidate trajectories is obtained, each of which is classified using a trajectory plausibility criteria. The candidate trajectory with the highest plausibility value is considered the ball trajectory.

Another very popular sports is tennis. Here, ball identification in video streams turns out to be very difficult since the ball moves at a high velocity and, at video resolution, only covers a few pixels in the image plane. The LucentVision system [Pingali00] enables real-time tracking of the players' positions and the ball trajectory from video images. Three pairs of video cameras are positioned around the tennis court. The ball's position in each camera view is tracked by performing frame differencing and color-based region identification. The ball's 3D positions are reconstructed via triangulation from stereo.

Primarily in the United States and several countries in Asia, baseball is among the most popular sports. It is a combination of many technically very different and challenging athletic elements. Pitching is the central most important part of the game. Here, the baseball moves very rapidly and it is thus very difficult to obtain detailed measurements of its motion with purely optical means. Alaways

examined in his PhD thesis [Alaways98] the aerodynamics of a curve ball (see Sect. 11.1.4). He used a system with ten high-speed video cameras operating at 240 Hz to capture the ball's flight path. For measuring initial flight parameters, i.e. spin axis and frequency, a marker-based method was employed. During the Summer Olympics 1996 in Atlanta, Alaways used two 120 Hz high-speed video cameras to track ball positions along the flight trajectory [Alaways01].

The K-Zone system [Gueziec03, Gueziec02] is technically similar and designed to track the trajectory of a baseball from multiple video streams in real-time using color information and a Kalman filter.

The measurement accuracy of most of the methods presented in this section is constrained by the limited image resolution of the employed video cameras. In contrast, the algorithm presented in Chap. 12 provides a very high spatial reconstruction accuracy since it capitalizes on the high resolution CCD sensors of modern digital photo cameras.

11.1.3 Hand Motion Tracking

The kinematics of a human hand can be described by means of a hierarchical skeleton structure in a similar way as the full human body (Chap. 2 Sect. 2.1). Hence, many approaches that have originally been designed for human full-body tracking are also applicable to pose estimation of the human hand (see also Chap. 3 Sect. 3.1.3). Nonetheless, many methods have been specifically developed for the latter task.

Nowadays, a variety of non-optical systems are commercially available that require the user to wear a glove with a built-in set of sensors that enables the determination of the bending angles of all finger joints. A glove with built-in touch, bend and inertial sensors is described in [Grimes83]. The DataGlove system [Zimmermann87] uses a set of optical fibers along the back of the fingers that attenuate the light they transmit if they are bent. The Dexterous Hand Master [Eberman93] is a mechanical exoskeleton that can measure the bending angles of finger joints by using Hall-effect sensors.

Optical systems employ one or several video cameras to record the moving hand and to infer parameters of motion. Marker-based systems require optical beacons made of retro-reflective [Vic] tape or LEDs [sel] to be placed on the hand. The 3D trajectories of the markers are tracked and articulated joint motion parameters are estimated. The discrimination between different markers in the image plane is facilitated if each marking is assigned a unique color [Dorner93]. The main advantage of marker-based approaches is that they enable pose determination at a very high accuracy. Another advantage of these optical methods is that the tracked subject does not need to wear cumbersome devices.

However, there are applications in which no attachment to the hand can be

tolerated, not even visual markings. In this case, marker-free optical methods can be employed. Here, one can distinguish between methods that perform pose or gesture analysis purely in 2D, and more sophisticated methods that capitalize on a full 3D hand model.

Not all systems for hand motion estimation perform motion capture in the strict sense, i.e. the estimation of joint motion parameters of a hand skeleton model. For instance, some algorithms perform a purely appearance-based classification to discriminate between different hand gestures. A classification scheme for sign language based on the hand location and the hand shape is described in [Tamura88]. The system uses a dynamic programming method to discriminate signs based on a polygonal approximation of the hand silhouette. A system for real-time sign language recognition based on trajectory analysis of the hand is presented in [Charapayphan92]. A different option is to perform a correlation estimation between recordings of hand gestures and a database of shape templates in order to discriminate between different hand signs [Darrell93]. An appearance-based approach is presented in [Athitsos03] where single hand poses are identified via an edge-feature-based comparison to a database of rendered hand models.

One step further than appearance-based methods take those approaches which perform a full estimation of hand motion parameters in 2D. One such algorithm is presented in [Wu01], where a 2D cardboard hand representation is used for pose determination.

The most detailed representation of human hand motion is obtained with techniques that employ an explicit 3D model. Different 3D model types have been applied for this task. In [Heap96], a point distribution model is used to track hand motion. However, since they resemble the real anatomy of the human hand most closely, kinematic models are applied frequently. They model the human hand as a linked kinematic chain of bones and interconnecting joints. The hand pose is completely determined by the rotation parameters of each joint and the translation of the root. The physical extent of the hand is modeled by some form of surface representation for the palm and the fingers.

In [Stenger01] a kinematic model consisting of quadric segments is employed to determine hand configurations from video. A Kalman filter based tracking framework enables robust pose determination. A kinematic model with a spline-based surface representation is used in [Kuch94] for hand posture recognition. Other types of geometric primitives, such as cylinders or superquadrics, are also commonly used as parts of hand representations for 3D hand motion capture [Davis99, Shimada98]. Physical hand shape models emphasize the deformation of the hand shape under the action of various forces. In [Vogler98] a deformable hand model is used for video-based gesture recognition.

Visual tracking of articulated hand motion is complicated by the fact that self-occlusions of the fingers occur frequently. An explicit 3D hand model can support

the motion capture algorithm in detecting such visibility ambiguities and thereby facilitate robust tracking. Most systems that make use of a 3D hand model employ the *analysis-by-synthesis* principle. Here, the idea is to project the 3D model into the image plane of each recording camera and to iteratively alter its configuration until the projection optimally conforms with the image data. In each iteration an error measure that assesses the matching quality is evaluated. Common error measures compare specific features of the projected model (e.g. silhouettes, edges or texture patches) and their equivalents in the images. For instance, the fingers are salient features that can robustly be identified in video footage. The knowledge of their positions in 3D space in combination with appropriate constraints on the finger motion enables inferring simple hand poses [Lee95].

A highly-detailed kinematic hand model with 27 degrees of freedom and cylinders for representing finger segments is used in the Digiteyes system [Rehg94]. For tracking, the edges of truncated cylinders are projected into the image plane and compared to edges features that are extracted from the images themselves. Finger tips are identified as well. The difference between measured and predicted joint and finger tip locations is minimized by means of a non-linear optimization scheme. A prototype implementation runs at a sustained frame rate of 10 Hz, but unfortunately it cannot properly resolve self-occlusions. The latter problem has been addressed in a follow-up publication [Rehg95].

The method proposed in [Wu99b] estimates the global hand pose prior to determining the pose of the fingers. The hand pose is found by solving a least median squares problem. The parameters of motion for the fingers are inferred via inverse kinematics.

In [Delamarre98] a stereo-based approach for hand tracking is pursued. A stereo correlation algorithm is employed to obtain a dense 3D scene reconstruction. A physically motivated force field is applied to attract the hand model to the reconstructed depth maps.

A data-glove has been used to learn constraints on articulated hand motion in the work presented in [Wu01]. The constraints enable reducing the dimensionality of the 27-dimensional pose space into a union of linear manifolds in 7-dimensional space via PCA. 28 basis configurations turned out to be sufficient such that each hand pose can be represented as a linear combination between two base poses. An importance sampling algorithm is used for tracking.

For more detailed elaboration of the subject we would like to refer the reader to one of the respective survey papers, e.g. [Wu99a].

Figure 11.3: The pitcher (in the background) throws the ball towards the batter (in the foreground) such that it enters the strike zone (red box).

11.1.4 A Primer on Baseball Pitching and the Physics of a Flying Ball

Baseball is amongst the most popular sports in the United States and many countries in Asia. Due to its variety of different athletic elements it is technically very challenging. The pursuit of athletic perfection in baseball already starts in the minor leagues and has lead to the publication of many textbooks on specific technical aspects [Stewart02, House00]. One of the most important technical parts in the game is pitching. In pitching, two actors play the leading part. One actor is the pitcher of one team who is throwing the baseball towards the batter of the opponent team (Fig. 11.3). The batter is supposed to hit the ball with a club and return it as far as possible into the field. The batter stands exactly 18.44 m (60 ft 6 in) away from the pitcher. In order to prevent the batter from hitting the ball, the pitcher is throwing it in such a way that its flight trajectory is as unpredictable as possible. While the flight shall be unpredictable, the rules of the game define a ball only as being valid if it arrives in a virtual box-shaped volume of space close to the batter, the so-called strike zone. The speed at which a world-class baseball player throws the ball can reach up to 100 mph.

In the long history of baseball, a variety of baseball pitches have been developed that differ in the way how the ball flies towards the batter [Stewart02, House00]. The differences in the ball's flight behavior originate from different initial conditions, i.e. different rotation axes, spins and velocities, that the ball was given by the pitcher. Each characteristic combination of initial conditions makes the ball's flight curve deviate in a specific way from its ideal flight parabola, i.e.

(a) (b) (c)

Figure 11.4: Characteristic hand and finger poses for a 2-seam fastball (a), a slider (b), and a curveball (c) [Courtesy of Popular Mechanics magazine].

the parabola that it would fly along if it was thrown in a pure vacuum where aerodynamic effects play no role. Four popular pitching techniques are the fastball, the slider, the curve-ball and the change-up. In our experiments in Chap. 12 we analyze these four different pitches that are all thrown as three-quarter deliveries, i.e. with a release point above and to the right of the head. The characteristics of these pitches are as follows:

- The fastball is the fastest pitch. It has fast back spin and, depending on whether it rotates over four or only two of its seams, the ball is called a 4-seamer or a 2-seamer. It flies above the ideal flight trajectory.
- The change-up also exhibits back spin but has a lower velocity and spin frequency.
- The curveball is released with forward spin which makes the ball descend faster during the last phase of its flight.
- The slider is thrown with as spin that makes the ball turn to one side towards the end of the flight.

The pitcher controls the initial conditions by means of the articulated hand and finger motion at the release point of the ball. Characteristic hand and finger poses for three pitches are shown in Fig. 11.4. The physical origins of the initial-condition-dependent flight trajectories lie in the fact that the ball is traveling through the streaming medium air and not a vacuum. Different initial conditions lead to different strengths of the aerodynamic forces that act on the ball while it is moving [Adair02, Alaways98]. In addition to the aerodynamic forces,

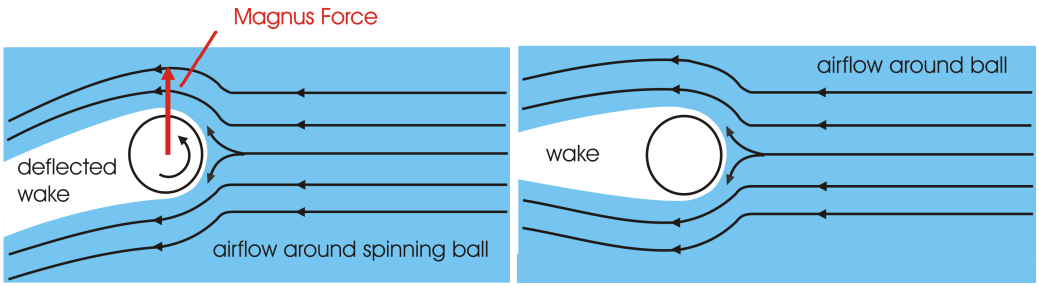


Figure 11.5: Illustration of the effect of the Magnus force on a spinning ball (a); the aerodynamics of a non-spinning ball in comparison (b).

the gravitational force has the strongest influence on the ball's flight behavior. The most important aerodynamic forces are the drag force and the lift forces. The drag force acts in a direction which is opposite to the ball's motion vector. The lift forces are all forces that act in orthogonal directions to the drag force. Several component forces contribute to the overall lift effect, the most dominant one being the Magnus force (Fig. 11.5). It is responsible for the effect that a spinning ball flying through air is laterally deviating from its ideal flight curve. The origins of the force are air pressure differences close to the ball's surface in lateral positions with respect to the heading direction.

Other lift forces are the cross force, whose strength is influenced by the seam orientation of the ball, and the viscous shear force (for details on both forces see [Alaways98]). The latter force is the origin of a phenomenon called spin rate decay which makes the spinning frequency of a flying ball decrease over time. Compared to the Magnus force, the latter two forces only have a minor influence on the flight behavior. In our work presented in Chap. 12, we have developed a physically based mathematical model of the flight of a baseball, that takes into account the gravity, the Magnus force and the drag force acting on the ball.

Chapter 12

Estimating High-Speed Motion with Multi-Exposure Photography

In this chapter, we present a novel and cost-effective principle for capturing high-speed large-scale motion which does not rely on specialized high frame rate video equipment [Theobalt04a]. Instead, it employs standard off-the-shelf digital still cameras, stroboscopic light sources, and the principle of multi-exposure photography. The algorithms and principle design issues detailed here are applicable to a large range of dynamic scenes. We have decided to demonstrate the strength of the novel method and its high accuracy by capturing the articulated hand motion of the pitcher and the flight of the ball during a baseball pitch. We further show that the acquired motion data can be used to generate instructive and, at the same time, entertaining visualizations of the captured events. The framework presented in this chapter introduces the following main scientific contributions:

- an approach for capturing high-speed motion using multi-exposure images obtained with low-cost commodity still cameras and stroboscopes;
- an algorithm to automatically compute the 3D positions and the initial flight parameters of a baseball from multi-exposure images;
- a procedure to reconstruct articulated hand motion from multi-exposure images;
- validation of the approach by means of a physically based model of the flight of a baseball;
- visualization methods for creating renditions of hand and ball motion that can highlight selected technical aspects.

Recording with multi-flash photography has some appealing advantages over recording with high-speed video equipment. To begin with, the acquisition process is a lot cheaper since consumer-grade hardware can be used. Secondly, it is possible to take advantage of the multi-million pixel CCD sensors of digital photo cameras. Finally, the data size of the recorded footage is significantly smaller since the temporal evolution of the dynamic scene is captured in one single image frame.

We proceed with a description of our recording setup in Sect. 12.1. Subsequently, we describe how the motion of the baseball is captured, and flight parameters are estimated in Sect. 12.2.1 through Sect. 12.2.4. We then outline how a physically based model of the ball's flight is used to validate the accuracy of the measured motion parameters, and how it serves as a building block in the generation of renditions, Sect. 12.2.5. The multi-flash recording of the pitcher's articulated hand motion, as well as the algorithmic framework to infer motion parameters from the images, are described in Sect. 12.3.1 through Sect. 12.3.3. The chapter concludes with an evaluation of results and a discussion in Sect. 12.4.

12.1 Setup

We use a flexible setup to robustly acquire different types of motion data under real-world conditions. To analyze flight trajectories of different pitches, we need to acquire image data that allows us to reconstruct the ball's initial flight parameters (i.e. norm and direction of its velocity, direction of its rotation axis, and spin) as well as the 3D positions of the ball along its trajectory. In addition, we want to capture the motion of the pitcher's hand and fingers before, during, and after releasing the ball.

Acquiring this type of information is very challenging since the involved speeds are considerable and the entire trajectory extends over a relatively wide area. To complicate things even further, high spatial accuracy is essential in both flight analysis and hand motion capture.

To obtain simultaneously high spatial as well as temporal resolution, we apply stroboscope photography (Chap. 11 Sect. 11.1.1). We capture images of the high-speed scene in a darkened room using regular digital still cameras that were set to long exposure times. The scene is illuminated with a stroboscope light that emits short light pulses at a suitable frequency. The resulting images depict, superimposed, the dynamic scene at different, closely-spaced time instants. The temporal sampling frequency is equal to the pulse frequency of the stroboscope. High spatial accuracy is easily achieved by using recent commodity digital cameras with multi-million pixel resolution.

To capture an entire baseball pitch, we set up our acquisition gear in a base-

(a) (b)

Figure 12.1: Ball acquisition setup. (a) A stereo pair of cameras (encircled in magenta) facing the black curtain on the right is capturing the ball's initial flight parameters. The ball is illuminated by a stroboscope (cyan). (b) A second stereo pair of cameras (magenta) and a strobe light (cyan) facing towards the black carpet in the back are responsible for capturing the ball's trajectory close to the "home base".

ment room which has a central free space area of approximately 25 m length, 4 m height, and 5 m width. This is sufficient to house the complete pitching corridor (18.44 m in length) as well as to put up the camera and lighting equipment. As imaging devices we employ consumer-market Olympus^{TM} Camedia C5050 still image cameras that provide a frame resolution of 2560x1920 pixels. This camera model features a large-aperture zoom lens that can be set to a comparatively wide angle. We use four cameras of this type in our setup. Recording software was developed enabling us to control the settings of all four cameras from a single PC and to trigger all camera shutters simultaneously.

Since we intend to record a fairly wide-area scene, we need a sufficiently luminous stroboscope light source that can illuminate a large volume at high frequencies. In our setup, we use two high-output strobe flashes which have an intensity of 5000 Lux each at a distance of 0.5 m from the lamp. At full intensity, the 20 μs-long flashes can be triggered at up to 80 Hz which is sufficiently fast for our purpose.

