Filtering and Optimization Strategies for Markerless Human Motion Capture with Skeleton-based Shape Models.

Dissertation

zur Erlangung des Grades des
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

vorgelegt von

Juergen Gall

Saarbrücken
2009

Datum des Kolloqiums: 07.07.2009

Dekan der Naturwissenschaftlich-Technischen Fakultät I:
Prof. Dr. Joachim Weickert

Mitglieder des Prüfungsausschusses:
Vorsitzender: Prof. Dr. Philipp Slusallek
1. Gutachter: Prof. Dr. Hans-Peter Seidel
2. Gutachter: Prof. Dr. Bodo Rosenhahn
3. Gutachter: Prof. Dr. Luc van Gool
Akademischer Mitarbeiter: Dr. Meinard Müller

Abstract

Since more than 2000 years, people have been interested in understanding and analyzing the movements of animals and humans which lead to the development of advanced computer systems for motion capture. Although marker-based systems for motion analysis are commercially successful, capturing the performance of a human or an animal from a multi-view video sequence without the need for markers is still a challenging task. The most popular methods for markerless human motion capture are model-based approaches that rely on a surface model of the human with an underlying skeleton. In this context, markerless motion capture seeks for the pose, i.e., the position, orientation, and configuration of the human skeleton that is best explained by the image data. In order to address this problem, we discuss the two questions:

1. What are good cues for human motion capture? *Typical cues for motion capture are silhouettes, edges, color, motion, and texture. In general, a multi-cue integration is necessary for tracking complex objects like humans since all these cues come along with inherent drawbacks. Besides the selection of the cues to be combined, reasonable information fusion is a common challenge in many computer vision tasks. Ideally, the impact of a cue should be large in situations when its extraction is reliable, and small, if the information is likely to be erroneous. To this end, we propose an adaptive weighting scheme that combines complementary cues, namely silhouettes on one side and optical flow as well as local descriptors on the other side. Whereas silhouette extraction works best in case of homogeneous objects, optical flow computation and local descriptors perform better on sufficiently structured objects. Besides image-based cues, we also propose a statistical prior on anatomical constraints that is independent of motion patterns. Relying only on image features that are tracked over time does not prevent the accumulation of small errors which results in a drift away from the target object. The error accumulation becomes even more problematic in the case of multiple moving objects due to occlusions. To solve the drift problem for tracking, we propose an analysis-by-synthesis framework that uses reference images to correct the pose. It comprises an occlusion handling and is successfully applied to crash test video analysis.*

2. Is human motion capture a filtering or an optimization problem? *Model-based human motion capture can be regarded as a filtering or an optimization problem. While local optimization offers accurate estimates but often looses track due to local optima, particle filtering can recover from errors at the expense of a poor accuracy due to overestimation of noise. In order to overcome the drawbacks of local optimization, we introduce a novel global stochastic optimization approach for markerless human motion capturing that is derived from the mathematical theory on interacting particle systems. We call the method* interacting simulated annealing *(ISA) since it is based on an interacting particle system that converges to the global optimum similar to simulated annealing. It estimates the human pose without initial information, which is a challenging optimization problem in a high dimensional space. Furthermore, we propose a tracking framework that is based on this optimization technique to achieve both the robustness of filtering strategies and a remarkable accuracy.*

In order to benefit from optimization and filtering, we introduce a multi-layer framework that combines stochastic optimization, filtering, and local optimization. While the first layer relies on interacting simulated annealing, the second layer refines the estimates by filtering and local optimization such that the accuracy is increased and ambiguities are resolved over time without imposing restrictions on the dynamics.

In addition, we propose a system that recovers not only the movement of the skeleton, but also the possibly non-rigid temporal deformation of the 3D surface. While large scale deformations or fast movements are captured by the skeleton pose and approximate surface skinning, true small scale deformations or non-rigid garment motion are captured by fitting the surface to the silhouette. In order to make automatic processing of large data sets feasible, the skeleton-based pose estimation is split into a local one and a lower dimensional global one by exploiting the tree structure of the skeleton.

Our experiments comprise a large variety of sequences for qualitative and quantitative evaluation of the proposed methods, including a comparison of global stochastic optimization with several other optimization and particle filtering approaches.

Zusammenfassung

Seit mehr als 2000 Jahren interessieren sich Menschen für den Ursprung und die Analyse menschlicher und tierischer Bewegungsabläufe was letztendlich zur Entwicklung modernster Computersysteme zur Bewegungserfassung führte. Obwohl sich Systeme mit aktiven oder passiven Markern am Markt erfolgreich durchgesetzt haben, ist die rein bildbasierte Bewegungserfassung von Menschen und Tieren mittels mehrerer Kameras immer noch eine große Herausforderung. Unter den markerlosen Verfahren sind modellbasierte Ansätze am weitesten verbreitet. Diese beruhen auf ein Oberflächenmodell des menschlichen Körpers dessen Verformungen über ein Skelett gesteuert werden. In diesem Zusammenhang reduziert sich die markerlose Bewegungserfassung auf das Finden der menschlichen Pose, die am besten mit den Bilddaten übereinstimmt, wobei die Pose durch die Position, Orientierung und Konfiguration des menschlichen Skelettes definiert ist. Wir widmen uns diesem Problem, indem wir die folgenden zwei Fragestellungen angehen:

1. Was sind geeignete Hinweisreize zur menschlichen Bewegungserfassung? *Typische Hinweisreize zur Bewegungserfassung sind Silhouetten, Kanten, Farbe, Bewegung und Oberflächenstruktur. Zur Erfassung komplexer Objekte wie Menschen ist im Allgemeinen eine Kombination von mehreren Hinweisreizen notwendig, da jeder einzelne Hinweisreiz einen spezifischen Nachteil aufweist. Neben der Auswahl geeigneter Hinweisreize ist die Informationsfusion eine generelle Herausforderung für viele Anwendungen des maschinellen Sehens. Idealerweise sollte der Einfluss eines einzelnen Reizes groß sein, wenn er zuverlässig extrahiert werden kann, und klein, wenn die gewonnene Information voraussichtlich fehlerhaft ist. Aus diesem Grund schlagen wir ein adaptives Gewichtungsschema vor, das komplementäre Hinweisreize vereint. Dies sind zum einen Silhouetten und zum anderen optischer Fluss sowie lokale Deskriptoren. Während die Silhouettenextraktion am besten für homogene Objekte funktioniert, eignen sich optischer Fluss und lokale Deskriptoren besser für ausreichend strukturierte Oberflächen. Neben visuellen Hinweisreizen schlagen wir die statistische Modellierung anatomischer Einschränkungen des Skelettes vor, und zwar unabhängig von etwaigen Bewegungsmustern.*
Wenn nur Bildmerkmale verwendet werden, die über die Zeit verfolgt werden, besteht die Gefahr, dass sich kleine Schätzfehler zu einem unkorrigierbaren Fehler aufsummieren. Dies führt zu einer Drift weg vom eigentlichen Ziel. Die Fehlerakkumulation tritt verstärkt bei Verdeckungen auf, wie sie bei mehreren Objekten häufig vorkommen. Um dieses Driftproblem zu lösen, schlagen wir ein Analyse-durch-Synthese-Verfahren vor, das synthetische Referenzbilder verwendet um Schätzfehler zu korrigieren. Darüberhinaus beinhaltet es ein aktives System zu Erkennung und Handhabung von Verdeckungen. Das Verfahren wurde bereits erfolgreich zur Crashtestanalyse eingesetzt.

2. Ist die menschliche Bewegungserfassung ein Filter- oder Optimierungsproblem? *Modellbasierte Bewegungserfassung kann als Filter- oder Optimierungsproblem betrachtet werden. Während lokale Optimierungsverfahren die menschliche Pose genau schätzen, aber häufig das Objekt auf Grund von lokalen Optima verlieren, sind Partikelfilteransätze fähig sich von Schätzfehlern zu erholen. Allerdings ist die Genauigkeit von Filteransätzen häufig ungenügend, da eine ungenaue Problemmodellierung meist mit einer Überschätzung des Signalrauschens kompensiert wird. Um die Nachteile der lokalen Optimierung zu überwinden, präsentieren wir ein stochastisches Verfahren zur globalen Optimierung. Das Verfahren ist für die markerlose Bewegungserfassung geeignet und leitet sich von der mathematischen Theorie über in-*

teragierende Partikelsysteme ab. Wir bezeichnen das Verfahren mit Interacting Simulated Annealing *(ISA), da es auf einem interagierenden Partikelsystem basiert, das ähnlich wie Simulated Annealing zum globalen Optimum konvergiert. Es schätzt die menschliche Pose ohne zusätzliches Vorwissen, was ein schwieriges Optimierungsproblem in einem hochdimensionalen Suchraum ist. Darüberhinaus führen wir ein Bewegungserfassungssystem ein, das auf diesem Optimierungsverfahren basiert und die Zuverlässigkeit von Filteransätzen mit einer bemerkenswerten Genauigkeit vereint.*

Um von den positiven Eigenschaften von Filter- und Optimierungsansätzen gleichzeitig zu profitieren, stellen wir ein mehrstufiges System vor, das stochastische Optimierung, Filterung und lokale Optimierung kombiniert. Während die erste Stufe auf Interacting Simulated Annealing beruht, verfeinert die zweite Stufe die geschätzte Pose mittels Filterung und lokaler Optimierung, so dass die Genauigkeit verbessert wird und Ambiguitäten über die Zeit aufgelöst werden. Hierbei werden jedoch keine Restriktionen bezüglich der Bewegungsdynamik auferlegt.

Des Weiteren schlagen wir ein System vor, das nicht nur die Bewegungen des menschlichen Skelettes erfasst sondern auch mögliche unstarre Deformationen der Oberfläche. Während grobe Deformationen oder schnelle Bewegungen von der Skelettpose und der damit verbundenen Oberflächendeformation eingefangen werden, werden feine Deformationen und die unstarre Bewegung von Kleidungsstücken durch das Anpassen der Oberfläche an die Bildsilhouette erfasst. Um auch große Datenmengen automatisch verarbeiten zu können, wird die Schätzung der Skelettpose in ein lokales und ein globales Optimierungsproblem mit einem kleineren Suchraum aufgeteilt, wobei die Baumstruktur des menschlichen Skelettes ausgenützt wird.

Unsere Experimente beinhalten eine große Bandbreite an Bildsequenzen zur qualitativen und quantitativen Evaluierung der vorgestellten Verfahren einschließlich eines Vergleiches unseres globalen stochastischen Optimierungsverfahrens mit mehreren anderen Partikelfilter- und Optimierungsansätzen.