During recording the floor and walls are covered with black carpet and cloth to facilitate foreground object segmentation and automatic marker tracking. Primarily, however, the dark material absorbs most light that has not hit foreground objects, preserving contrast and preventing quick saturation of the multi-exposure images. Finally, a heavy dark carpet hanging down from the ceiling at the end of

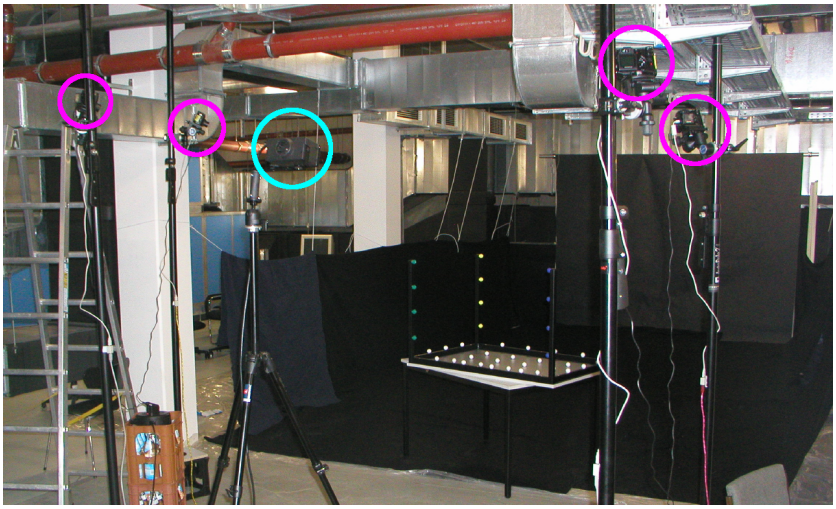


Figure 12.2: Two stereo pairs of cameras (magenta) and a strobe light (cyan) are placed in a semi-circular arrangement around the pitcher to capture the hand motion of different pitches. The object used for extrinsic camera calibration is shown in the center.

the flight corridor absorbs the impact of the ball.

In our recordings, four simultaneously triggered cameras look at the scene from different positions. Two different arrangements of imaging sensors and light sources are needed to record either initial flight parameters and ball positions (see Fig. 12.1) or the hand motion of the pitcher (see Fig. 12.2).

To record the baseball in flight, two stereo pairs of cameras and two stroboscopes are used to capture the initial and final phase of the ball flight, respectively (Fig. 12.1). Details about the setup for acquisition of ball motion are given in Sect. 12.2.2.

For recording the hand motion, the four cameras and one light source are placed in a semi-circular arrangement looking at the pitcher from behind and above, see Fig. 12.2 and Sect. 12.3.2 for further details on this step.

A crucial and—for a large setup like ours—challenging task is the accurate calibration of the cameras. Fortunately, we can take advantage of the algorithmic toolbox for camera calibration that is available in our multi-view video studio (Chap. 4). We apply a camera model for short focal length cameras [Heikkila96]. Intrinsic camera parameters are estimated from images of a planar checkerboard pattern. Radial and tangential lens distortion are modeled up to second order [Jain95]. Each multi-exposure image is distortion-corrected prior to any further processing. Extrinsic camera parameters are estimated using images of our 3D calibration object, see Fig. 12.2. Camera position and orientation are metrically calibrated.

Finally, we rely on our professional baseball pitcher who, as we have verified, performs different pitches with great faithfulness. This allows us to correlate our measurements of hand motion with the measurements of initial flight parameters and flight trajectory.

12.2 Tracking the Ball

The estimation of motion parameters for the flying baseball demands a tracking method that does not have any influence on its flight characteristics. An optical capturing approach that does not require any structural modifications of the ball's surface is thus our method of choice. One possibility would be a tracking approach based on salient features on the ball's outer coating.

However, the outer layer has a very uniform texture and the seams are not significant enough to allow for their robust identification in the photographs. Furthermore, the contrast of a single baseball depiction in a multi-flash photograph is rather low. This is due to the fact that, because of its motion, multiple exposures of the scene background are overlaid with only a single exposure of the ball in the foreground.

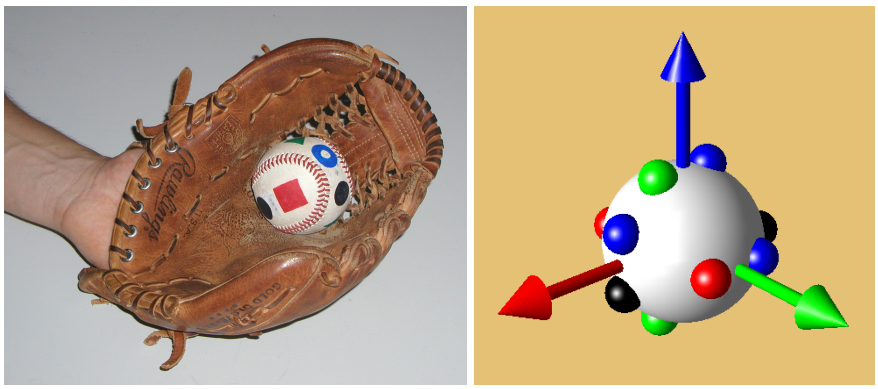
We have thus decided to employ colored optical markings on the ball to facilitate parameter estimation. The markings are painted with colored pens, and therefore the aerodynamic properties of the leather coating are not altered.

12.2.1 Preparation of the Ball

Four different types of markers are used which differ in color and shape (red square, blue ring, green triangle, black circle). Over the entire surface of the ball, each marker type is used three times. Eight markers are arranged in the ball's equatorial plane, in 30°-pairs and with 60° inter-pair separation. The remaining four markers are located in a second, orthogonal plane at 30° distance from the poles. Marker types are assigned such that at least two different markers are visible from any viewpoint. In addition, the (fixed) coordinate system of the ball can be determined from the marker positions for an arbitrary viewing direction. Fig. 12.3 shows the original baseball used in our experiments and the positions of the markers in the coordinate system of the ball.

12.2.2 Recording the Flight of the Ball

In our experiments, we focus on the fastball, the curveball, the slider, and the change-up (see also Chap. 11 Sect. 11.1.4), all of them performed as three-quarter



(a) (b)

Figure 12.3: (a) Baseball equipped with optical markers in pitcher's glove. (b) Illustration of the ball's local coordinate system. Markers are depicted as small colored spheres on the ball.

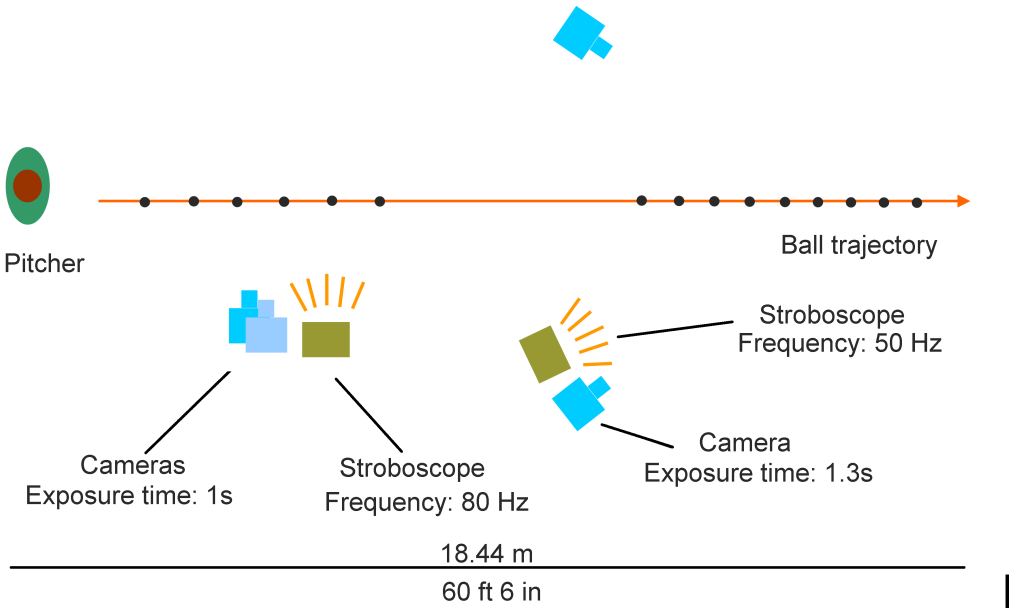


Figure 12.4: Schematic illustration of the flight measurement setup (view from above).

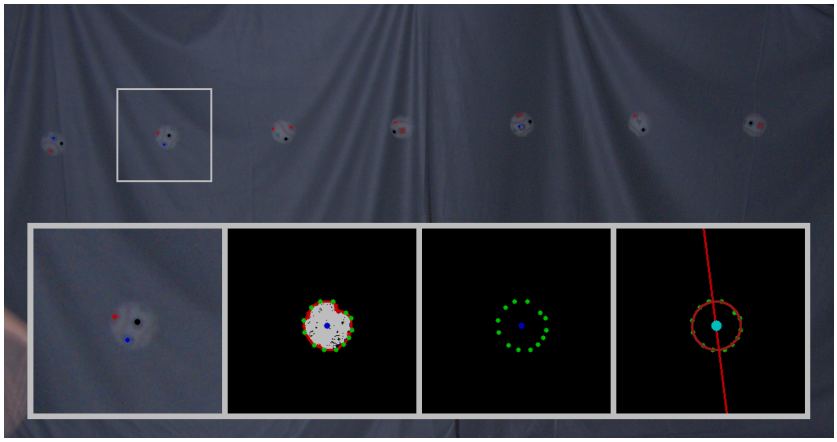


Figure 12.5: Multi-exposure image of the ball used to estimate its initial flight parameters. Automatically detected markers are shown as colored dots. Inset, left to right: magnified image region, result after background subtraction, detected ball silhouette and predicted center point, fitted circle and final center point (see Sect. 12.2.3).

deliveries, i.e. with a release point above and to the right of the head. Each of these pitches was recorded multiple times.

To acquire information about the flight of a baseball, two pairs of cameras are used that focus on different aspects of the ball's trajectory (Fig. 12.4). The front two cameras take multi-exposure pictures of the first 5 m of the flight path right after the ball has left the pitcher's hand. The cameras are placed 3.5 m away from the trajectory and are vertically aligned with a baseline of approximately 0.8 m, see Fig. 12.1a. One strobe light is placed close to the cameras and illuminates the scene such that the ball silhouette appears as a circular shape in the images. In both cameras' multi-exposure image, the ball is seen at several subsequent positions and orientations, flying from left to right in Fig. 12.5. The number of visible ball positions is determined by the pulse frequency of the stroboscope. At a strobe light frequency of 80 Hz, 6–10 ball positions are captured, depending on the speed of the pitch.

The stereo camera pair in the back part of the setup records the last third of the flight trajectory where the most interesting variations between different pitches occur. The cameras are placed approximately 2.8 m high and 4 m apart on either side of the flight corridor, see Fig. 12.1b and Fig. 12.4. A second stroboscope is located below the right camera and illuminates the ball at 50 Hz. This lower frequency is chosen to better spatially separate the ball's depictions in the multi-exposure images. In contrast to the camera setup in the front, the illumination direction in the back setup causes partially illuminated ball silhouettes as shown

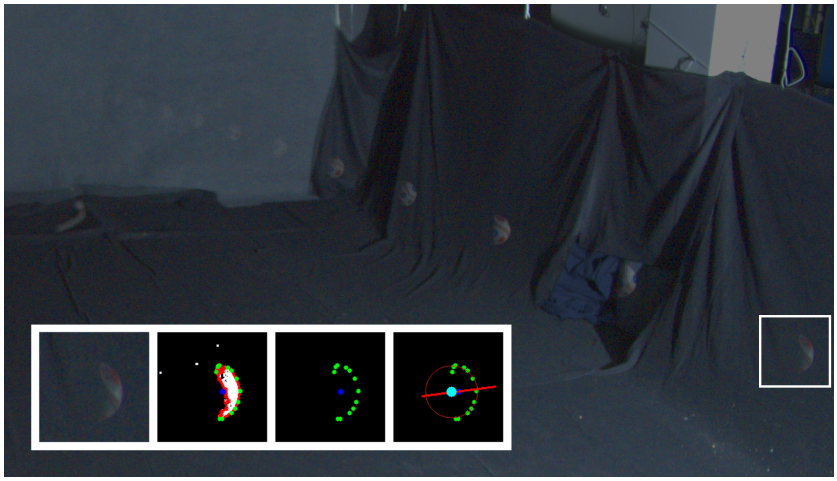


Figure 12.6: Multi-exposure image taken by one of the back cameras. The half-moon shape of the balls is due to the lateral position of the stroboscope illuminating the flight path. The inset shows the same processing steps as those in Fig. 12.5.

in Fig. 12.6. We compensate for this before reconstructing 3D ball positions, see Sect. 12.2.3.

During recording, the shutters of the front cameras are open for 1 s, while the shutters of the back cameras expose for 1.3 s. All cameras are triggered simultaneously. As a trade-off between image noise and brightness, we run each camera with ISO 200 sensitivity.

From the ball centers in both stereo pairs, the 3D positions of the ball in flight are recovered via triangulation, see Fig. 12.9b. In addition, from the marker positions in the front images the orientation of the ball's coordinate frame is computed. This information is used to determine the ball's rotation axis and spin frequency, see Fig. 12.9a. The frequency at which we sample the ball's flight is sufficiently high to enable faithful reconstruction of flight parameters. The Nyquist-Shannon [Nyquist28, Shannon49] sampling theorem tells us that the correct reconstruction of a signal from sample points is only possible if the sampling frequency is at least twice the highest frequency in the signal. According to [Adair02], the highest rotation frequencies commonly observed on a baseball are in the range of 1600 rpm, i.e. around 26.7 Hz. Having our first stroboscope run at 80 Hz, we are on the save side since our sampling frequency is more than twice as high as the rotation frequency of even the fastest spinning ball.

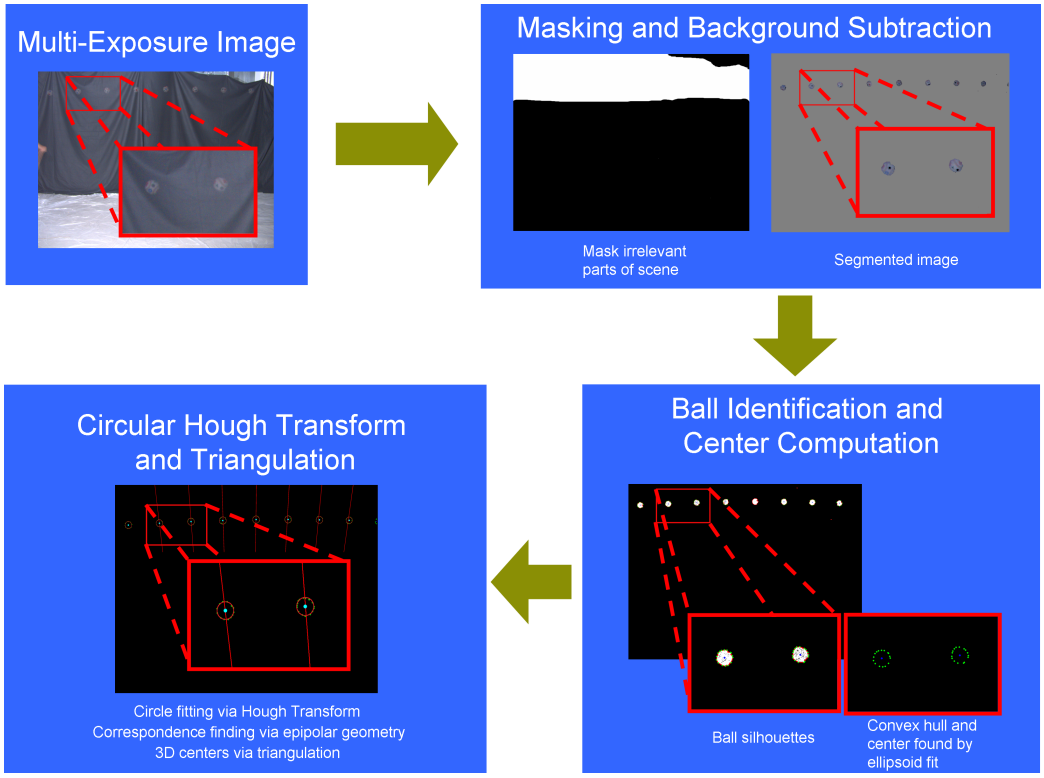


Figure 12.7: Algorithmic steps employed to reconstruct 3D ball positions along the trajectory. The workflow is illustrated using images of one of the cameras positioned in the front part of the setup as examples.

12.2.3 Reconstructing Ball Positions on the Trajectory

In each image that was captured for reconstructing the flight, the ball's outlines are separated from the background by means of a color-based background subtraction method (Chap. 2 Sect. 2.3.1). Since the acquisition setup is static, those parts of the scene which do not change over time can be masked out. Only parts of the images that potentially show the flying ball are subject to the ball silhouette identification procedure.

From the front and back stereo pair of images, the positions of the flying ball in 3D space are computed as follows (see Fig. 12.7 for an overview). The ball's silhouettes form connected components in the binary images that are identified using a contour finding algorithm from the OpenCV library [Intel02]. This algorithm detects the outer contour points of each connected component in the foreground. Smaller noise regions are eliminated by imposing a threshold on the region size. The outer contour of each region is approximated as a polygon using a Douglas-

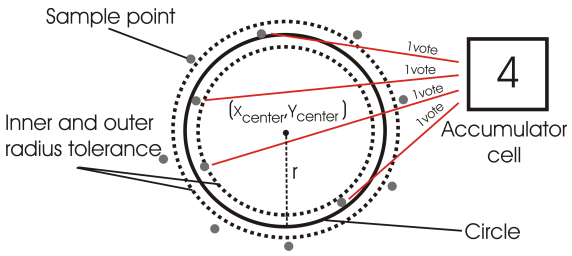


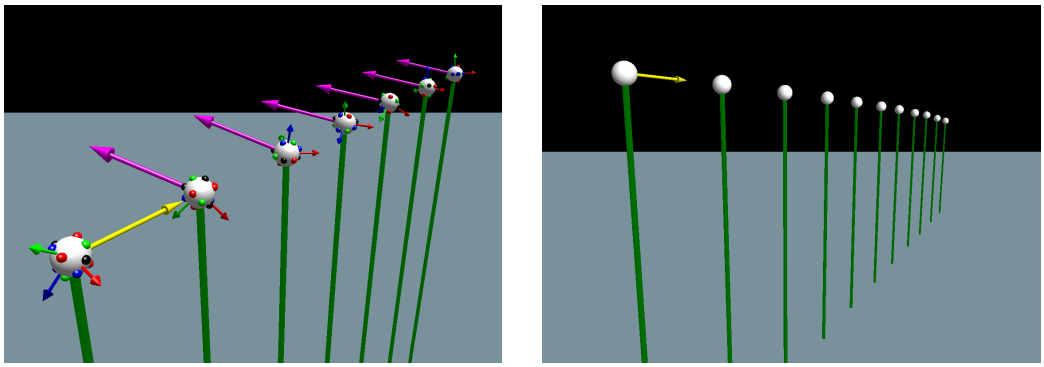
Figure 12.8: Principle of the Circular Hough Transform illustrated. Each point in the image plane that lies on the circle generates one vote for the current combination of circle parameters.

Peucker approximation [Douglas73]. To correct small concavities at the silhouette boundaries that originate from errors in the background subtraction, we compute the convex set of the vertices of each boundary polygon [Slansky70].

A first estimate of the centers of the projected balls is obtained by finding the center of an ellipse that is fitted to the convex hulls of each of the silhouette boundaries in either stereo image. Based on these first center estimates in the image plane, an approximate reconstruction of the ball centers in 3D becomes feasible. To this end, the correspondences between silhouette centers in both camera views need to be established. From epipolar geometry of stereo camera pairs it is known that each image point in one camera view has a corresponding point in the other camera view which lies somewhere along a line in the image, the so-called epipolar line [Faugeras93, Hartley00]. Hence, we establish correspondences by finding for each ellipse center in one image the ellipse center closest to the respective epipolar line in the other image.

The 3D positions of the ball are computed via triangulation. This first estimate is further improved by fitting circles to the convex hull points of the silhouettes in the images using a Circular Hough Transform [Ballard81]. The Circular Hough-Transform (CHT) is a method to fit a circle which is specified via its implicit equation to a set of points in the image plane. A circle is fully defined by three parameters, two for its center in the image plane and one for its radius. The CHT fits a circle by accumulating votes in a three-dimensional array structure. The accumulator array has one cell for each combination of the three implicit circle parameters out of a limited search interval. A vote is added to an accumulator cell if one of the points in the image plane fulfills the implicit circle equation up to a threshold (Fig. 12.8). Parameter combinations are exhaustively sampled, and the one which obtained the highest number of votes determines the optimal fit.

Since the 3D position of the ball is known approximately and the ball's dimensions have been measured beforehand, the radius of the reprojected balls in either view can be predicted. The three-dimensional search space of the Circular



(a) (b)

Figure 12.9: Reconstructed initial flight parameters (a) and flight positions (b). Distances of the balls from the ground are shown in green, the rotation axes are shown in magenta, and the initial velocity direction is depicted in yellow.

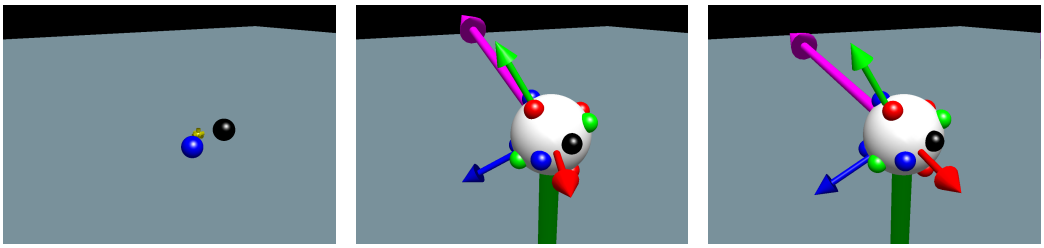
Hough Transform is thereby reduced to two dimensions (image coordinates of the circle's center). The circle centers form our new estimates of the projected ball centers, and the refined ball center positions in 3D are found via triangulation.

The insets of Fig. 12.5 and Fig. 12.6 show the results of the individual fitting steps for the front and back cameras, respectively. One can clearly see that the procedure to determine the ball centers in the image plane also robustly handles the case of partially visible silhouettes.

An estimate of the accuracy of this automatic approach is given by the average Euclidean distance between estimated ball centers in the images and the reprojected reconstructed ball center locations. For the front stereo pair this error is on average less than two pixels if the ball silhouettes have a diameter of 70 pixels. For the back images the reprojection error is less than 3 pixels. Ball positions that are very distant from the light source in the back part of the scene are difficult to detect since the contrast in the images becomes very low. Nonetheless, we still manage to detect up to 11 ball positions in the back part as shown in Fig. 12.9b.

12.2.4 Reconstructing Initial Flight Parameters

After the 3D ball positions in the front and back part of a trajectory have been reconstructed, the initial flight parameters for that data set, i.e. velocity, rotation axis, and spin frequency, are determined (Fig. 12.9a). Fig. 12.10 gives a brief overview of the employed technique: from the reconstructed 3D marker positions,



(a) (b) (c)

Figure 12.10: Stages of the fitting process. (a) to (c): position of markers, result of prediction, and result of final fitting.

an initial guess for the flight parameters is extrapolated, which is then refined using the ball model from Fig. 12.3b.

First, we reconstruct the 3D positions of the markers on the ball from their projections in the two front camera views. For each of the four marker colors in either camera we define an interval in RGB space. For each marker color, candidate pixels in the images after background subtraction are identified via thresholding. In the segmented image, candidate marker positions form connected components. Noisy pixels are eliminated via morphological operations, and (possibly erroneous) small connected components are discarded. We approximate the projected location of a marker's center as the center of mass of the marker region in the image plane.

To triangulate the 3D marker positions the correspondences between the marker projections in both camera views are established via the previously described epipolarity constraint. After their 3D positions have been reconstructed, the isolated marker positions need to be assigned to the correct ball positions. We do this by assigning each marker to the closest ball center position in 3D.

From the sequence of orientations of the ball's coordinate system immediately after release of the ball, its initial spin frequency and rotation axis are derived. In theory, it is sufficient to know the 3D positions of the ball's center and of two uniquely identified markers to determine its orientation. Unfortunately, it is impossible to decide from the color of a marker alone which one of the three instances of this marker type on the ball this is. In addition, we need to take into account that our measurements are subject to noise which may lead to wrongly classified, wrongly located, or missed out markers in the images. Physics tells us that the orientation of the rotation axis and the spin frequency of an ideal flying ball do not change over time. Considering the above, we determine the initial flight parameters by means of the following numerical optimization scheme:

The algorithm processes the n subsequent ball positions at the beginning of the trajectory separately and in their temporal order. The orientation of the ball at position k with respect to the world coordinate system can be represented as a rotation matrix $\mathbf{R}(\alpha_k, \beta_k, \gamma_k)$, where $(\alpha_k, \beta_k, \gamma_k)$ are the ZYZ Euler angles (Chap. 2 Sect. 2.1.1). Our goal is to find for each subsequent pair of 3D ball positions at $k-1$ and k the rotation axis $\vec{\omega}_{k-1,k}$ and rotation angle $\delta_{k-1,k}$ that correspond to the relative rotation transformation $\mathbf{R}_{k-1,k}$ between $\mathbf{R}(\alpha_{k-1}, \beta_{k-1}, \gamma_{k-1})$ and $\mathbf{R}(\alpha_k, \beta_k, \gamma_k)$.

At position k, the algorithm exploits temporal coherence by predicting the orientation of the ball $\mathbf{R}(\alpha_{\text{pred}}, \beta_{\text{pred}}, \gamma_{\text{pred}})$ by rotating orientation $\mathbf{R}(\alpha_{k-1}, \beta_{k-1}, \gamma_{k-1})$ further by $\delta_{k-2,k-1}$ around axis $\vec{\omega}_{k-2,k-1}$. Starting from this parameter set $(\alpha_{\text{pred}}, \beta_{\text{pred}}, \gamma_{\text{pred}})$, the algorithm uses Powell's method [Press02] to find parameters $(\alpha_k, \beta_k, \gamma_k)$ that minimize the energy function:

$$\begin{aligned} E(\alpha_k, \beta_k, \gamma_k) &= a_1 E_1 + a_2 E_2 \\ &= a_1 \sum_{i \in M_k} (\Delta_m(i, \alpha_k, \beta_k, \gamma_k))^2 + \\ &\quad a_2 \sum_{j \in \{x,y,z\}} (\Delta_{ax}(j, \alpha_k, \beta_k, \gamma_k, \alpha_{pred}, \beta_{pred}, \gamma_{pred}))^2 \end{aligned} \tag{12.1}$$

with a_1 and a_2 being weighting factors. M_k is the set of detected markers at ball position k, $\Delta_m(i, \alpha_k, \beta_k, \gamma_k)$ is the angular distance between reconstructed marker i and the closest marker of the same type in the ball model in the current orientation. $\Delta_{ax}(j, \alpha_k, \beta_k, \gamma_k, \alpha_k, \beta_k, \gamma_k, \alpha_{pred}, \beta_{pred}, \gamma_{pred})$ is the angular distance between the local coordinate axis $j \in \{x, y, z\}$ of the ball in orientation $\mathbf{R}(\alpha_k, \beta_k, \gamma_k)$ and the same axis in orientation $\mathbf{R}(\alpha_{\text{pred}}, \beta_{\text{pred}}, \gamma_{\text{pred}})$.

From the optimal orientation $\mathbf{R}(\alpha_k, \beta_k, \gamma_k)$ and the one from the previous time step $\mathbf{R}(\alpha_{k-1}, \beta_{k-1}, \gamma_{k-1})$, the rotation axis $\vec{\omega}_{k-1,k}$ and the rotation angle $\delta_{k-1,k}$ are computed by transferring the relative transform $\mathbf{R}_{k-1,k}$ to axis-angle representation using the technique described in [Murray94].

From the sequence of rotation angles and the stroboscope frequency f_s, the spin frequency f is computed as

$$f = \|\vec{\omega}\| = \frac{f_s}{n} \cdot \sum_{i=1}^{n} \delta_{i-1,i} \tag{12.2}$$

In our method, we do not strictly enforce the constancy of the rotation axis and spin frequency, but instead introduce this criterion as a weighted regularization term E_2. The energy function permits variations in the axis since they might be necessary to compensate for measurement errors. In addition, by this means we can make allowance for the fact that our ball is potentially not an ideal ball. It may exhibit slight material inconsistencies that may lead to a small precession

of the rotation axis. In our experiments, we weight the influence of E_1, which assesses the overlap with the measured marker positions, higher than the one of E_2 ($a_1 = 0.9$, $a_2 = 0.1$). We assume that the initial rotation axis is constant, but that there possibly are measurement errors. Thus, we minimize the impact of erroneous measurements on our results by computing the rotation axis at several ball positions and averaging afterwards.

The direction of the initial velocity vector coincides with the direction of the connecting line between the first two ball positions, its magnitude is computed from the strobe frequency and the Euclidean distance of the first two ball positions.

The above method relies on robustly measured marker positions and rotation axes for the first pair of ball positions. For these positions, we run the optimization with $a_2 = 0$ in Equation (12.1). If this initialization fails due to too few or badly located markers, a manual initialization is feasible.

In our experiments, we were still able to recover valid initial flight parameters even if for some balls none or just one marker was reconstructed due to the fact that some markers are more reliably detected in the images than others. The black markers were detected in almost 100 % of cases, red markers were correctly found in 90 % of cases. The blue and green markers were more difficult to find. In a comparative experiment, it turned out that a different color scheme with more luminous marker colors significantly increases the robustness of marker detection. Marker-detection robustness is further enhanced if the pixel noise is reduced via a dark-frame subtraction.

12.2.5 Validation and Visualization

For the ball flight data (3D positions and initial parameters), no ground truth information is available. To validate our acquisition setup and tracking algorithms, we show that the data obtained through our measurements and processing are consistent with the prediction of a physically based model that takes into account the dominating forces acting on a spinning ball traveling through air (see Sect. 11.1.4). In accordance to [Adair02] and [Alaways01], we compute the velocity $\vec{\mathbf{v}}(t)$ of a baseball with mass m using the first-order ordinary differential equation

$$m\dot{\vec{\mathbf{v}}}(t) \;=\; \vec{\mathbf{F}}_G + \vec{\mathbf{F}}_D(\vec{\mathbf{v}}(t)) + \vec{\mathbf{F}}_M(\vec{\mathbf{v}}(t)) \tag{12.3}$$

pitch type	ε_{avg}	ε_{max}	$\sphericalangle(\vec{\mathbf{v}}_0^{\text{ref}}, \vec{\mathbf{v}}_0)$	$\|\vec{\mathbf{v}}_0^{\text{ref}}\|$	$\Delta(\|\vec{\mathbf{v}}_0^{\text{ref}}\|, \|\vec{\mathbf{v}}_0\|)$	$\sphericalangle(\vec{\omega}^{\text{ref}}, \vec{\omega})$	$\|\vec{\omega}^{\text{ref}}\|$	$\Delta(\|\vec{\omega}^{\text{ref}}\|, \|\vec{\omega}\|)$
fastball (2 seams)	18 mm	39 mm	1.3°	63.2 mph	1.9 mph	0.4°	1596 rpm	22 rpm
fastball (4 seams)	18 mm	41 mm	2.5°	64.2 mph	0.8 mph	0.1°	1612 rpm	17 rpm
curveball	19 mm	39 mm	0.7°	61.9 mph	1.4 mph	0.3°	1623 rpm	7 rpm
slider	15 mm	25 mm	3.8°	65.7 mph	0.7 mph	0.4°	1491 rpm	13 rpm
change-up	13 mm	35 mm	1.4°	60.6 mph	1.1 mph	0.3°	1258 rpm	32 rpm

Table 12.1: Comparison of our measurements with reference trajectories obtained from a physically based model (Sect. 12.2.5). For a variety of pitches, the average error ε_{avg} and the maximum error ε_{max} between the reference trajectory and our measured ball positions are given (Euclidean distance between trajectory and center of ball). The precision of our measured initial flight parameters is specified by: $\sphericalangle(\vec{\mathbf{v}}_0^{\text{ref}}, \vec{\mathbf{v}}_0)$ (angle between reference and measured velocity direction), $\Delta(\|\vec{\mathbf{v}}_0^{\text{ref}}\|, \|\vec{\mathbf{v}}_0\|)$ (difference between reference and measured initial speed), $\sphericalangle(\vec{\omega}^{\text{ref}}, \vec{\omega})$ (angle between reference and measured spin axis direction), and $\Delta(\|\vec{\omega}^{\text{ref}}\|, \|\vec{\omega}\|)$ (difference between reference and measured spin frequency). Absolute values of reference initial speed $\|\vec{\mathbf{v}}_0^{\text{ref}}\|$ and spin frequency $\|\vec{\omega}^{\text{ref}}\|$ are given for the sake of completeness.

with the *gravitational force* $\vec{\mathbf{F}}_G$, the *drag force* (or *air resistance*) $\vec{\mathbf{F}}_D$, and the *Magnus force* $\vec{\mathbf{F}}_M$ defined as:

$$\begin{aligned}
\vec{\mathbf{F}}_G &= m \cdot \vec{\mathbf{g}}\,, \\
\vec{\mathbf{F}}_D(\vec{\mathbf{v}}(t)) &= -\frac{1}{2} \cdot C_D(\vec{\mathbf{v}}(t)) \cdot \rho \cdot A \cdot \|\vec{\mathbf{v}}(t)\|^2 \cdot \frac{\vec{\mathbf{v}}(t)}{\|\vec{\mathbf{v}}(t)\|}\,, \\
\vec{\mathbf{F}}_M(\vec{\mathbf{v}}(t)) &= \frac{1}{2} \cdot C_L(\vec{\mathbf{v}}(t), \vec{\omega}) \cdot \rho \cdot A \cdot \|\vec{\mathbf{v}}(t)\|^2 \cdot \frac{\vec{\omega} \times \vec{\mathbf{v}}(t)}{\|\vec{\omega} \times \vec{\mathbf{v}}(t)\|}
\end{aligned}$$

The vector $\vec{\omega}$ represents the spin axis of the ball, which is assumed to be constant during the flight[1]. In our computations, we use the following constants:

mass of baseball	m	$= 0.145\,\mathrm{kg}$
radius of baseball	r	$= 0.0369\,\mathrm{m}$
cross-sectional area of baseball	A	$= 4.2776 \cdot 10^{-3}\,\mathrm{m}^2$
magnitude of gravity	$\|\vec{\mathbf{g}}\|$	$= 9.80665\,\mathrm{m\,s}^{-2}$
air density	ρ	$= 1.225\,\mathrm{kg\,m}^{-3}$
air viscosity	μ	$= 1.8369 \cdot 10^{-5}\,\mathrm{kg\,m}^{-1}\,\mathrm{s}^{-1}$

Values for air density and air viscosity are given for 20° C at sea level. To compute the *drag coefficient* $C_D(\vec{\mathbf{v}}(t))$, we have fitted a polynomial curve to the data presented in [Adair02] and [Alaways01]. We first compute the Reynolds number

$$Re(\vec{\mathbf{v}}(t)) = 2 \cdot r \cdot \|\vec{\mathbf{v}}(t)\| \cdot \frac{\rho}{\mu}$$

which is then used to evaluate the drag coefficient

$$\begin{aligned}
C_D(\vec{\mathbf{v}}(t)) = {} & 2.23 - \\
& 0.28342 \cdot 10^{-4} \cdot Re(\vec{\mathbf{v}}(t)) + 0.13179 \cdot 10^{-9} \cdot Re(\vec{\mathbf{v}}(t))^2 - \\
& 0.25083 \cdot 10^{-15} \cdot Re(\vec{\mathbf{v}}(t))^3 + 0.17083 \cdot 10^{-21} \cdot Re(\vec{\mathbf{v}}(t))^4
\end{aligned}$$

In [Adair02], the values of the drag coefficient are plotted against the velocity of the ball for a very smooth ball, a new baseball, and a rough baseball. We have decided to fit our polynomial curve to the data given for the rough ball which leads to a slightly lower drag coefficient compared to an ideal baseball. There are two reasons for this: Firstly, after many experiments the outer coating of the ball exhibited already many scratches and fissures. Secondly, [Alaways98] suggests that in our velocity range below 70 mph the real drag coefficient can vary substantially between different pitches and pitchers. In fact, for slower velocities

[1] For a perfectly homogeneous ball, the spin axis does not change. In practice, a small precession might occur due to the inhomogeneous density of natural materials used for baseballs.

the real drag coefficient is tendentially lower than indicated by the curve for the ideal ball given in [Adair02].