Acknowledgements. I would like to thank **Supervisors** Bodo Rosenhahn, Hans Peter Seidel. **Funding** Max Planck Center for Visual Computing and Communication, Cluster of Excellence on Multimodal Computing and Interaction. **Collaborators** Edilson de Aguiar, Hernán Badino, Thomas Brox, Daniel Cremers, Stefan Gehrig, Nils Hasler, Jürgen Potthoff, Christoph Schnörr, Carsten Stoll, Christian Theobalt, Thorsten Thormählen, Michael Wand. **Data Acquisition** Boris Ajdin, Elke Galli, Dong Han, Maria Jacob, Sonja Lienard, Mohammed Shaheen, Silke Thory. **External Data** Stefano Corazza, Adrian Hilton, Daimler Research, Leonid Sigal, Jonathan Starck. **Software** Ilya Baran, Yan Ke, David Lowe. **Images** Bartleby, Bibliothèque Interuniversitaire de Médecine (Paris), Dariu Gavrila, Vitronic. **Local Support** Sabine Budde, Martin Fuchs, Conny Liegl.

Contents

1

Introduction

> *For if one of the parts of an animal be moved, another must be at rest, and this is the purpose of their joints; animals use joints like a centre, and the whole member, in which the joint is, becomes both one and two, both straight and bent, changing potentially and actually by reason of the joint.*
> – Aristotle

The interest in understanding the movements of animals and humans goes back to the Greek philosopher Aristotle (384-322 B.C.E.) who studied the gait of animals [Ari07] and regarded the bodies as mechanical systems consisting of limbs and joints. *Kinematic trees*, nowadays called kinematic chains, for modeling human motion can also be found in the sketchbooks of Leonardo da Vinci (1452-1519), see [RKM08, Chapter 1]. The first detailed study on human motion, which contained also quantitative measurements, was performed by Alfonso Borelli (1608-1679) [Bor89]. He discovered that the human movement follows mechanical principles. While Borelli analyzed the movements still by eye, Wilhlem Weber (1804-1891) and Eduard Weber (1806-1871) established a theory of locomotion by analyzing the human gait with accurate chronometers and telescopes [WW92]. At the end of the 19th century, the technological progress revolutionized the field of motion analysis. The French astronomer Pierre-Jules-César Janssen (1824-1907) invented a multi-exposure camera that took forty-eight exposures in seventy-two seconds on a daguerreotype disc [HM96]. The technique called geometric chronophotography was used by Etienne-Jules Marey (1830-1904) to study locomotion of animals and humans [Mar73, Mar94]. He also introduced marker-based human motion capture where the subject wore a black dress with metal buttons and shinning bands to mark limbs, see Figure 1.1. Another important representative for chronophotography was Eadweard Muybridge (1830-1904) whose famous work “Animals in Motion” [Muy57] contains several movement studies captured by a series of cameras. The marker-based approach of Marey was further developed by Wilhelm Braune and Otto Fischer [BF87]. They attached light rods to the subject’s limbs to study human motion. This principle is known as moving light displays and was used by psychologist Gunnar Johannsson to investigate human motion perception [Joh76]. He discovered that the sparse spatio-temporal information of the markers is enough for humans to recognize different activities from a sequence of images showing only the set of light dots. This fundamental observation laid the foundation of nowadays marker-based human motion capture systems. The step towards computer-driven motion analysis was taken in the 80s, when Rashid presented a computer system for tracking and clustering the points of moving light displays [Ras80]. The rapid technological

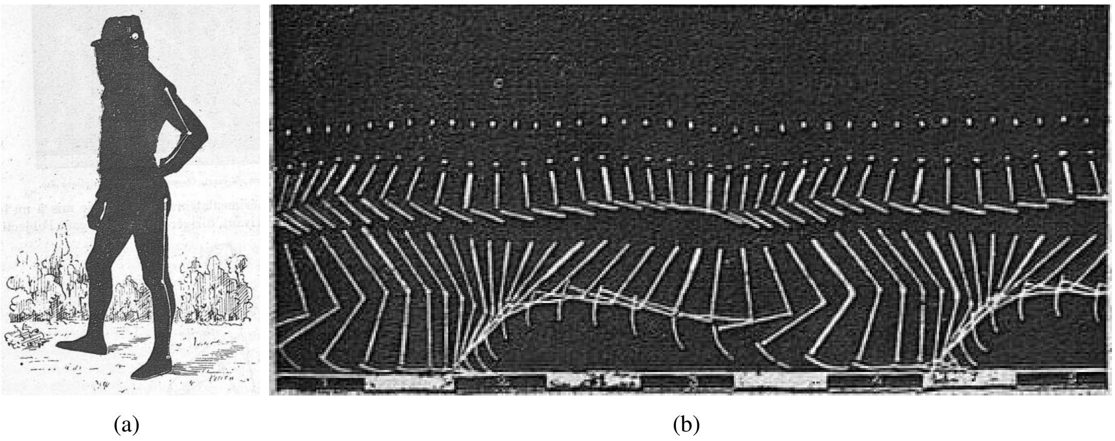


Figure 1.1: Marker-based motion capture by Etienne-Jules Marey (1830-1904).[1]**a)** Black suit with marked limbs. **b)** Geometric chronophotograph of a running sequence with such a black suit.

progress of computers and camera systems since then has resulted in commercial marker-based human motion capture systems, e.g. [Ari08, Mot08, Qua08, Sim08, Vic08], that are successfully applied to various areas like entertainment, movement analysis, and engineering. These systems still follow the basic principle of Marey, Braune, and Fischer: passive or active markers are attached to the subject and the subject's movement is captured by several high-speed cameras. From the tracked motion of the markers, the motion of the kinematic model is estimated.