According to [Alaways01], the *lift coefficient* C_L can be computed as

$$C_L(\vec{\mathbf{v}}(t), \vec{\omega}) \;=\; 1.5 \cdot r \cdot \frac{\|\vec{\omega}\|}{\|\vec{\mathbf{v}}(t)\|}$$

For the special case of a fastball across two or four seams, better approximations of C_L can be obtained from the diagrams in [Alaways01].

To compute the position $\vec{\mathbf{p}}(t)$ of the flying ball at any time t, we need to know the initial position $\vec{\mathbf{p}}_0 = \vec{\mathbf{p}}(0)$, the initial velocity $\vec{\mathbf{v}}_0 = \vec{\mathbf{v}}(0)$, and the spin axis $\vec{\omega}$ with the spin frequency f encoded in its length: $f = \|\vec{\omega}\|$. With $\vec{\mathbf{v}}(t)$ being the solution of the ODE (12.3), the position $\vec{\mathbf{p}}(t)$ is computed as:

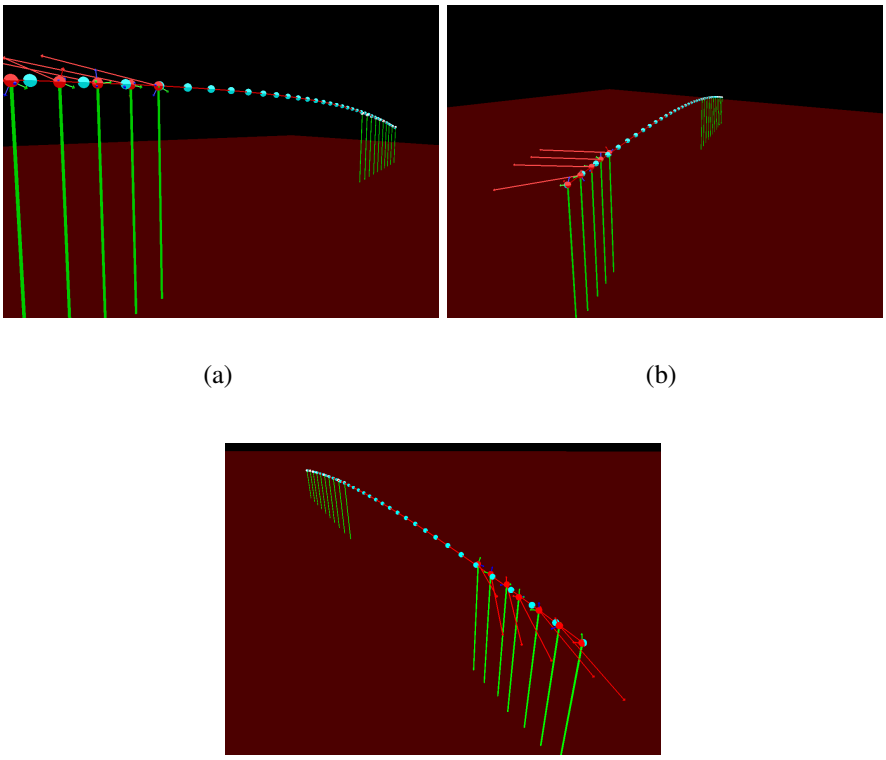
$$\vec{\mathbf{p}}(t) \;=\; \vec{\mathbf{p}}_0 + \int_0^t \vec{\mathbf{v}}(\tau)\,\mathrm{d}\tau \tag{12.4}$$

We employ a numerical integration scheme to solve the ODE (12.3). Very good and stable results have been obtained using the Runge-Kutta-Fehlberg method `DOPRI5` from [Hairer93].

Using Equation (12.4), we can compute the reference trajectory of a baseball for a given set of initial flight parameters $\vec{\mathbf{p}}_0$, $\vec{\mathbf{v}}_0$, and $\vec{\omega}$, and compare it to our measurements. Since the trajectory computed from the ODE (12.3) is quite sensitive w.r.t. variations in the initial flight parameters, we search for an exact solution of (12.3) that minimizes the error both for the measured ball positions and for the measured initial flight parameters using Powell's optimization method [Press02]. The resulting *optimal reference trajectory* is then used to compute the error of our measurements. Table 12.1 lists this error for a variety of different pitch types. In Fig. 12.11 several flight trajectories of the ball that were obtained by fitting the physically based model to the measured data are visualized.

The comparatively low average speed of the pitches is due to the high number of pitches per recording session which exceeded the usual training pensum of a baseball professional by far. All the same we tried to manage with as few sessions as possible so as not to keep the pitcher from his regular training too often.

We employ the physically based flight model not only to assess the accuracy of our approach but also to predict the state of the ball in those sections of the flight trajectory for which we have not taken any measurements. This allows us to create impressive novel visualizations of the ball's inflight behavior that give new insight into the art of pitching. For instance, the complete flight parabola of the ball can be visualized from any arbitrary virtual viewpoint. At the same time, the speed of the ball can be virtually slowed down in order to better visualize its spin. In Fig. 12.17 and Fig. 12.20 examples of such novel visualizations can be seen.



(a) (b)

(c)

Figure 12.11: Visualization of measured and fitted flight curves of the ball for a change-up (a), a fastball 2-seam (b), and a curveball (c). Measured ball positions and measured rotation axes are shown in red. The blue ball positions lie on the fitted flight trajectory according to the physically-based model.

12.3 Tracking the Hand

We estimate the hand poses of the pitcher in the most essential phase of the pitching motion from multi-exposure photographs by means of a marker-based optical tracking approach. Constraints similar to those that already guided the algorithmic design decisions in ball motion capture also apply to the case of hand motion capture with multi-exposure images. The tracking method must work without any modification of the pitcher's hand that could possibly handicap him. Ideally, one would want to resort to a marker-free algorithm. However, the low contrast, the small size of the hand in the image plane, as well as the frequently occurring self-occlusions of the fingers suggest that a marker-based approach is the better

alternative. The kinematics of the hand is modeled by means of a detailed kinematic skeleton model. Motion parameters are estimated from the 3D locations of optical markers on the hand that are reconstructed from the images. The technical and algorithmic aspects of the approach are detailed in the following.

12.3.1 Preparation of the Pitcher's Hand

In order to determine the locations of the finger joints in the recorded images, we have to mark them on the pitcher's hand. The pitcher wears a thin, transparent rubber glove onto which colored markers made of reflective tape are glued, see Fig. 12.16a. The markers are placed on the joint positions, on the finger nails, and on three distinct positions on the back of the hand. Four different marker colors are distributed such that the distance between any two markers of the same color is maximized. In total 18 positions on the hand are tagged and assigned a unique position label. To facilitate identification of the markers in the multi-exposure images, the skin underneath the glove is painted with black make-up. During recordings the pitcher wears black clothes and a black face mask to prevent misclassifications (Fig. 12.13). In preliminary tests we have made sure that the pitcher is not handicapped by the attachments to the hand.

12.3.2 Recording the Hand Motion

For acquisition of hand motion, all four cameras and one stroboscope are positioned in a semi-circular arrangement behind the pitcher, see Fig. 12.12 for a schematic illustration and Fig. 12.2 for a photograph. In front of the pitcher, the walls and the floor of the flight corridor are covered with black cloth. All cameras are focused on the region where the pitcher releases the ball. The camera positions are chosen in such a way that two cameras observe the hand motion from the left and two from the right side of the pitcher's location. This way, occlusions of the hand markers during the complex pitching movement are minimized and sufficiently separated exposures of the hand in the images are obtained. The strobe light is located directly behind and above the pitcher such that the focus of illumination coincides with the release position of the ball. During our recordings, the stroboscope operates at 75 Hz, a frequency that leads to a high number of visible hand positions sufficiently separated in the images for all pitch types. All four cameras are triggered synchronously with an exposure time of one second. We have recorded the same four pitches as for the trajectory measurements. Again, all pitches were performed as three-quarter deliveries.

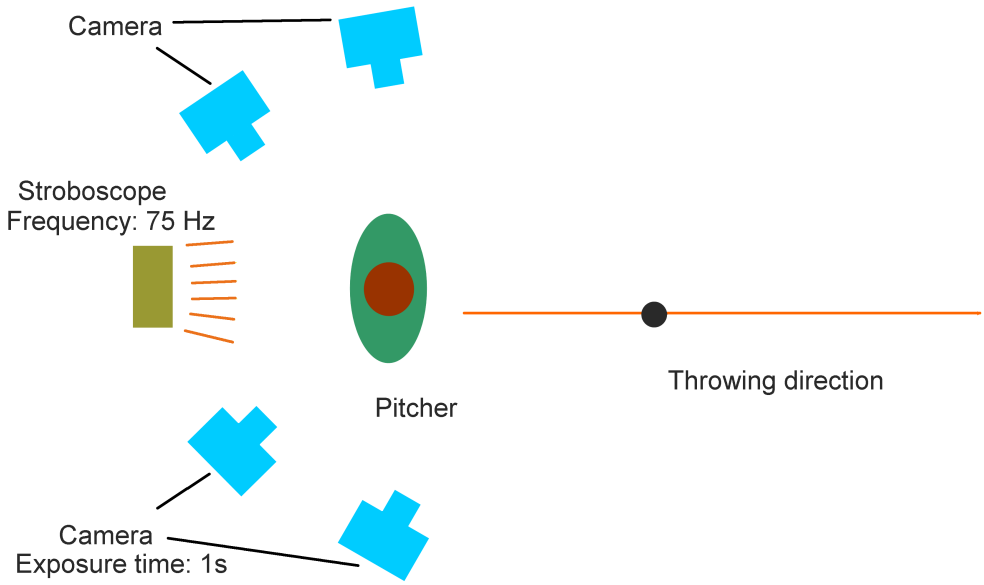
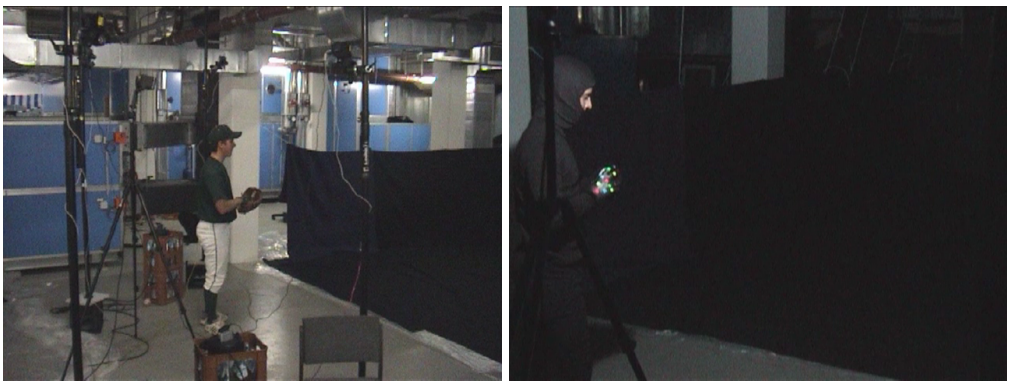


Figure 12.12: Schematic illustration of the hand acquisition setup (view from top).



(a) (b)

Figure 12.13: Photograph of the measurement setup with the pitcher in the center (a). Under measurement conditions the light is dimmed down and the pitcher wears black garment (b).

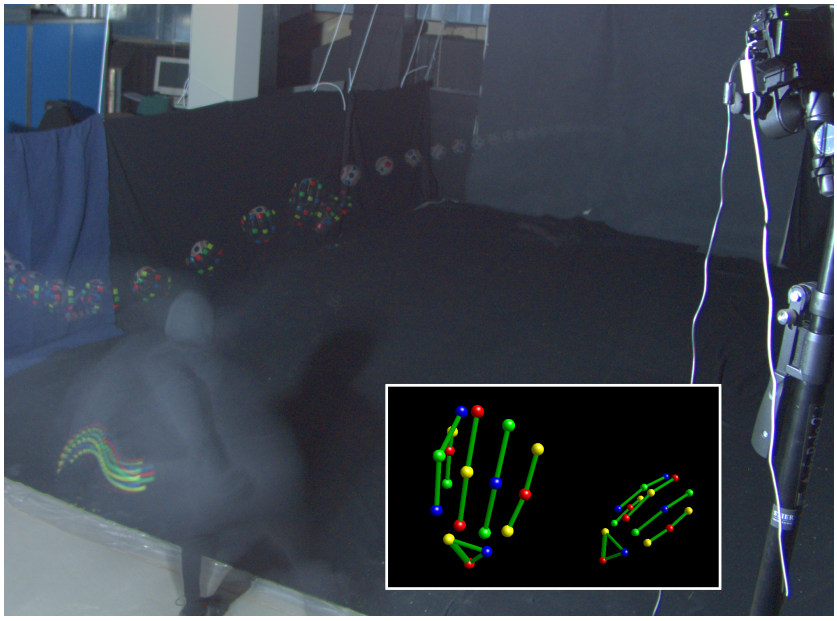


Figure 12.14: Multi-exposure image of one camera recording the hand motion during pitching. Inset: reconstructed hand marker positions for two hand poses.

12.3.3 Reconstructing 3D Positions of Hand Markers

The first step in deriving the motion parameters of the hand is the reconstruction of the 3D positions of the markers that were attached to the back of the hand. The algorithmic workflow of the marker reconstruction procedure is illustrated in Fig. 12.15, its individual steps are detailed in the following:

The input to the procedure are the four calibrated multi-exposure images. In a first interactive step, a rectangular region of interest (ROI) in each image is defined. The ROI contains that part of an image which shows the three to four hand poses close to the ball's release point. The subsequent image processing operations are only applied to the ROI subregions.

After the ROIs have been defined, the projected marker locations in each camera view are determined. The marker identification procedure is identical to the ball marker identification procedure that is described in Sect. 12.2.4. For each marker color (green, yellow, blue and pink), a separate iteration of the marker identification procedure is performed. An interval of allowable pixel colors is defined for each type of marker. Connected image regions above a minimum size whose pixels fall into one of the intervals are considered as projected marker locations. The projected centers of the markers are approximated as the centers of mass of the marker regions. Since all irrelevant parts of the scene are colored

black, the reflective markers emerge very clearly in the images, see Fig. 12.14.

Technically, the four cameras work as two separate stereo pairs. One pair consists of the cameras looking at the pitchers hand from right and above, the second pair consists of the cameras looking at the scene from the left. In order to reconstruct the 3D marker locations, the correspondences between projected marker locations in both cameras have to be established for either stereo pair. This is achieved by exploiting the epipolarity constraint for stereo cameras (Chap. 2 Sect. 2.2.3). Given a marker position in camera 1, its corresponding marker position in camera 2 lies along an epipolar line in camera 2. Since the cameras are fully-calibrated, this line can be directly computed if the image plane location of the marker in camera 1 is known. Given the marker position in camera 1 and the epipolar line in camera 2, all marker depictions in camera 2 are assessed according to their orthogonal distance d to the epipolar line (Fig. 12.15). This way, a list D of marker locations in camera 2 which is sorted according to their distance to the epipolar line is obtained. The marker which is closest to the line is assumed to be the best match. Due to self-occlusions or slight calibration inaccuracies, the correct correspondence may not have been established. A plausibility check can identify inconsistent matches. The plausibility check reconstructs the 3D marker location with the current correspondence. If the reprojection of the estimated 3D marker position does not coincide with the originally determined image plane locations, the correspondences have been established wrongly. In this case, the method proceeds with the next match from D. It iterates until a correct match is found or the distance to the epipolar line is larger than a threshold.

Once the correspondences are established, each stereo pair reconstructs the marker locations in 3D. At this point, our method currently requires the interaction with the user. The user is asked to assign to each reconstructed marker the label of its equivalent on the anatomical hand model. If a marker was reconstructed from both stereo pairs, its 3D location is averaged. The result of this step is a set of labeled isolated marker positions.

Currently, every contingent cluster of markers in 3D space is regarded as one hand position. Automatic identification of hand positions via simple k-means clustering [Mitchell97] is feasible. A fully-automatic approach that, in addition to identifying the clusters, also assigns the correct labels by taking into account the colors of the neighboring markers is also feasible and left to future work.

12.3.4 Motion Parameter Estimation and Hand Visualization

For motion reconstruction, we limit ourselves to those hand positions in which the three markers on the back of the hand are visible in at least two cameras. Only

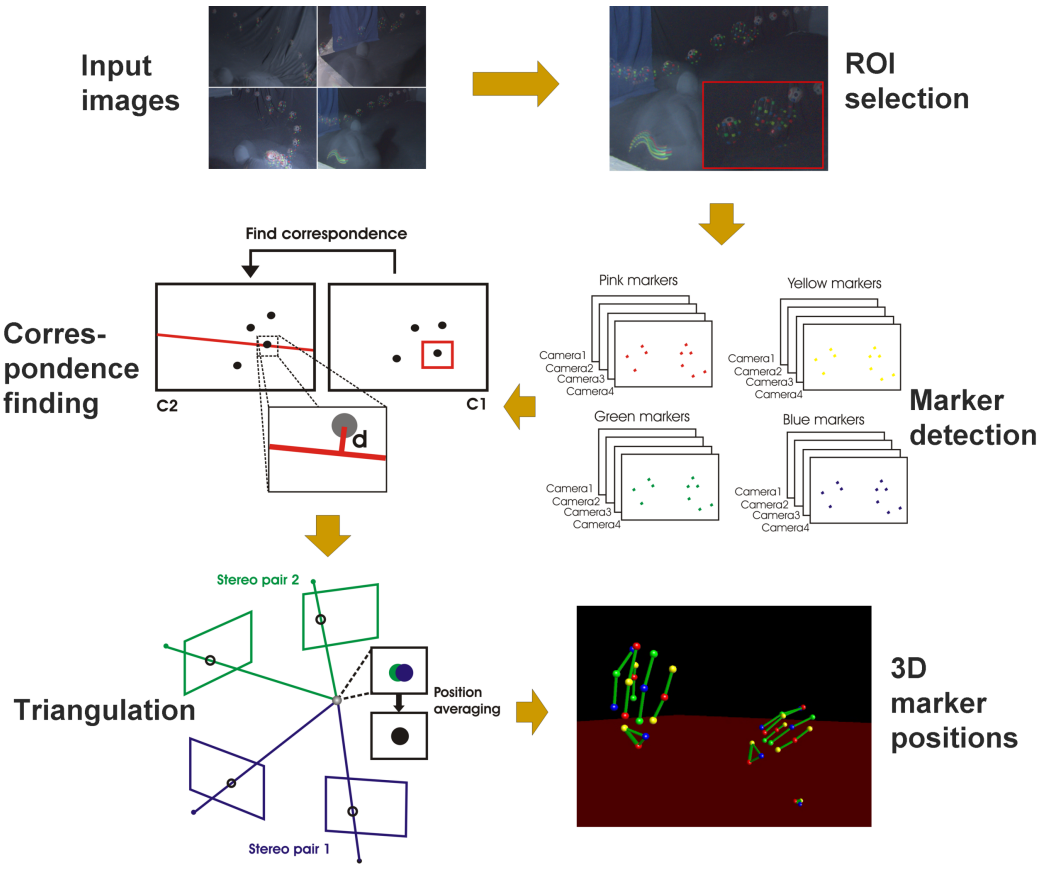


Figure 12.15: Algorithmic workflow connecting the individual steps in 3D hand marker reconstruction.

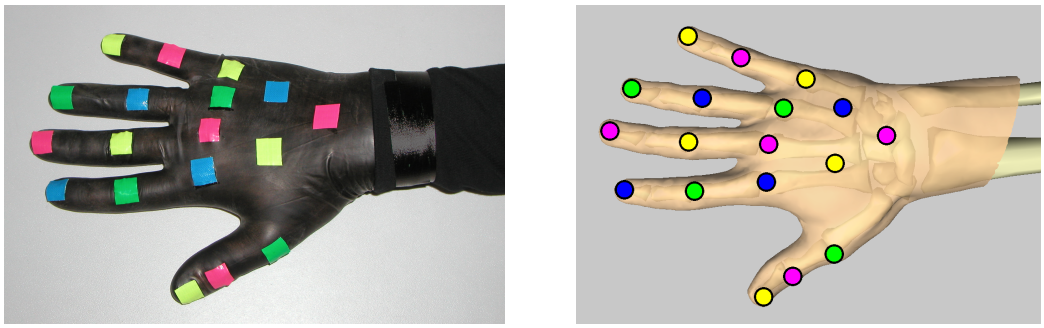
then are the position and orientation of the hand root fully determined. Our setup is arranged such that this condition is fulfilled for an average of four hand positions around the release point. These hand positions are also the most interesting ones in terms of their motion characteristics since they represent that part of the motion cycle in which the hand and finger movements determine the specific rotation axes and spin frequencies of the ball. For some pitches it is not possible to reconstruct the position of all finger joints in each reconstructed hand position. This can happen for those pitch types where a finger is required to be ahead of the ball shortly before it is released such that it is occluded from all cameras.

For representing the movements of hand and fingers, we use an animatable hand model. In particular, our hand model is composed of a skin mesh and the underlying bone structure, see Fig. 12.16b. The hand's kinematics are defined via a hierarchical kinematic chain consisting of the finger bones and the intercon-

necting joints. The root of the chain, located at the wrist joint, provides three translational and three rotational degrees of freedom. Each metacarpophalangeal joint (at the root of the finger) provides two degrees of freedom, and each remaining finger joint features one degree of freedom. Animation of the complete hand model is controlled by specifying temporally varying joint parameters for the skeleton. We employ a physics-based approach to compute the deformation of the skin tissue for a given configuration of the bones inside the hand. The skin mesh is identified with a mass-spring network with biphasic stiffness coefficients computed according to [Van Gelder98]. For the sake of brevity, we refer to the approach presented in [Albrecht03] for a detailed description of this physically based animation technique. We have to make sure that it matches the pitcher's hand in size and proportions. To this end, we apply a radial basis warping function as described in [Albrecht03] to create a "personalized" hand model that matches the size and proportions of the pitcher's hand. The warped model is then equipped with markers at the same positions as on the glove, cf. Fig. 12.16b.

The customized hand model is employed to determine a set of pose parameters for each hand position for which a sufficient number of 3D marker positions has been determined. We achieve this by arranging the hand model in such a way that the 3D positions of the virtual markings on the model optimally comply with the 3D marker positions reconstructed from the image data. First, the global position and orientation of the hand is determined. The three rotational and translational degrees of freedom of the wrist joint are computed by searching for the transformation that optimally aligns the three reconstructed markers on the back of the hand and their equivalent virtual markers on the model. This transformation is estimated in a least squares sense by employing the point set registration scheme that has already been employed in Chap. 9 Sect. 9.3.2 [Horn87]. After the global hand orientation is computed, the finger joint parameters are estimated. We sequentially determine the parameters for each finger joint separately, following the skeleton hierarchy of each finger from the root to the tip. On each hierarchy level, a rotation transformation that minimizes the distance between a marker on the succeeding joint and its reconstructed position in 3D space can be directly inferred. After traversing each finger up to its tip, all joint rotations are specified.

The pose determination method provides us with a set of parameters for three to four key poses close to the ball's release point. Our detailed hand model puts us in the position to photo-realistically render the complete hand motion in-between the key frames. Intermediate time steps are reconstructed via key frame interpolation. The physically based flesh simulation assures plausible skin deformation. Example renditions can be seen in Fig. 12.19 and Fig. 12.18.



(a) (b)

Figure 12.16: (a) Markers for tracking are attached to the pitcher's hand. (b) Corresponding marker positions on the personalized hand model.

12.4 Results and Discussion

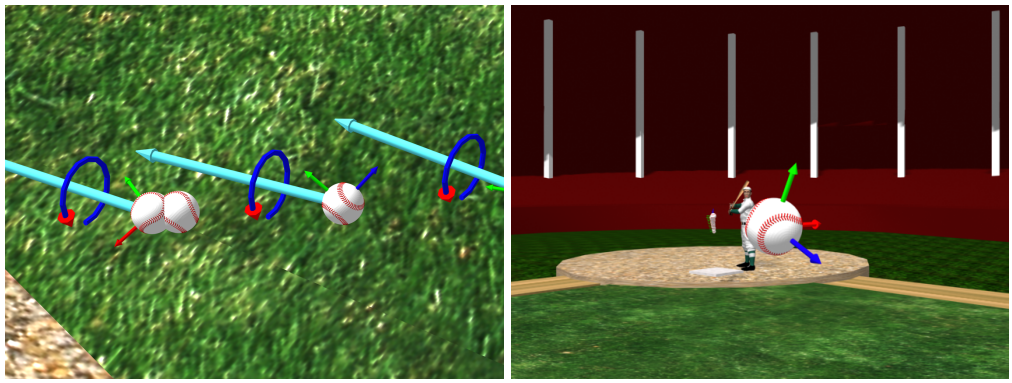
Through experiments we were able to demonstrate that the proposed novel framework for high-speed motion analysis works with very high accuracy, and is equally well suited for ball motion and hand motion capture. On the one hand the captured motion data enable a detailed numerical analysis of different athletic elements of baseball pitching. On the other hand these motion data enable us to take one step further and create renditions of the captured events that are both instructional and entertaining.

For validation of our acquisition setup and tracking algorithms, we have performed the consistency check described in Sect. 12.2.5. As a result, we conclude that our measurements are very accurate. Average errors between the measured 3D ball position and the predicted flight trajectory are as low as 13–19 mm, which corresponds to about 18–25 % of the diameter of the baseball.

The calibration error for the camera setup (evaluated based on the distances between the measured image positions of the markers on the calibration object and their simulated image positions) was on average below one pixel. This assures a high-accuracy 3D reconstruction of the ball and the hand markers.

For the ball, the average distance between a measured feature in the image plane and its reprojected 3D location is below two pixels. The reprojection error for the center of the ball is about one pixel. The discrepancies between measured and predicted marker positions potentially originate from slight inaccuracies during feature localization in the image plane.

Due to the lack of ground truth data for the hand motion we cannot assess the estimated motion parameters directly. However, we can perform a multi-view



(a) (b)

Figure 12.17: (a) Simulated flight of the baseball through the measured initial ball positions. The magenta arrows show the measured rotation axis, the spin direction is also visualized. (b) "Ballycam": Virtual flight behind the baseball.

plausibility check in the same way as it is done for the ball. The reprojection errors of the reconstructed hand markers are equally small as those obtained for the ball measurements.

The high-quality motion data we acquired for different baseball pitches permit new ways of visualization that provide interesting feedback to the athlete, the coach, and the sports enthusiast alike. The flight of the baseball can be visualized from any camera perspective, see Fig. 12.20. In particular, the ball's initial flight parameters and their relation to the flight trajectory can be rendered into instructive sequences. It is even possible to virtually fly behind the ball along its flight parabola and take a closer look at its in-flight behavior, Fig. 12.17.

The visualization of the hand's movements during release of the ball in slow motion provides a new type of visual feedback for the performing pitcher. Fig. 12.19 depicts two snapshots of such an animation. The multi-exposure images acquired for tracking the hand motion show both the hand poses and the ball markers. We have thus reconstructed hand motion and flight parameters from the same set of stroboscope photographs. This way, it is possible to visualize the influence of finger motion on the flight parameters of the ball. In Fig. 12.18, the characteristic finger motion that adds the necessary spin to a slider is clearly visible. In particular, the middle finger exerts high pressure on the ball to build up high spin. Due to the acceleration of the middle finger during the pitch, this finger moves further than the other fingers after release of the ball. The rotation of the ball in Fig. 12.18 is consistent with the movement of the fingers.

Our approach is subject to a couple of limitations. The duration of a motion sequence that can be captured via multi-exposure photography is naturally limited by the recording principle. Furthermore, under some circumstances it might be difficult to establish appropriate environmental conditions prior to recording. However, these limitations apply to high-speed video acquisition in the same way.

In conclusion, we have presented a novel accurate and cost-effective algorithm to capture rapid motion. We don't see it as a replacement of traditional motion capture technology but as a supplement which can be used in cases when traditional techniques fail. The baseball pitch is just an example for the application scenarios we have in mind. Other possible scenarios are tennis serves or the athlete's motion in several track and field events such as javelin or discus. Tennis players, for instance, would benefit from a precise analysis of the correlation between the movement of their racket, speed and spin of the ball, and the resulting ball trajectory during a serve. The acquired high-precision motion data enable the creation of 3D visualizations of athletic movements from arbitrary novel viewpoints. This novel form of 3D visualization can eventually add a new feeling of immersion to the experience of the sports enthusiast at home who is enjoying a TV broadcast. In the future, we plan to extend our framework to enable capturing of human full-body motion.

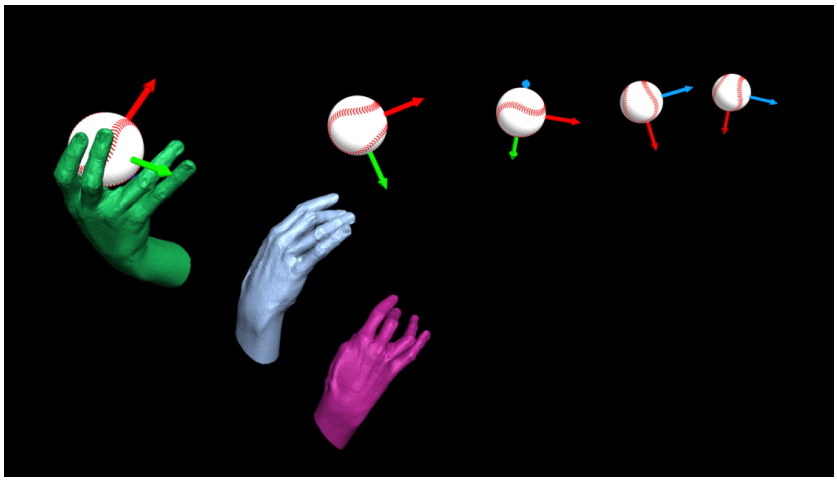


Figure 12.18: Visualization of ball and hand motion obtained from multi-exposure images. The hand motion during release of the baseball is captured and shown together with the resulting flight characteristics of the ball.

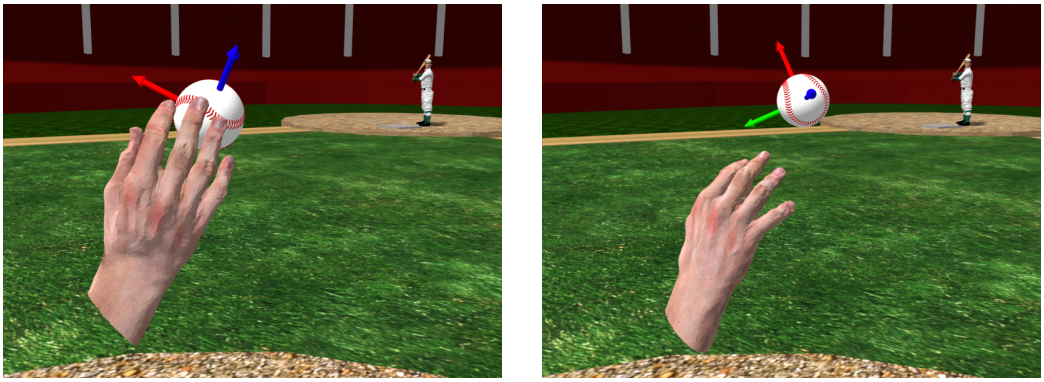


Figure 12.19: Visualization of hand and fingers during and after release of the ball. In this change-up pitch, the ball is spinning backwards around a rotation axis orthogonal to the flight direction. This can be seen by comparing the direction of the axes of the ball's local coordinate frame.

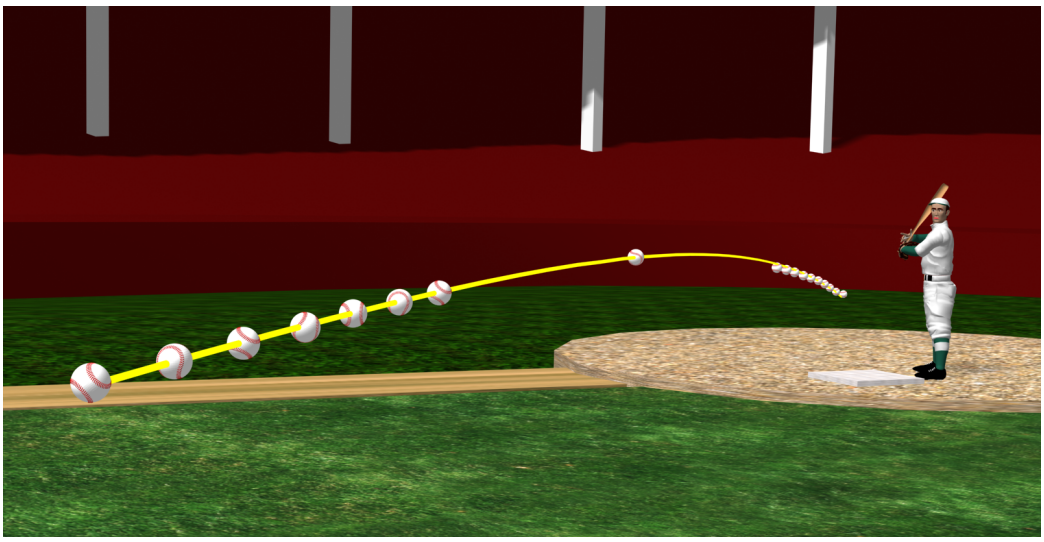


Figure 12.20: Visualization of a change-up trajectory in a stadium. The yellow path shows the reference trajectory obtained from the physical model. The average offset of the measured ball positions to this reference path is as low as 13 mm.

Chapter 13

Conclusions and Outlook to the Future

This thesis has presented novel techniques for optically *capturing*, *analyzing* and *rendering* dynamic real-world scenes involving human actors. In the course of this work, we have presented novel algorithmic solutions to each of these three sub-problems. Furthermore, we have demonstrated that by treating them in conjunction and not in separation, novel forms of immersive visual media, such as free-viewpoint video, can be created. The methods outlined in this work have been originally designed to capture and visualize moving humans. However, many of the fundamental principles are applicable to a much broader range of real-world scenes.