1.1 Why human motion capture?

Since more than 2000 years, people have been fascinated by the movements of animals and humans which lead to the development of advanced computer systems for motion capture. It is not only the general interest in understanding nature that drives people working in this field, but it is also the wide range of applications that affect our everyday lives. A variety of applications can be found in surveys like [Gav99]. For instance, human motion capture data is used for character animation in games and movies. Other examples from the field of virtual reality are avatars, interactive virtual worlds, 3D-TV, and teleconferencing. In the course of the rapid increase of web-based applications, it is expected that the market for video-based human motion capture systems will grow further. Alone the company Oxford Metrics Group, which sales the Vicon systems, reported a turnover of £19.6m for 2007 [OMG07]. Even though the film and game industry are the most popular customers for human motion capture systems, the industrial applications also include virtual training, robotics, ergonomics, virtual design, and crash analysis. Other areas are sport science and medical diagnostics where gait analysis, rehabilitation, sports performance, biomechanical research, and medical robotics are only some examples for applications. Furthermore, surveillance and model-based video coding are worth mentioning.

[1] ©Bibliothèque interuniversitaire de médecine (Paris).

1.2 Why markerless motion capture?

Although marker-based systems for motion analysis are commercially successful, they have several drawbacks. The attachment of markers is not only time-consuming and uncomfortable for the subject to wear it can also significantly change the pattern of locomotion [FWA03]. In general, the requirements on the environment, lighting conditions, and clothing limit their application, e.g. for outdoor sequences or natural environments. Furthermore, a manual intervention is sometimes necessary when tracking or identification of markers fails. Since the skeletal movement is usually estimated from a finite set of markers placed on the skin by assuming rigid body parts, soft tissue artifacts are introduced and affect the estimation of the movement, see e.g. [RBN$^+$97] or [RKM08, Chapter 15]. This problem becomes more evident when the markers are placed on clothes that are not tight. Hence, markerless motion capture is a promising alternative where the motion is estimated directly from the images without attached markers.

In contrast to marker-based systems, markerless human motion capture is still a challenging task. Although substantial research has been conducted in this field since the 80s [OB80, MG01a, MHK06], there are only few companies like [Org08] that offer commercial products. The wide range of applications and the potential benefits of markerless human motion capture make the development of such a system interesting for researchers and companies where the ultimate goal is

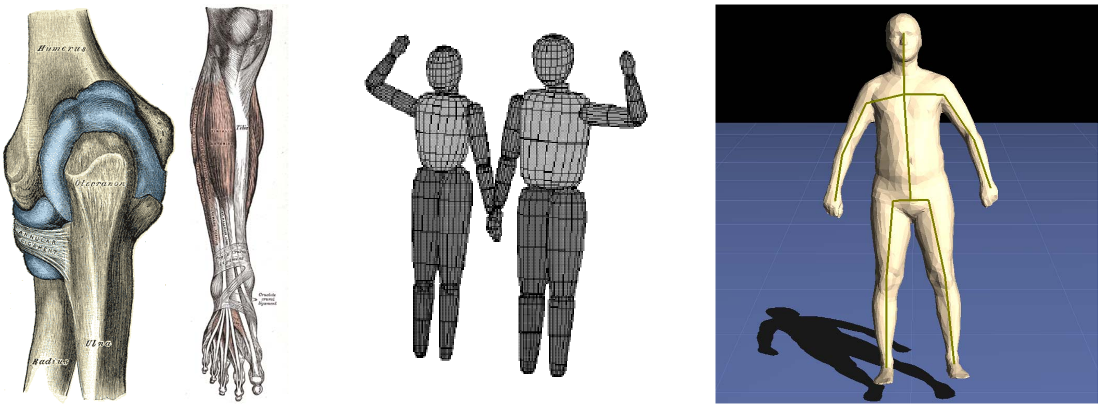
> *the design of a markerless tracking system that captures the motion of a human in an outdoor scene with the accuracy of a commercial marker-based system in an indoor scene.*

This means that the system is expected to be *robust* enough for outdoor scenes that are more challenging than a controlled laboratory environment since the background is usually non-static, the lighting conditions can be difficult, and multiple moving objects and occlusions need to be handled. Furthermore, its *accuracy* must also be measurable against marker-based systems.

1.3 Why model-based motion capture?

A prior knowledge that is frequently used in markerless human motion capture are surface models of humans with underlying skeletons, see Figure 1.2. This is a very natural representation that is motivated by the anatomy of humans and animals. Admittedly, the human body is very complex. The skeleton of an adult, for example, consists of 206 distinct bones [Gra18] and the estimation of each bone is beyond the realm of possibility for any human motion capture system. On this account, the models are more or less approximations of humans where the degrees of freedom (DOF) of the kinematic skeleton are reduced to a manageable size – which is usually between 25 and 40 DOF. While the shape of the model was approximated with simple cylinders [Hog83] at the beginning of the 80s, the substantial progress in computer graphics and 3D scanning technology allows nowadays more detailed 3D surface models like SCAPE [ASK$^+$05]. Another aspect is the surface deformation according to the movement of the underlying skeleton. The simplest methods are rigid transformations, which are used for articulated models, and skeleton-subspace deformations, but also more realistic deformation schemes have been developed mainly for character animation in the last years, see e.g. [LCF00].

Model-based approaches are especially suited to markerless motion capture since they provide an intuitive way to constrain the search space by the degrees of freedom of the skeleton. Furthermore, the advances in the field of computer graphics have made the acquisition, processing, and



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

Figure 1.2: **From left to right: a, b)** Anatomical illustrations of an elbow joint and muscles of a leg taken from [Gra18]. **c)** 3D models of humans taken from [Gav96]. **d)** Scanned 3D model with underlying skeleton. The model was rigged by Pinocchio [BP07].

deformation of 3D surface models efficient and common practice. Hence, from a model-based perspective, motion capture seeks for the pose of the human, i.e., the position, orientation, and deformation of the human skeleton that is best explained by the image data.

1.4 Optimization and Filtering

The essential question for markerless human motion capture that needs to be solved is:

> *What is the best way to determine the sequence of human poses that fits a given image sequence best?*

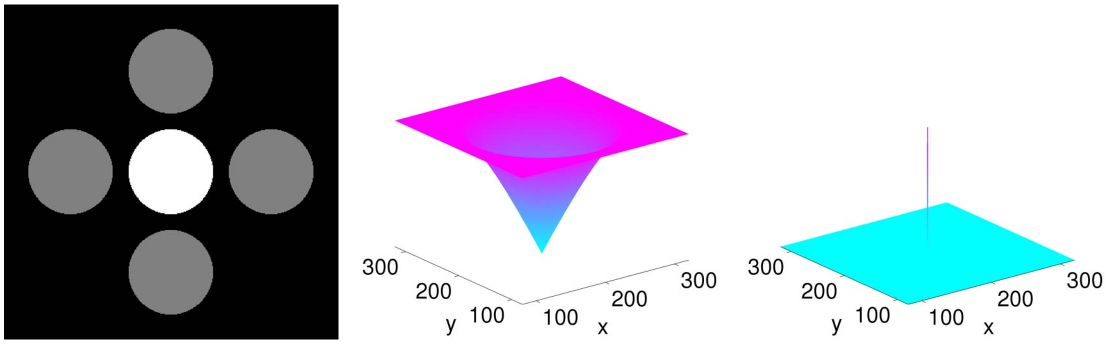
The techniques for model-based human motion capture that appeared in the last decade can be classified into two groups, namely filtering and optimization strategies. The filtering approaches regard the images as noisy observations of the unknown true state that is the position, rotation, and joint configuration of the human model in each frame. They assume that the dynamics of the human can be modeled by a stochastic process, usually a Markov process, and that the images are generated from the true pose by a stochastic process disturbed by noise. Depending on the underlying processes, the solutions are based on Kalman filtering [Kal60] or particle filtering [GSS93].

The optimization approaches assume the existence of a cost function based on some image features such that the true pose is a global optimum of the function. The cost function may depend on the estimates from previous frames as it occurs from Bayesian modeling where a posterior distribution for a single frame is optimized. After optimization, however, only the estimate but not the distribution is taken into account for the next frame – in contrast to filtering where the uncertainty in the estimate is propagated over time. Since standard global optimization techniques are very expensive, local optimization algorithms like gradient descent are commonly used. So far, neither filtering nor optimization performed significantly better than the other, since both strategies have advantages and disadvantages.

Filtering methods are known to be robust and can recover from errors since they can model noise and resolve ambiguities over time. Particularly, particle filters are popular due to the multimodality of the solution since they approximate a distribution instead of a single value. Furthermore, they do not require linearity of the involved model like the Kalman filter. However, the available convergence results assume that the underlying stochastic processes are known – which in practice is rarely the case. Finding the right models for human motion tracking – both for the dynamics and for the likelihood – is very difficult and so far unsolved. Instead, the weakness of the models is often handled by overestimating the noise yielding a poor performance in high dimensional spaces.

Energy minimization approaches are usually more flexible with regard to the underlying model and can be solved by local or global optimization. While global optimization is limited by the time constraints of tracking, local optimization suffers from local optima. This has the effect that tracking fails in case of fast motion and the methods usually cannot recover from errors.

In summary, the decision for filtering or optimization is not only a trade-off between *robustness* and *accuracy*, but it also affects the perspective on the problem and thus the modeling. A well approximated likelihood for filtering is usually not an ideal energy function for optimization and vice versa as illustrated by the synthetic 2D example in Figure 1.3. The cone-shaped energy function differs from the unique solution for the likelihood, namely the Dirac measure, on the one hand. On the other hand, an energy function that is constant except at the global minimum like a Dirac measure is a worst-case scenario for optimization since it can only be solved by guessing the solution.



(a) (b) (c)

Figure 1.3: Synthetic 2D example with a disc. **From left to right: a)** The white circle is the noise-free silhouette of the disk located at the center of the image, namely at $(200, 200)$. The gray discs indicate samples which were taken from 80 to 320 in x- and y-directions. **b)** The energy function can be modeled as cone-shaped function with global minimum at $(200, 200)$. In this case, the non-overlapping pixels between the white silhouette and the gray samples are counted. At $(200, 200)$, the gray disc covers completely the white silhouette which yields an energy of zero. **c)** Since the image is noise-free, we know that the image can only be generated by a gray disc that completely covers the silhouette. Hence, the exact likelihood is a Dirac-measure that is 1 at $(200, 200)$ and 0 otherwise. While there exists a unique solution for the likelihood, namely the Dirac measure, the energy function can be modeled in various ways.