In part I of this thesis, we have first described the technical components of our multi-view video acquisition studio, Chap. 4. It has proven its flexibility and versatility in many of our research projects. The technical descriptions may also serve as a practical guide for other researchers in the field who want to build a similar facility. The remainder of part I detailed novel techniques for solving two fundamental problems in video-based human motion analysis, marker-free motion capture and automatic non-intrusive model estimation. In Chap. 5, we have shown that our hybrid marker-free motion capture approach robustly fits a sophisticated skeleton to human motion data. It effectively combines motion estimation from dynamic voxel models and feature tracking to compensate for weaknesses of each individual technique. Our fully-automatic body model reconstruction approach described in Chap. 6 enables the estimation of kinematic skeleton models of arbitrary moving subjects from multi-view video streams. No a priori information about the object in the scene is required. Our experiments have shown that it is capable of reliably inferring skeleton structures of humans

and animals. In conjunction, the methods from Chap. 5 and Chap. 6 form the building blocks of a complete pipeline for automatic marker-free motion estimation. In future, we therefore plan to integrate both methods into a single framework. We also intend to combine the automatic skeleton estimation procedure with a method to learn a model of the body surface.

In part II, we have described the evolution of a system for reconstructing and rendering free-viewpoint videos of human actors. The core component of the approach is a novel model-based marker-free motion capture method based on image silhouettes, Chap. 8. It enables the reconstruction of the time-varying 3D appearance of a human actor from only a handful of video cameras. Only by means of our newly developed non-intrusive method it becomes possible to use the same video footage for motion estimation and texture reconstruction. The rendered free-viewpoint videos photo-realistically reproduce the appearance of a human in the real world. The motion as well as time-varying details in surface texture are realistically visualized. Even complex human ballet dance can be convincingly rendered from arbitrary virtual camera perspectives.

Since the video footage is not modified, we can employ texture information not only for modeling the surface appearance but also for making motion estimation more robust, Chap. 9. Our enhanced motion capture approach is based on scene flow reconstruction and robustly corrects pose inaccuracies that may exist in the silhouette-fitted poses. Free-viewpoint videos reconstructed with the enhanced approach exhibit an improved model-to-texture registration.

In Chap. 10, a further advantage of our model-based approach comes into play. We have demonstrated that our commitment to an a priori human body model enables us to reconstruct even time-varying surface reflectance properties of the moving actors from multi-view video. This way, 3D videos can be realistically rendered under illumination conditions very different from the time of recording. With our research on free-viewpoint video, we have shown that a marker-free motion estimation method has actually made dynamic geometry reconstruction and reflectance estimation from few camera views feasible. In the future, we want to further capitalize on our compact free-viewpoint video format and investigate ways for the efficient transmission of the data via network channels with strongly limited capacity. Furthermore, we plan to replace the current segmented body model with a single-skin representation which will alleviate some rendering artifacts that are inflicted by the specific geometry representation we employ. Today, free-viewpoint video reconstruction is still a very young research field and many fundamental algorithmic questions have not been answered yet. With our work we have contributed to the advancement of the field by demonstrating that it is feasible to create convincing virtual actors even with a fairly moderate hardware overhead.

In part III of the thesis, we have introduced an alternative cost-effective approach for capturing high-speed motion with high precision. The algorithm is based on the principle of multi-exposure photography with regular still cameras. We have demonstrated the performance of the novel approach by capturing motion parameters of the ball and even pose parameters for the pitcher's hand during a baseball pitch. The visual and numerical validations of the measured data show that our method enables us to capture even large-scale high-speed events at high accuracy. Furthermore, in combination with appropriate shape and motion models for the ball and the hand, instructive and entertaining novel visualizations of the captured events have been rendered. We have thus once more demonstrated that the joint solution to the problems of acquisition, analysis and rendering enables a more detailed understanding and visualization of real-world events. We believe that this approach can be extended to full-body human motion capture.

We are convinced that the algorithmic solution presented in this thesis are a practical proof of our hypothesis that the joint investigation of computer vision and computer graphics problems paves the road for fascinating novel applications. In particular the creation of the next generation of immersive 3D visual media will only be feasible if the problems of scene acquisition, scene reconstruction, and scene rendering are considered in conjunction. The methods presented in this thesis are a collection of algorithmic recipes that put this important insight into practice.

Bibliography

[Adair02] R.K. ADAIR. *The Physics of Baseball*. HarperCollins, New York, NY, 3rd edition, 2002.

[Adelson91] E. H. ADELSON AND J. R. BERGEN. The Plenoptic Function and the Elements of Early Vision. *M. Landy and J. A. Movshon, (eds) Computational Models of Visual Processing*, 1991.

[Aggarwal99] J. K. AGGARWAL AND Q. CAI. Human Motion Analysis: A Review. *CVIU*, 73(3):428–440, March 1999.

[Alaways98] L.W. ALAWAYS. *Aerodynamics of the Curve-Ball: An Investigation of the Effects of Angular Velocity on Baseball Trajectories*. PhD thesis, University of California, Davis, Davis, CA, 1998.

[Alaways01] L.W. ALAWAYS, S.P. MISH, AND M. HUBBARD. Identification of Release Conditions and Aerodynamic Forces in Pitched-Baseball Trajectories. *Journal of Applied Biomechanics*, 17:63–76, 2001.

[Albrecht03] I. ALBRECHT, J. HABER, AND H.-P. SEIDEL. Construction and Animation of Anatomically Based Human Hand Models. In *Proc. Symposium on Computer Animation (SCA '03)*, pages 98–108, 2003.

[Aliaga01] D.G. ALIAGA AND I. CARLBOM. Plenoptic stitching: a scalable method for reconstructing 3D interactive walk throughs. In *Proc. of SIGGRAPH'01*, pages 443–450. ACM Press, 2001.

[Allen02] B. ALLEN, B. CURLESS, AND Z. POPOVIC. Articulated Body Deformations from Range Scan Data. In *Proceedings of ACM SIGGRAPH'02*, pages 612–619, 2002.

[Athitsos03] V. ATHITSOS AND S. SCLAROFF. Estimating 3D Hand Pose from a Cluttered Image. In *Proc. of CVPR'03*, volume 2, page 432, 2003.

[Avidan97] S. AVIDAN AND A. SHASHUA. Novel view synthesis in tensor space. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 1034. IEEE Computer Society, 1997.

[Ballard81] D.H. BALLARD. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[Banégas01] F. BANÉGAS, M. JAEGER, D. MICHELUCCI, AND M. ROELENS. The ellipsoidal skeleton in medical applications. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 30–38. ACM Press, 2001.

[Bardinet98] E. BARDINET, L.D. COHEN, AND N. AYACHE. A parametric deformable model to fit unstructured 3D data. *Computer Vision and Image Understanding*, 71(1):39–54, 1998.

[Barron94] J.L. BARRON, D.J. FLEET, AND S.S. BEAUCHEMIN. Performance of Optical Flow Techniques. *IJCV*, 12:1:43–77, 1994.

[Baumberg94] A. BAUMBERG AND D. HOGG. An Efficient Method for Contour Tracking using Active Shape Models. In *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 194–199, 1994.

[Bernardini01] F. BERNARDINI, I.M. MARTIN, AND H. RUSHMEIER. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics*, 7(4):318–332, October - November 2001.

[Bhaskaran99] V. BHASKARAN AND K. KONSTANTINIDIS. *Image and Video Compression Standards*. Kluwer, 1999.

[Boivin01] S. BOIVIN AND A. GAGALOWICZ. Image-Based Rendering of Diffuse, Specular and Glossy Surfaces From a Single Image. In *Proc. of SIGGRAPH'01*, pages 107–116. ACM Press, 2001.

[Borovikov00] E. BOROVIKOV AND L. DAVIS. A Dristibuted System for Real-Time Volume Reconstruction. In *Proceedings of Intl. Workshop on Computer Architectures for Machine Perception*, page 183ff, 2000.

[Bottino01] A. BOTTINO AND A. LAURENTINI. A Silhouette-Based Technique for the Reconstruction of Human Movement. *CVIU*, 83:79–95, 2001.

[Bradski98] G. BRADSKI. Computer Vision Face Tracking as a Component of a Perceptual User Interface. In *IEEE Workshop of Applications of Computer Vision*, pages 214–218, 1998.

[Bregler98] C. BREGLER AND J. MALIK. Tracking People with Twists and Exponential Maps. In *Computer Society Conference on Computer Vision and Pattern Recognition 98*, pages 8–15, 1998.

[Bronstein91] I.N. BRONSTEIN AND K.A. SEMENDJAJEW. *Taschenbuch der Mathematik*. Teubner, 1991.

[Bruce94] R.R. BRUCE. *Seeing the Unseen: Dr. Harold E. Edgerton and the Wonders of Strobe Alley*. MIT Press, 1994.

[Byrd95] R. BYRD, P. LU, J. NOCEDAL, AND C. ZHU. A limited memory algorithm for bound constrained optimi- zation. *SIAM J. Sci. Comp.*, 16(5):1190–1208, 1995.

[Cai96] Q. CAI AND J. AGGARWAL. Tracking Human Motion using Multiple Cameras. In *Proc. of Intl. Conference on Pattern Recognition*, pages 68–72, 1996.

[Calvert94] T. CALVERT AND M. CHAPMAN. *Analysis and Synthesis of Human Movement*, pages 432–474. Academic Press, 1994.

[Capin99] T.K. CAPIN AND D. THALMANN. Controlling and Efficient coding of MPEG-4 Compliant Avatars. In *Proc. IWSNHC3DI'99*, Santorini, Greece, 1999.

[Carceroni01] R.L. CARCERONI AND K.N. KUTULAKOS. Multi-View Scene Capture by Surfel Sampling: From Video Streams to Non-Rigid 3D Motion Shape & Reflectance. In *ICCV*, pages 60–67, 2001.

[Carlsson00] S. CARLSSON. Recognizing Walking People. In *Proc. of European Conference on Computer Vision*, volume 1 of *LNCS 1842*, pages 472–486. Springer, 2000.

[Carranza03] J. CARRANZA, C. THEOBALT, M.A. MAGNOR, AND H.-P. SEIDEL. Free-Viewpoint Video of Human Actors. *ACM Transactions on Graphics (Proc. of SIGGRAPH'03)*, 22(3):569–577, July 2003.

[Chai00] JIN-XIANG CHAI, SHING-CHOW CHAN, HEUNG-YEUNG SHUM, AND XIN TONG. Plenoptic sampling. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 307–318. ACM Press/Addison-Wesley Publishing Co., 2000.

[Chang96] I.-C. CHANG AND C.-L. HUANG. Ribbon-based motion Analysis of Human Body Movements. In *Proc. of Intl. Conference on Pattern Recognition*, pages 436–440, 1996.

[Charapayphan92] C. CHARAPAYPHAN AND A.E. MARBLE. Image processing system for interpreting motion in ASL. *Journal of Biomedical Engineering*, 14(5), 1992.

[Chen92] Z. CHEN AND H.J. LEE. Knowledge-guided visual perception of 3D human gait from a single camera sequence. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):336–342, 1992.

[Chen93] S.E. CHEN AND L. WILLIAMS. View interpolation for image synthesis. In *Proc. of SIGGRAPH'93*, pages 279–288. ACM Press, 1993.

[Cheung00] K.M. CHEUNG, T. KANADE, J.-Y. BOUGUET, AND M. HOLLER. A real time system for robust 3D voxel reconstruction of human motions. In *Proc. of CVPR*, volume 2, pages 714 – 720, June 2000.

[Cheung03] G. CHEUNG, BAKER. S., AND T. KANADA. Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estiamtion and Motion Capture. In *Proc. of CVPR*, 2003.

[Chevalier03] L. CHEVALIER, F. JAILLET, AND BASKURT. A. Segmentation and Superquadric Modeling of 3D Objects. In *Proceedings of WSCG 2003*, 2003.

[Cootes95] T. F. COOTES, C. J. TAYLOR, D. H. COOPER, AND J. GRAHAM. Active shape models - their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59, 1995.

[Cos] Cosyco. http://www.cosyco.de.

[Covelle00] M.M. COVELLE, A. RAHIMI, M. HARVILLE, AND T.J. DARRELL. Articulated Pose Estimation using Brighness and Depth Constancy Constraints. In *Proc. of IEEE Intl. Conf on Computer Vision and Pattern Recognition*, volume 2, pages 438–445, 2000.

[Darrell93] T. DARRELL AND A. PENTLAND. Space-time Gestures. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 335–340, 1993.

[Davis93] J. DAVIS AND M. SHAH. Gesture recognition. Technical Report CS-TR-93-11, University of Central Florida, 1993.

[Davis99] J. DAVIS AND M. SHAH. Towards 3-D gesture recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(3):381–393, 1999.

[de Aguiar04] E. DE AGUIAR, C. THEOBALT, M. MAGNOR, H. THEISEL, AND H.-P. SEIDEL. M3 : Marker-free Model Reconstruction and Motion Tracking from 3D Voxel Data. In *12th Pacific Conference on Computer Graphics and Applications, PG 2004*, pages 101–110, Seoul, Korea, October 2004. IEEE.

[Debevec00] P. DEBEVEC, T. HAWKINS, C. TCHOU, H.-P. DUIKER, W. SAROKIN, AND M. SAGAR. Acquiring the Reflectance Field of a Human Face. *Proc. ACM Conference on Computer Graphics (SIGGRAPH'00),* New Orleans, USA, pages 145–156, August 2000.

[Delamarre98] Q. DELAMARRE AND O.D. FAUGERAS. Finding pose of hand in video images: a stereo based approach. In *Proc. 3rd Intl. Conf. on Automatic Face and Gesture Recognition*, pages 585–590, 1998.

[Delamarre99] Q. DELAMARRE AND O. FAUGERAS. 3D Articulated Models and Multi-View Tracking with Silhouettes. In *Proc. of ICCV 99*, pages 716–721, 1999.

[Deutscher00] B. DEUTSCHER, A. BLAKE, AND I. REID. Articulated Body Motion Capture by Annealed Particle Filtering. In *Proc. of CVPR'00*, volume 2, page 2126ff, 2000.

[D'Orazio02] T. D'ORAZIO, N. ANCONA, G. CICIRELLI, AND M. NITTI. A Ball Detection Algorithm for Real Soccer Image Sequences. In *Proc. of ICPR'02*, volume 1, page 10210, 2002.

[Dorner93] B. DORNER. Hand shape identification and tracking for sign language interpretation. In *IJCAI Workshop on Looking at People*, 1993.

[Douglas73] D.H. DOUGLAS AND T.K. PEUCKER. Algorithms for the reduction of the number of points required to represent a line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

[Drummond01] T. DRUMMOND AND R. CIPOLLA. Real-time Tracking of Highly Articulated Structures in the Presence of Noisy Measurements. In *Proc. of ICCV*, volume 2, pages 315–320, 2001.

[Eberly02] D. EBERLY. Rotation Representations and Performance Issues. Technical report, Magic Software, Inc., 2002.

[Eberman93] EBERMAN. *Product information. Exos Dexterous Handmaster.*, 1993.

[Eisert01] P. EISERT. *Very low bit-rate video coding using 3-D models*, volume 20 ; D29 of *Berichte aus der Kommunikations- und Informationstechnik*. Shaker-Verlag, 2001.

[Farin99] G. FARIN. *Curves and Surfaces for CAGD: A Practical Guide*. Morgan Kaufmann, 1999.

[Faugeras93] O. FAUGERAS. *Three-dimensional computer vision : a geometric viewpoint*. MIT Press, 1993.

[Fehn04] C. FEHN. Depth-Image-Based Rendering (DIBR), Compression and Transmission for a New Approach on 3D-TV. In

Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems XI, pages 93–104, San Jose, CA, USA, 2004.

[Fernando04] R. FERNANDO. *GPU Gems: Programming Techniques, Tips and Tricks for Real-time Graphics*. Nvidia, 2004.

[Freeman96] W. FREEMAN, K. TANAKA, J. OHTA, AND K. KYUMA. Computer Vision for Computer Games. In *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 100–105, 1996.

[Fukunaga90] K. FUKUNAGA. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 1990.

[Gardner03] A. GARDNER, C. TCHOU, T. HAWKINS, AND P. DEBEVEC. Linear light source reflectometry. *ACM Trans. Graphics.*, 22(3):749–758, 2003.

[Gavrila96] D.M. GAVRILA AND L.S. DAVIS. 3D Model-Based Tracking of Humans in Action: A Multi-View Approach. In *Proc. of CVPR*, pages 73–80, 1996.

[Gavrila99] D.M. GAVRILA. The Visual Analysis of Human Movement. *Computer Vision and Image Understanding*, 73(1):82–98, January 1999.

[Georghiades03] A.S. GEORGHIADES. Recovering 3-D Shape and Reflectance From a Small Number of Photographs. In *Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering*, pages 230–240, 2003.

[Geurtz93] A. GEURTZ. *Model-based Shape Estimation*. PhD thesis, Polytechnic Institute of Lausanne, 1993.

[Gibson01] SIMON GIBSON, TOBY HOWARD, AND ROGER HUBBOLD. Flexible Image-Based Photometric Reconstruction using Virtual Light Sources. *Computer Graphics Forum*, 20(3), 2001. ISSN 1067-7055.

[Gleicher99] M. GLEICHER. Animation from Observation: Motion Capture and Motion Editing. *Computer Graphics*, 4(33):51–55, November 1999.

[Gleicher02] M. GLEICHER AND N. FERRIER. Evaluating Video-Based Motion Capture. In *CA '02: Proceedings of the Computer Animation*, page 75. IEEE Computer Society, 2002.