1.5 Contribution

In this work, we address the question *"What is the best way to determine the sequence of human poses that fits a given image sequence best?"* from Section 1.4 by discussing the following two subquestions in the context of markerless human motion capture with skeleton-based shape models:

1. *What are good cues for human motion capture?*
2. *Is human motion capture a filtering or an optimization problem?*

1.5.1 What are good cues for human motion capture?

Typical cues for motion capture are silhouettes, edges, color, motion, and texture. In general, a multi-cue integration is necessary for tracking complex objects like humans since all these cues come along with inherent drawbacks. Besides the selection of the cues to be combined, reasonable information fusion is a common challenge in many computer vision tasks. Ideally, the impact of a cue should be large in situations when its extraction is reliable, and small, if the information is likely to be erroneous. We propose an adaptive weighting scheme that combines complementary cues, namely silhouettes on one side and optical flow as well as local descriptors on the other side. Whereas silhouette extraction works best in case of homogeneous objects, optical flow computation and local descriptors perform better on sufficiently structured objects [BRGC09].

Relying only on image features that are tracked over time does not prevent the accumulation of small errors which results in a drift away from the target object. The error accumulation becomes even more problematic in the case of multiple moving objects due to occlusions. To solve the drift problem for tracking, we propose an analysis-by-synthesis framework that uses reference images to correct the pose. It comprises an occlusion handling that discards image features which are detected to be occluded [GRS06, GRS08a].

The proposed concept can be applied to crash test sequences in order to estimate position and orientation of a dummy's head for instance. The analysis of crash test videos is an important task for the automotive industry in order to improve the passive safety components of cars. In particular, the motion estimation of crash test dummies helps to improve the protection of occupants and pedestrians. In contrast to conventional marker-based systems which provide only sparse 3D measurements, our approach estimates all six degrees of freedom of dummy body parts like the head. This opens up new opportunities for analyzing pedestrian crashes where many biomechanical effects are not fully understood [GRGS08, GBG08].

Besides image-based cues, prior knowledge is another important source of information for markerless human motion capture. The body shape and the kinematic structure of humans are already exploited by a surface model with an underlying skeleton. It reduces significantly the search space but it does not take into account physical restrictions on the kinematic model. For instance, anatomical limits of joints like knees and elbows constrain the search space as well as unrealistic self-intersections. Instead of modeling the physical restrictions as hard constraints, we allow for the simplification and approximation of the kinematic model by integrating this prior knowledge as soft constraints [GRBS06].

1.5.2 Is human motion capture a filtering or an optimization problem?

Markerless human motion capture can be regarded as a filtering or an optimization problem as discussed in Section 1.4. While the filtering approaches rely often on particle filters, the optimization problem is commonly solved by iterative methods like gradient descent. Local optimization provides very accurate results given that the state vector is initialized near the global optimum. Since it searches only for the locally best solution, it usually cannot recover from errors and requires an initialization. Without additional prior information, the tracking often fails in case of fast motions and ambiguities. In order to overcome the drawbacks of local optimization, global optimization like simulated annealing can be applied for motion capture [CMC$^+$06]. Filtering approaches exploit temporal coherence, handle noise, and are able to recover from errors, but they are usually too imprecise for motion analysis in high dimensional spaces. For this reason, a heuristic approach, called annealed particle filter [DBR00], has been proposed to combine the ideas of particle filtering and simulated annealing for motion capturing. The annealed particle filter, however, does not perform annealing in the classical sense where the temperature is monotonically decreased, but relies on the fluctuating survival rate of the particles. Particle filters belong to the more general class of interacting particle systems. They approximate a distribution of interest by a finite number of particles where the particles interact between the iteration steps. In the context of filtering, they are known as particle filters and approximate the posterior distribution, but there also exist interacting particle systems with annealing properties, which makes them suitable for optimization.

We introduce a novel global stochastic optimization approach for markerless human motion capturing that is derived from the mathematical theory on interacting particle systems [Mor04]. We call the method *interacting simulated annealing* (ISA) since it is based on an interacting particle system that converges to the global optimum similar to simulated annealing [GPS$^+$07, GRS08b]. It estimates the human pose without initial information, which is a challenging optimization problem in a high dimensional space and is essential for initialization and texture acquisition [GRS07]. Furthermore, we propose a tracking framework that is based on this optimization technique to achieve both the *robustness* of filtering strategies and a remarkable *accuracy* [GPS$^+$07]. The latter is demonstrated by a quantitative error analysis that includes the `HumanEva-II` benchmark [SB06] and a comparison with several optimization and particle filtering approaches.

In order to benefit from optimization and filtering, we introduce a multi-layer framework that combines stochastic optimization, filtering, and local optimization. While the first layer relies on interacting simulated annealing, the second layer refines the estimates by filtering and local optimization such that the accuracy is increased and ambiguities are resolved over time without imposing restrictions on the dynamics [GRBS08].

In addition, we propose a system that recovers not only the movement of the skeleton, but also the possibly non-rigid temporal deformation of the 3D surface. While large scale deformations or fast movements are captured by the skeleton pose and approximate surface skinning, true small scale deformations or non-rigid garment motion are captured by fitting the surface to the silhouette. In order to make automatic processing of large data sets feasible, the skeleton-based pose estimation is split into a local one and a lower dimensional global one by exploiting the tree structure of the skeleton. We show on various sequences that our approach can capture the 3D motion of animals and humans accurately even in the case of rapid movements and wide apparel like skirts [GSA$^+$09].

1.6 Assumptions

Throughout the paper we assume that a skeleton-based shape model as in Figure 1.2 d) is available. Hence, we will not cover the acquisition of such models. In general, the 3D surface models can be acquired by a 3D scanner as shown in Figure 1.4 a) or extracted from silhouettes or stereo data [KM98, HBG$^+$00, PF03, MTHC03, CBK05]. Other sources are repositories of

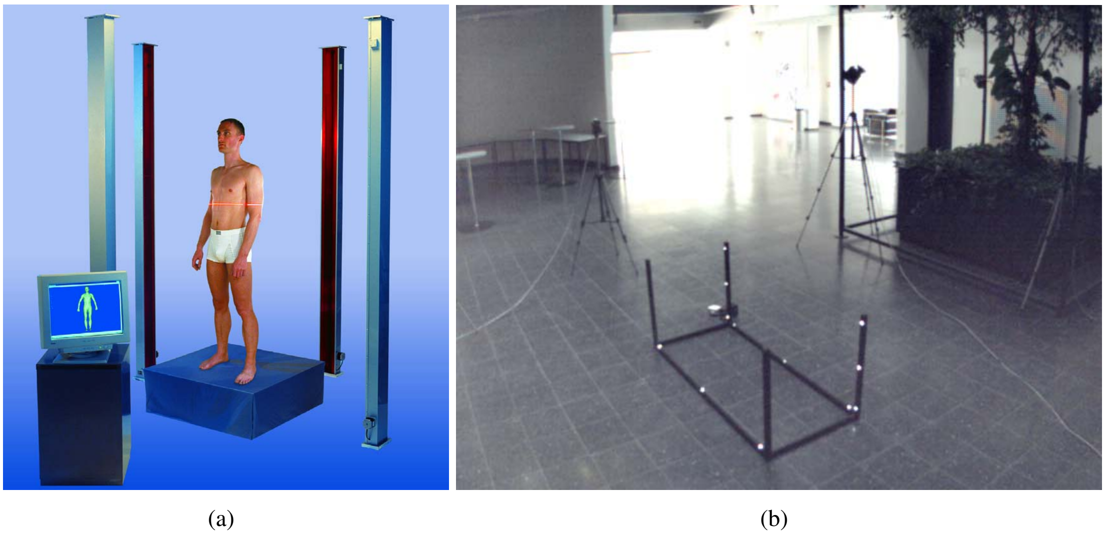

(a) (b)

Figure 1.4: **From left to right: a)** Human body scanner [Vit08]. **b)** Mobile multi-camera capture system with calibration target.

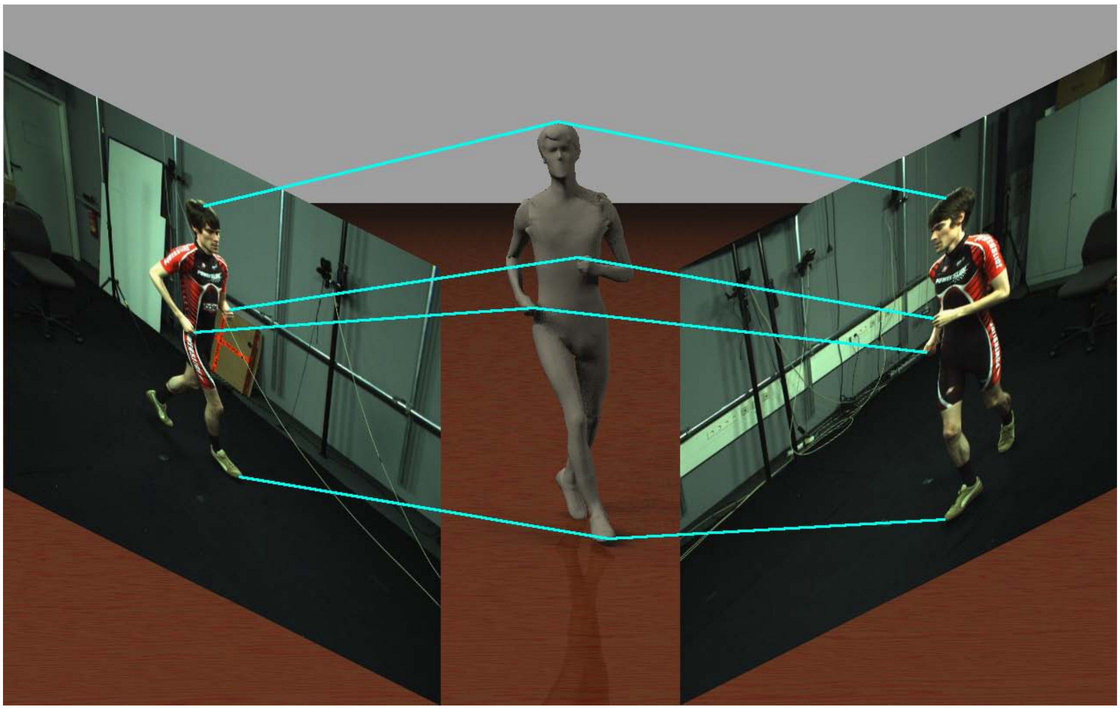


Figure 1.5: Since the cameras are calibrated, any 3D point on the surface mesh can be projected onto the image plane of each camera. The projection rays for 4 points are indicated by the cyan lines. According to our assumptions, we seek for the pose of the human, i.e. the position and orientation in the world coordinate system and the deformation of the human skeleton, such that the projections onto the image planes are consistent with the image data.

scanned humans [MCA07] or generic models like the SCAPE model [ASK$^+$05], which needs to be learned from a set of 3D scans of different people. The skeleton can be inserted manually or automatically [BP07].