[Goesele00] M. GOESELE, H. LENSCH, W. HEIDRICH, AND H.P. SEIDEL. Building a Photo Studio for Measurement Purposes. In *Proceedings of VMV2000*, pages 231–238, 2000.

[Goesele04] M. GOESELE. *New Acquisition Techniques for Real Objects and Light Sources in Computer Graphics*. PhD thesis, Universität des Saarlandes, July 2004.

[Goldman04] D.B. GOLDMAN, B. CURLESS, A. HERTZMANN, AND S. SEITZ. Shape and Spatially-Varying BRDFs From Photometric Stereo. Technical Report UW-CSE-04-05-03, University of Washington, 2004.

[Goncalves95] L. GONCALVES, E. DIBERNARDO, E. URSELLA, AND P. PERONA. Monocular Tracking of the Human Arm in 3D. In *Proc. of CVPR*, pages 764–770, 1995.

[Gortler96] S.J. GORTLER, R. GRZESZCZUK, R. SZELISKI, AND M.F. COHEN. The Lumigraph. In *Proc. of ACM SIGGRAPH'96*, volume 30, pages 43–54, 1996.

[Grammalidis01] N. GRAMMALIDIS, G. GOUSSIS, G. TROUFAKOS, AND M.G. STRINTZIS. Estimating Body Animation Parameters From Depth Images Using Analysis By Synthesis. In *Proc. of Second International Workshop on Digital and Computational Video (DCV'01)*, page 93ff, 2001.

[Gre] Gretag MacBeth. www.gretagmacbeth.com.

[Grimes83] G.J. GRIMES. Digital Data Entry Glove Interface Device. Technical Report US Patent 4,414,537, Bell Telephone Laboratories, 1983.

[Gross03] M.H. GROSS, S. WÜRMLIN, M. NÄF, E. LAMBORAY, C.P. SPAGNO, A.M. KUNZ, E. KOLLER-MEIER, T. SVOBODA, L.J. VAN GOOL, S. LANG, K. STREHLKE, A. VANDE MOERE, AND O.G. STAADT. blue-c: a spatially immersive display and 3D video portal for telepresence. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, 22(3):819–827, 2003.

[Gueziec02] A. GUEZIEC. Tracking Pitches for Broadcast Television. *IEEE Computer*, 35(3):38–43, 2002.

[Gueziec03] A. GUEZIEC. Tracking a Baseball for Broadcast Television. In *SIGGRAPH Course Notes*, 2003.

[Guo94] Y. GUO, G. XU, AND S. TSUJI. Tracking Human Body Motion Based on a Stick-Figure Model. *Journal of Visual Communication and Image Representation*, 5(1):1–9, 1994.

[Hairer93] E. HAIRER, S.P. NØRSETT, AND G. WANNER. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer–Verlag, New York, 2nd edition, 1993.

[Haritaoglu98] I. HARITAOGLU, D. HARWOOD, AND L.S. DAVIS. W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People. In *Conference on Automatic Face and Gesture Recognition 98 (Tracking and Segmentation of Moving Figures)*, pages 222–227, 1998.

[Hartley00] R. HARTLEY AND A. ZISSERMAN. *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge, 2000.

[Hawkins04] T. HAWKINS, A. WENGER, C. TCHOU, A. GARDNER, F. GÖRANSSON, AND P. DEBEVEC. Animatable Facial Reflectance Fields. In *Proc. of Eurographics Symposium on Rendering*, pages 309–319, 2004.

[Heap96] T. HEAP AND D. HOGG. Towards 3D Hand Tracking using a Deformable Model. In *Proc. of 2nd Intl. Conf. on Automatic Face and Gesture Recognition*, page 140, 1996.

[Heikkila96] J. HEIKKILA AND O. SILVEN. Calibration procedure for short focal length off-the-shelf CCD cameras. In *Proc. of 13th International Conference on Pattern Recognition.*, pages 166–170, 1996.

[Herda00] L. HERDA, P. FUA, R. PLAENKERS, R. BOULIC, AND D. THALMANN. Skeleton-Based Motion Capture for Robust Reconstruction of Human Motion. In *Proceedings of Computer Animation 2000*. IEEE CS Press, 2000.

[Herda04] L. HERDA, R. URTASUN, AND P. FUA. Hierarchical Implicit Surface Joint Limits to Constrain Video-based Human Motion Capture. In *Proc. of ECCV*, volume 2, pages 405–418, 2004.

[hiw] www.hiviz.com.

[Horn86] B.K.P. HORN. *Robot Vision*. MIT Press, Cambridge, 1986.

[Horn87] B.K.P. HORN. Closed-form Solution of Absolute Orientation using unit Quaternions. *Journal of the Optical Sociey of America*, 4(4):629–642, 1987.

[Horprasert98] T. HORPRASERT, I. HARITAOGLU, D. HARWOOD, L. DAVIS, C. WREN, AND A. PENTLAND. Real-time 3D Motion Capture. In *Second Workshop on Perceptual Interfaces*, 1998.

[House00] T. HOUSE. *The Pitching Edge*. Human Kinetics, Champaign, IL, 2nd edition, 2000.

[Hunter95] E. HUNTER, J. SCHLENZIG, AND R. JAIN. Posture Estimation in Reduced-model gesture input systems. In *proc. of Intl. Workshop on Automatic Face and Gesture Recognition*, pages 290–295, 1995.

[Intel02] INTEL. Open Source Computer Vision Library. http://www.sourceforge.net/projects/opencvlibrary, 2002.

[Isaksen00] A. ISAKSEN, L. MCMILLAN, AND S.J. GORTLER. Dynamically reparameterized light fields. In *Proc. of SIGGRAPH'00*, pages 297–306. ACM Press, 2000.

[ISO/IEC00] ISO/IEC. Overview of the MPEG-4 Standard. `http://www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm`, July 2000.

[Jain95] R. JAIN, R. KASTURI, AND B.G. SCHUNCK. *Machine Vision*. McGraw Hill International, 1995.

[Janesick01] J.R. JANESICK. *Scientific Charge Coupled Devices*. SPIE, 2001.

[Johannson73] G. JOHANNSON. Visual Perception of Biological Motion and a Model for its Analysis. *Perception and Psychophysics*, 14(2):201–211, 1973.

[Jolliffe86] I. T. JOLLIFFE. *Principal Component Analysis*. Springer, 1986.

[K56] K5600. http://www.k5600.com.

[Kähler03] K. KÄHLER. *A Head Model with Anatomical Structure for Facial Modeling and Animation*. PhD thesis, Universität des Saarlandes, May 2003.

[Kakadiaris95] I.A. KAKADIARIS AND D. METAXAS. 3D Human Body Model Acquisition from Multiple Views. In *Proc. of ICCV'95*, pages 618–623, 1995.

[Kakadiaris96] I. A. KAKADIARIS AND D. METAXAS. Model-Based Estimation of 3D Human Motion with Occlusion Based on Active Multi-Viewpoint Selection. In *Proc. CVPR*, pages 81–87, Los Alamitos, California, U.S.A., 1996. IEEE Computer Society.

[Kanade98] T. KANADE, H. SAITO, AND S. VEDULA. The 3D Room: Digitizing Time-Varying 3D Events by Synchronized Multiple Video Streams. Technical Report CMU-RI-TR-98-34, Robotics Institute - Carnegie Mellon University, 1998.

[Kang00] S.B. KANG AND H.Y. SHUM. A Review of Image-based Rendering Techniques. In *IEEE/SPIE Visual Communications and Image Processing (VCIP)*, pages 2–13, 2000.

[Kjeldsen96] R. KJELDSEN AND J.R. KENDER. Toward the use of gesture in traditional user interfaces. In *Proc, of IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 151–156, 1996.

[Koch93] R. KOCH. Dynamic 3D Scene Analysis through Synthesis Feedback Control. *PAMI*, 15(6):556–568, 1993.

[Kuch94] J.J KUCH AND T.S. HUANG. Vision-based hand modeling and gesture recognition for human computer interaction. Master's thesis, University of Illinois, Urbana Champaign, 1994.

[Kutulakos00] K.N. KUTULAKOS AND S.M. SEITZ. A Theory of Shape by Space Carving. *Int. J. Comput. Vision*, 38(3):199–218, 2000.

[Lafortune97] E. LAFORTUNE, S. FOO, K. TORRANCE, AND D. GREENBERG. Non-Linear Approximation of Reflectance Functions. In *SIGGRAPH*, pages 117–126, August 1997.

[Latombe91] J.C. LATOMBE. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[Laurentini94] A. LAURENTINI. The Visual hull Concept for Silhouette-Based Image Understanding. *PAMI*, 16(2):150–162, February 1994.

[Lee95] J. LEE AND KUNII. T. Model-based Analysis of Hand Posture. *IEEE Computer Graphics and Applications*, pages 77–86, 1995.

[Lensch01] H.P.A. LENSCH, W. HEIDRICH, AND H.-P. SEIDEL. A Silhouette-Based Algorithm for Texture Registration and Stitching. *Graphical Models*, 64(3):245–262, 2001.

[Lensch03] H.P.A. LENSCH, J. KAUTZ, M. GOESELE, W. HEIDRICH, AND H.-P. SEIDEL. Image-Based Reconstruction of Spatial Appearance and Geometric Detail. *ACM Transactions on Graphics*, 22(2):27, April 2003.

[Lensch04] H.P.A. LENSCH. *Efficient, Image-Based Appearance Acquisition of Real-World Objects*. PhD thesis, Universität des Saarlandes, Göttingen, Germany, March 2004.

[Leonardis97] A. LEONARDIS, A. JAKLIC, AND F. SOLINA. Superquadrics for Segmenting and Modeling Range Data. *IEEE PAMI*, 19(11):1289–1295, 1997.

[Leung95] M. LEUNG AND Y. YANG. First sight : A human body outline labeling system. *PAMI*, 17(4):359–379, 1995.

[Levoy96] M. LEVOY AND P. HANRAHAN. Light field rendering. In *Proc. of SIGGRAPH'96*, pages 31–42. ACM Press, 1996.

[Li01] M. LI. Correspondence Analysis Between The Image Formation Pipelines of Graphics and Vision. In *Proceedings of the IX Spanish Symposium on Pattern Recognition and Image*

Analysis, pages 187–192, Benicasim(Castellón), Spain, May 2001.

[Li02] M. LI, H. SCHIRMACHER, M. MAGNOR, AND H.-P. SEIDEL. Combining Stereo and Visual Hull Information for Online Reconstruction and Rendering of Dynamic Scenes. In *Proceedings of the 5th Conference on Multimedia Signal Processing*, pages 9–12, St. Thomas, US Virgin Islands, 2002. IEEE.

[Li04a] M. LI, M. MAGNOR, AND H.-P. SEIDEL. Handware-Accelerated Rendering of Photo Hulls. In M.-P. Cani and M. Slater, editors, *Proc. of EUROGRAPHICS 2004*, pages 635–642, 2004.

[Li04b] M. LI, M.A. MAGNOR, AND H.-P. SEIDEL. A Hybrid Hardware-Accelerated Algorithm for High Quality Rendering of Visual Hulls. In *Proc. of Graphics Interface 2004*, pages 41–48, London, Canada, 2004.

[Lipman04] Y. LIPMAN, O. SORKINE, D. COHEN-OR, D. LEVIN, C. RÖSSL, AND H.-P. SEIDEL. Differential Coordinates for Interactive Mesh Editing. In *Shape Modeling International 2004 (SMI 2004)*, pages 181–190, Genova, Italy, 2004. IEEE.

[Log] Logitech Inc. www.logitech.com.

[Log04] J. LOG, M. ERIKSSON, J. SULLIVAN, AND S. CARLSSON. Monocular 3D Reconsruction of Human Motion in Long Action Sequences. In *Proc. of European Conference on Computer Vision (ECCV)*, volume 4, pages 442–455, 2004.

[Loncaric98] S. LONCARIC. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[Lucas81] B. LUCAS AND T. KANADE. An iterative image registration technique with an application to Stereo Vision. In *Proc. DARPA IU Workshop*, pages 121–130, 1981.

[Luck01] J. LUCK, D. SMALL, AND C.Q. LITTLE. Hierarchical 3D Pose Estimation for Articulated Human Body Models from a Sequence of Volume Data. In *Proceedings of the International Workshop on Robot Vision (RoboVis)*, pages 27–34, 2001.

[Luck02] J. LUCK AND D. SMALL. Real-Time Markerless Motion Tracking Using Linked Kinematic Chains. In *Proc. of CVPRIP*, 2002.

[MacCormick00] J. MACCORMICK AND M. ISARD. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. of European Conference on Computer Vision*, volume 2, pages 3–19, 2000.

[Magnor04] M. MAGNOR AND C. THEOBALT. Model-based Analysis of Multi-view Video Data. In *2004 Southwest Symposium on Image Analysis and Interpretation*, pages 41–45, Lake Tahoe, USA, March 2004. IEEE.

[Malassiotis02] S. MALASSIOTIS, N. AIFANTI, AND M.G. STRINTZIS. A Gesture Recognition System Using 3D Data. In *Proceedings of the 1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT-02)*, pages 190–193. IEEE Computer Society, 2002.

[Manfrotto] MANFROTTO. http://www.manfrotto.com.

[Mark03] W.R. MARK, R.S. GLANVILLE, K. AKELEY, AND M.J. KILGARD. Cg: a system for programming graphics hardware in a C-like language. *ACM Trans. Graph. (Proc. of SIGGRAPH'03)*, 22(3):896–907, 2003.

[Marschner98] S. MARSCHNER. *Inverse Rendering for Computer Graphics*. PhD thesis, Cornell University, 1998.

[Martinez95] G. MARTINEZ. 3D Motion Estimation of Articulated Objects for Object-Based Analysis-Synthesis Coding (OBASC). In *VLBV 95*, 1995.

[Matsuyama02] T. MATSUYAMA AND T. TAKAI. Generation, Visualization, and Editing of 3D Video. In *Proc. of 3DPVT'02*, page 234ff, 2002.

[Matusik00] W. MATUSIK, C. BUEHLER, R. RASKAR, S.J. GORTLER, AND L. MCMILLAN. Image-Based Visual Hulls. In *Proc. of SIGGRAPH00*, pages 369–374, 2000.

[Matusik01] W. MATUSIK, C. BUEHLER, AND L. MCMILLAN. Polyhedral Visual Hulls for Real-Time Rendering. In *Proc. of 12th Eurographics Workshop on Rendering*, pages 116–126, 2001.

[Matusik02] W. MATUSIK, H. PFISTER, A. NGAN, P. BEARDSLEY, R. ZIEGLER, AND L. MCMILLAN. Image-based 3D photography using opacity hulls. In *Proc. of SIGGRAPH '02*, pages 427–437. ACM Press, 2002.

[Matusik03] W. MATUSIK, H. PFISTER, M. BRAND, AND L. MCMILLAN. A data-driven reflectance model. *Proc. of SIGGRAPH'03*, 22(3):759–769, 2003.

[Matusik04] W. MATUSIK AND H. PFISTER. 3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. *ACM Trans. Graph.*, 23(3):814–824, 2004.

[McMillan95] L. MCMILLAN AND G. BISHOP. Plenoptic modeling: an image-based rendering system. In *Proc. of SIGGRAPH'95*, pages 39–46. ACM Press, 1995.

[McMillan97] L. MCMILLAN. An image-based approach to three-dimensional Computer Graphics. Technical Report TR97-013, Univeristy of North Carolina at Chapel Hill, 1997.

[Menache95] A. MENACHE. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann, 1995.

[Met] Metamotion Inc. www.metamotion.com.

[Meyer02] M. MEYER, M. DESBRUN, P. SCHRÖDER, AND A. BARR. discrete differentialgeometry operators for triangulated 2-manifolds. In *Proc. of Vis Math*, pages 35–37, 2002.

[Mikić01] I. MIKIĆ, M. TRIVERDI, E. HUNTER, AND P. COSMAN. Articulated body posture estimation from multicamera voxel data. In *Proc. of CVPR*, volume 1, page 455ff, 2001.

[Mitchell97] T.M. MITCHELL. *Machine Learning*. McGraw Hill, 1997.

[Mittal03] A. MITTAL, L. ZHAO, AND DAVISM L.S. Human Body Pose Estimation using Silhouette Shape Analysis. In *Proc. of Conf. on Advanced Video and Signal-based Surveillance (AVSS)*, page 263ff, 2003.

[Moeslund00] T.B. MOESLUND. *Interaction in Virtual Inhabited 3D Worlds*, chapter Interacting with a Virtual World Through Motion Capture, chap. 11. Springer-Verlag, 2000.

[Moeslund01] T.B. MOESLUND AND E. GRANUM. A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, 81(3):231–268, March 2001.

[Moezzi97] S. MOEZZI, L.-C. TAI, AND P. GERARD. Virtual View Generation for 3D Digital Video. *IEEE MultiMedia*, 4(1):18–26, January–March 1997.

[Mulligan00] J. MULLIGAN AND K. DANIILIDIS. View-independent Scene Acquisition for Telepresence. In *Proceedings of the International Symposium on Augmented Reality*, pages 105–108, 2000.

[Murray94] R.M. MURRAY, Z. LI, AND S.S. SASTRY. *A mathematical introduction to robotic manipulation*. CRC Press, 1994.

[Muybridge87] E. MUYBRIDGE. *Animal Locomotion: An Electro-Photographic Investigation of Consecutive Phases of Animal Movements 1872–1885*. University of Pennsylvania, Philadelphia, PA, 1887.

[myd] http://www.mydr.com.au.

[Narayanan98] P.J. NARAYANAN, P. RANDER, AND T. KANADE. Constructing Virtual Worlds using Dense Stereo. In *Proc. of ICCV'98*, pages 3–10, 1998.

[Nelson94] R. NELSON AND R. POLANA. Low Level Recognition of Human Motion. In *Proc. IEEE Workshop on Non-Rigid and Articulated Motion*, pages 77–82, 1994.

[Nishino01] K. NISHINO, Y. SATO, AND K. IKEUCHI. "Eigen-Texture Method: Appearance Compression and Synthesis based on a 3D Model". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1257–1265, 2001.

[Niyogi94] S. NIYOGI AND E. ADELSON. Analyzing and Reconizing Walking Figures in XYT. In *Proc. of IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 469–474, 1994.

[Nyquist28] H. NYQUIST. Certain topics in telegraph transmission theory. *Trans. AIEE*, 47:617–644, April 1928.

[O'Rourke80] J. O'ROURKE AND N.I BADLER. Model-based image analysis of human motion using constraint propagation. *PAMI*, 2(6), 1980.

[Palmer99] S.E. PALMER. *Vision Science - Photons to phenomenology*. MIT Press, 1999.

[Pavlovic97] V.I. PAVLOVIC, R. SHARMA, AND T.S. HUANG. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *Pattern Analysis and Machine Intelligence*, 19(7):677–695, July 1997.

[Phong75] B.-T. PHONG. Illumnation for Computer Generated Pictures. *Communications of the ACM*, pages 311–317, 1975.

[Pingali00] G. PINGALI, J. YVES, A. OPALACH, AND I. CARLBOM. LucentVision: Converting Real World Events into Multimedia Experiences. In *Proc. of Intl Conf. on Multimedia and Expo (ICME)*, pages 1433–1436, 2000.

[Plaenkers03] R. PLAENKERS AND P. FUA. Articulated Soft Objects for Multi-view Shape and Motion Capture. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10), 2003.

[Pol] Polhemus. www.polhemus.com.

[Press02] W.H. PRESS, S.A. TEUKOLSKY, W.T. VETTERLING, AND B.P. FLANNERY. *Numerical recipes in C++*. Cambridge University Press, 2002.

[Ramamoorthi01] R. RAMAMOORTHI AND P. HANRAHAN. A Signal-Processing Framework for Inverse Rendering. In *Proceedings of SIGGRAPH 2001*, pages 117–128. ACM Press, 2001.

[Ramanan03] D. RAMANAN AND D.A. FORSYTH. Findinga nd Tracking People from the Bottom Up. In *Proc. of IEEE Intl. Conference on Computer Vision and Pattern Recognition*, volume 1, pages 467–474, 2003.

[Redert02] A. REDERT, M. OP DE BEECK, C. FEHN, W. IJSSELSTEIJN, M. POLLEFEYS, L.J. VAN GOOL, E. OFEK, I. SEXTON, AND P. SURMAN. ATTEST: Advanced Three-dimensional Television System Technologies. In *3DPVT*, pages 313–319, 2002.

[Rehg94] J.M. REHG AND T. KANADE. Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking. In *ECCV (2)*, pages 35–46, 1994.

[Rehg95] J.M. REHG AND T. KANADE. Model-Based Tracking of Self-Occluding Articulated Objects. In *ICCV*, pages 612–617, 1995.

[Ringer02] M. RINGER AND J. LASENBY. Multiple-Hypothesis Tracking for Automatic Human Motion Capture. In *Proc. of European Conference on Computer Vision*, volume 1, pages 524–536, 2002.

[Rowat79] P.F. ROWAT. *Representing the Spatial Experience and Solving Spatial Problems in a Simulated Robot Environment*. PhD thesis, University of British Columbia, 1979.

[Rushmeier97] H. RUSHMEIER, G. TAUBIN, AND A. GUÉZIEC. Applying Shape from Lighting Variation to Bump Map Capture. In *8th Eurographics Workshop on Rendering Workshop*, pages 35–44, June 1997.

[Rusinkiewicz00] S. RUSINKIEWICZ AND S. MARSCHNER. Measurement I - BRDFs. Script of course CS448C: Topics in Computer Graphics, held at Stanford University, October 2000.

[Sand03] P. SAND, L. MCMILLAN, AND J. POPOVIĆ. Continuous Capture of Skin Deformation. In *Proc. of ACM SIGGRPAH03*, pages 578–586, 2003.

[Sato97] Y. SATO, M.D. WHEELER, AND I. KATSUSHI. Object Shape and Reflectance Modeling from Observation. In *Proceedings of SIGGRAPH 97*, pages 379–388, August 1997.

[Seitz96] S.M. SEITZ AND C.R. DYER. View morphing. In *Proc. of SIGGRAPH '96*, pages 21–30. ACM Press, 1996.

[sel] www.innovision-systems.com.

[Shade98] J. SHADE, S. GORTLER, L.-W. HE, AND R. SZELISKI. Layered depth images. In *Proc. of SIGGRAPH '98*, pages 231–242. ACM Press, 1998.

[Shakhnarovich03] G. SHAKHNAROVICH, P. VIOLA, AND T. DARRELL. Fast Pose Estimation with Parameter-Sensitive Hashing. In *Proc. of IEEE Intl. Conference on Computer Vision*, volume 4, pages 750–757, 2003.

[Shakunaga91] T. SHAKUNAGA. Pose Estimation of Joited Structures. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 566–572, 1991.

[Shannon49] C.E. SHANNON. Communication in the presence of noise. *Proc. Institute of Radio Engineers*, 37(1):10–21, Januar 1949.

[Shimada98] N. SHIMADA. Hand gesture estimation and model refinement using monocular camera - ambiguity limitation by inequality constraints. In *Proc. of 3rd Conf. on Face and Gesture Recognition*, pages 268–273, 1998.

[Sidenbladh00] H. SIDENBLADH, M.J. BLACK, AND J.D. FLEET. Stochastic Tracking of 3D Human Figures using 2D Image Motion. In *Proc. of ECCV*, volume 2, pages 702–718, 2000.

[Sidenbladh02] H. SIDENBLADH, M. BLACK, AND R. SIGAL. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *Proc. of ECCV*, volume 1, pages 784–800, 2002.

[Silaghi98] M.-C. SILAGHI, R. R.PLAENKERS, R. BOULIC, P. FUA, AND D. THALMANN. Local and Global Skeleton Fitting Techniques for Optical Motion Capture. In *Modeling and Motion Capture Techniques for Virtual Environments*, number 1537 in Lecture Notes in Artificial Intelligence, No1537, pages 26–40. Springer, 1998.

[Siteco] SITECO. http://www.siteco.de.

[Slansky70] J. SLANSKY. Recognition of convex blobs. *Pattern Recognition*, 2:3–10, 1970.

[Sminchisescu03] C. SMINCHISESCU AND B. TRIGGS. Kinematic Jump Processes For Monocular 3D Human Tracking. In *Proc.of IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages I 69–76, 2003.

[Sniedovich92] M. SNIEDOVICH. *Dynamic programming*. Marcel Dekker, Inc., 1992.

[Sobotta01] J. SOBOTTA. *Atlas of Human Anatomy*. Lippincot, Williams & Wilkins, 2001.

[Starner97] T. STARNER AND A.P. PENTLAND. Real-Time American Sign Language from Video Using Hidden Markov Models. In *MBR97*, page Chapter 10, 1997.

[Starner98] T. STARNER, J. WEAVER, AND A.P. PENTLAND. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, December 1998.

[Stenger01] B. STENGER, P. R. S. MENDONÇA, AND R. CIPOLLA. Model Based 3D Tracking of an Articulated Hand. In *Proc. of CVPR*, volume II, 2001.

[Stewart02] J. STEWART. *The Pitching Clinic*. Burford Books, Inc., Short Hills, NJ, 2002.

[Sullivan02] J. SULLIVAN AND S. CARLSSON. Recognizing and Tracking Human Action. In *Proc. of European Conference on Computer Vision*, volume 1 of *LNCS 2350*, pages 629–644, 2002.

[Szeliski97] R. SZELISKI AND H.Y. SHUM. Creating full view panoramic image mosaics and environment maps. In *Proc. of SIGGRAPH'97*, pages 251–258. ACM Press, 1997.

[Tamura88] S. TAMURA AND S. KAWASAKI. Recognition of Sign Language Motion Images. *Pattern Recognition*, 21(4):343–353, 1988.

[Theobalt02a] C. THEOBALT, M. MAGNOR, P. SCHÜLER, AND H.-P. SEIDEL. Combining 2D Feature Tracking and Volume Reconstruction for Online Video-Based Human Motion Capture. In *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2002)*, pages 96–103, Beijing, China, 2002. IEEE.

[Theobalt02b] C. THEOBALT, M. MAGNOR, P. SCHÜLER, AND H.-P. SEIDEL. Multi-Layer Skeleton Fitting for Online Human Motion Capture. In *Proceedings of Vision, Modeling, and Visualization VMV 2002*, pages 471–478, Erlangen, Germany, 2002.

[Theobalt03a] C. THEOBALT, J. CARRANZA, M. MAGNOR, AND H.-P. SEIDEL. Enhancing Silhouette-based Human Motion Capture with 3D Motion Fields. In *Proc. of Pacific Graphics'03*, pages 185–193, 2003.

[Theobalt03b] C. THEOBALT, J. CARRANZA, M. MAGNOR, AND H.-P. SEIDEL. A Parallel Framework for Silhouette-based Human Motion Capture. In *Proc. of VMV'03*, pages 207–214, 2003.

[Theobalt03c] C. THEOBALT, M. LI, M. MAGNOR, AND H.-P. SEIDEL. A Flexible and Versatile Studio for Multi-View Video Recording. In *Vision, Video and Graphics 2003*, pages 9–16, Bath, UK, July 2003.

[Theobalt04a] C. THEOBALT, I. ALBRECHT, J. HABER, M. MAGNOR, AND H.-P. SEIDEL. Pitching a Baseball — Tracking High-Speed Motion with Multi-Exposure Images. *ACM Transactions on Graphics (Proc. of SIGGRAPH'04)*, 23(3):540–547, 2004.

[Theobalt04b] C. THEOBALT, J. CARRANZA, M. MAGNOR, AND H.-P. SEIDEL. 3D Video - Being Part of the Movie. *ACM SIGGRAPH Computer Graphics*, 38(3):18–20, August 2004.

[Theobalt04c] C. THEOBALT, J. CARRANZA, M. MAGNOR, AND H.-P. SEIDEL. Combining 3D Flow Fields with Silhouette-based Human Motion Capture for Immersive Video. *Graphical Models*, 66:333–351, September 2004.

[Theobalt04d] C. THEOBALT, E. DE AGUIAR, M. MAGNOR, H. THEISEL, AND H.-P. SEIDEL. Marker-free Kinematic Skeleton Estimation from Sequences of Volume Data. In *ACM Symposium on Virtual Reality Software and Technology (VRST 2004)*, pages 57–64, Hong Kong, China, 2004. ACM.

[Theobalt04e] C. THEOBALT, M. MAGNOR, P. SCHÜLER, AND H.-P. SEIDEL. Combining 2D Feature Tracking and Volume Reconstruction for Online Video-based Human Motion Capture. *International Journal of Image and Graphics*, 4(4):563–583, October 2004.

[Theobalt05] C. THEOBALT, N. AHMED, E. DE AGUIAR, G. ZIEGLER, H. LENSCH, M. MAGNOR, AND H.-P. SEIDEL. Joint Motion and Reflectance Capture for Creating Relightable 3D

Videos. Research Report MPI-I-2005-4-004, Max-Planck-Institut fuer Informatik, Saarbruecken, Germany, April 2005.

[Thrun98] S. THRUN. Learning Maps for Indoor Mobile Robots. *Artificial Intelligence*, 99(1):21–71, 1998.

[Tsai86] R.Y. TSAI. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In *Proc. of CVPR*, pages 364–374, June 1986.

[Van Gelder98] A. VAN GELDER. Approximate Simulation of Elastic Membranes by Triangulated Spring Meshes. *Journal of Graphics Tools*, 3(2):21–41, 1998.

[Vedula99] S. VEDULA, S. BAKER, P. RANDER, R. COLLINS, AND T. KANADE. Three-Dimensional Scene Flow. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV-99)*, volume II, pages 722–729. IEEE, 1999.

[Vedula02] S. VEDULA, S. BAKER, AND T. KANADE. Spatio-Temporal View Interpolation. In *Proceedings of the 13th ACM Eurographics Workshop on Rendering*, pages 65–75, June 2002.

[Vic] www.vicon.com.

[Vid] VideoSavant. http://www.ioindustries.com.

[Vogler98] C. VOGLER AND D. METAXAS. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *Proc. of ICCV*, pages 363–369, 1998.

[Wang03] J.R. WANG AND N. PARAMESWARAN. Survey of Sports Video Analysis: Research Issues and Applications. In *Proc. of VIP2003*, 2003.

[Ward92] G.J. WARD. Measuring and Modeling Anisotropic Reflection. In *Proceedings of SIGGRAPH92*, pages 265–272, 1992.

[wei] www.weinbergervision.com.

[Weik99] S. WEIK AND C.-E. LIEDTKE. Three-dimensional Motion Estimation for Articulated human templates using a sequence of stereoscopic image pairs. In *VCIP99*, 1999.

[Weik01] S. WEIK AND C.-E. LIEDTKE. Hierarchical 3D Pose Estimation for Articulated Human Body Models from a Sequence of Volume Data. In *Robot Vision*, 2001.

[Weng90] J. WENG, P. COHEN, AND M. HERNIOU. Calibration of Stereo Cameras Using a Non-Linear Distortion Model. In *ICPR90*, pages Vol–I246–253, 1990.

[Whittle96] M.W. WHITTLE. *Gait Analysis*. Butterworth Heinemann, 1996.

[Wren97] C.R. WREN, A. AZARBAYEJANI, T. DARRELL, AND A. PENTLAND. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[Wu99a] Y. WU AND S. HUANG. Human Hand Modeling, Analysis and Animation in the Context of HCI. In *Proc. of IEEE Intl. Conf. on Image Processing*, pages 6–10, 1999.

[Wu99b] Y. WU AND HUANG T.S. Capturing articulated human hand motion: A divide-and-conquer approach. In *Proc. of ICCV*, pages 606–611, 1999.

[Wu01] Y. WU, J. LIN, AND T. HUANG. Capturing Natural Hand Articulation. In *Proc. of ICCV*, pages 426–432, 2001.

[Wuermlin02] S. WUERMLIN, E. LAMBORAY, O.G. STAADT, AND M.H. GROSS. 3D Video Recorder. In *Proc. of Pacific Graphics 2002,*, pages 325–334, 2002.

[Yonemoto00] S. YONEMOTO, D. ARITA, AND R. TANIGUCHI. Real-time Human Motion Analysis and IK-based Human Figure Control. In *Proceedings of IEEE Workshop on Human Motion*, pages 149–154, 2000.

[Yu98] Y. YU AND J. MALIK. Recovering Photometric Properties of Architectural Scenes from Photographs. In *Proceedings of SIGGRAPH 98*, pages 207–218, 1998.

[Yu99] Y. YU, P. DEBEVEC, J. MALIK, AND T. HAWKINS. Inverse Global Illumination: Recovering Reflectance Models of Real Scenes From Photographs. In *Proc. of SIGGRAPH 99*, pages 215–224, August 1999.

[Yu03] W. YU, C. XU, H.W. LEONG, Q. TIAN, Q. TANG, AND K.W. WAN. Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 11–20, 2003.

[Zhang99] R. ZHANG, P.-S. TSAI, J. CRYER, AND M. SHAH. Shape from Shading: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.

[Zimmermann87] T.G. ZIMMERMANN. A Hand Gesture Interface Device. In *Proc. of Human Factors in Computing Systems and Graphics Interface*, pages 189–192, 1987.

[Zitnick04] C.L. ZITNICK, S.B. KANG, M. UYTTENDAELE, S. WINDER, AND R. SZELISKI. High-quality video view interpolation using a layered representation. *ACM Trans. Graph. (Proc. of SIGGRAPH'04)*, 23(3):600–608, 2004.

Curriculum Vitae – Lebenslauf

Curriculum Vitae

1976	born in Saarbrücken, Germany
1982-1986	Grundschule Aschbach
1986-1995	Geschwister-Scholl-Gymnasium Lebach
1995-1996	Study of Chemistry, University of Saarland, Saarbrücken, Germany
1996-2001	Study of Computer Science, University of Saarland
1999-2000	Study of Artificial Intelligence (specializing in Intelligent Robotics), University of Edinburgh, Scotland
10/2000	MSc degree in Artificial Intelligence, University of Edinburgh
10/2000	John Howe Award, University of Edinburgh
10/2000	Best MSc Student in Artificial Intelligence Award, University of Edinburgh
4/2001	Diplom (Dipl.-Inf.) degree in Computer Science, University of Saarland
5/2001-	Ph.D. Student at the Max-Planck-Institut für Informatik, Saarbrücken, Germany
4/2005	ATI Fellowship 2005/2006

Lebenslauf

1976	geboren in Saarbrücken
1982-1986	Grundschule Aschbach
1986-1995	Geschwister-Scholl-Gymnasium Lebach
1995-1996	Studium der Chemie, Universität des Saarlandes, Saarbrücken
1996-2001	Studium der Informatik, Universität des Saarlandes
1999-2000	Studium der Künstlichen Intelligenz (Spezialgebiet: Intelligent Robotics), University of Edinburgh, Schottland
10/2000	Abschluß als MSc in Artificial Intelligence, University of Edinburgh
10/2000	John Howe Preis, University of Edinburgh
10/2000	Preis als bester MSc Student in Künstlicher Intelligenz, University of Edinburgh
4/2001	Abschluß als Diplom Informatiker (Dipl.-Inf.), Universität des Saarlandes
5/2001-	Promotion am Max-Planck-Institut für Informatik, Saarbrücken
4/2005	ATI Fellowship 2005/2006