Furthermore, the image sequences are assumed to be captured by 2-5 cameras that are synchronized and calibrated. For instance, the cameras are connected to a mobile system for hardware synchronization and image storage that is equipped with a rechargeable battery. For calibration, Jean-Yves Bouguet's toolbox [Bou08] and a 3D calibration rig with known geometry and LEDs have been used, see Figure 1.4 b). The camera calibration provides a common world coordinate system for all cameras. Hence, the projection from a 3D point in the world coordinate system to the 2D image plane of each camera is known as illustrated in Figure 1.5.

These two assumptions are common for model-based human motion capture from multi-view video sequences. While the surface model can be approximated and acquired by the mentioned techniques, the calibration of the cameras can be performed with minimal effort before capturing the sequence. In general, we will not impose strong restrictions neither on the environment nor on the movements. We do *not* assume that

- the motion pattern is known a-priori or part of a special subset of motions,
- the camera views are redundant,
- the scene has been captured in a controlled studio environment with static background.

1.7 Overview

The work is structured as follows. While Chapter 2 discusses related work, a brief introduction to underlying mathematical techniques is given in Chapter 3. Chapter 4 mainly discusses the modeling problem of filtering approaches in the context of model-based human motion capture. Furthermore, a prior on physical restrictions on the kinematic chain is proposed to constrain the state space and to improve weak models of the human dynamics within the framework of particle filters. The question *"What are good cues for human motion capture?"* is addressed in Chapter 5 where various image cues are discussed using a fixed local optimization scheme. Finally, an analysis-by-synthesis framework is introduced that combines complementary cues to track a variety of objects. The potential of the framework is demonstrated on various sequences and on a challenging real-world problem, namely crash test video analysis. In Chapter 6, a global optimization method is introduced that overcomes the dilemma of local optima and that is suitable for the optimization problems as they arise in human motion capturing. Besides a discussion of the asymptotic behavior, an exhaustive parameter evaluation is provided. Furthermore, the optimization technique is used to solve the pose initialization and pose tracking problem where a comparison with several other optimization and particle filtering approaches is given. The chapter concludes by addressing the question *"Is human motion capture a filtering or an optimization problem?"*. Chapter 7 focuses on high-performance tracking systems for human motion capture that combine the techniques from the previous chapters and outperform the current state-of-the-art. Finally, Chapter 8 concludes with an outlook on future research.

2

Related Work

Markerless human motion capture has been studied since more than 25 years and is still a very active research area in computer vision. The surveys [MG01a, MHK06] count nearly 500 publications in this field. In order to provide a structured overview of related work, the chapter is divided into several sections which are not disjoint. In Section 2.1, various 3D model representations for motion capture are briefly discussed. While Section 2.2 gives an overview of image-cues that have been proposed in the literature, Section 2.3 divides the approaches into optimization and filtering strategies. Section 2.4 covers various learning approaches that take additional prior knowledge into account.

2.1 Model Representation

One of the first human models for motion estimation has been proposed by O'Rourke and Badler [OB80]. The model consists of about 600 overlapping spheres and has been used for synthesizing images and estimating the human pose on the synthesized images. Other representations for human modeling are cylinders [Hog83], stick figures [LC85], polygonal meshes [YK91], patches [KMI93], truncated cones [GBUP95], superquadrics [GD96], boxes [MDN97], ellipsoids [BM98], and scaled prismatics [CR99]. A human modeled by superquadrics is shown in Figure 1.2 c). Recently, more realistic models have been proposed like implicit surfaces based on metaballs [PF03]. Other approaches rely on high resolution polygon meshes [CTMS03, SH03, MTHC03] that are fitted to accurate silhouettes from several camera views. The availability of 3D scans of humans has resulted in example-based models [ACP02, ASK^{+}05] where the model deformations are derived from a finite set of scans. Such a model has been applied to human motion capture in [BSB^{+}07].

While these approaches assume that the limbs are connected, graphical models or pictorial structures [FH00, IF01, SBR^{+}04] model the limbs as weakly connected rigid body parts by penalizing large gaps between the limbs. They have been used for bottom-up approaches where possible positions of body parts are detected independently. The final human pose is then estimated by assembling the limbs. Even though the bottom-up approaches with graphical models are particularly useful for initialization, the relaxation of the skeleton constraints allows unrealistic deformations like a varying length of a limb over time.

2.2 Image-based Cues

Typical cues for motion capture are edges, silhouettes, color, motion, and texture.

2.2.1 Edges

The classic approach to pose estimation is by means of an edge detector applied to the images. Given a model of the object surface, its silhouette or edges can be matched to the detected edges, seeking to maximize the consistency of both [Low87, Low91, RK94, DDDS03]. One of the first approaches in human motion capture, which has been proposed by Hogg [Hog83], relies on edge information. In order to make the matching feasible, the search space is reduced by a bounding box that is detected by background subtraction. Gavrila and Davis [GD96] have used chamfer matching to compare image and model edges. They simplify the task by assuming tight clothes where the body parts have different colors. Though plausible and fast, the main drawback of edge-based approaches is the numerous local minima. They are caused by many spurious edges due to noise, background clutter, or texture on the object itself.

2.2.2 Silhouettes

Region-based approaches that rely on silhouettes follow a similar concept as the edge-based approaches. Here the overlap error of the projected surface with the object region in the image is sought to be minimized. In [KMI93], the comparison is performed by an XOR operation. They assume that the 3D position of the root is given and estimate only the rotations. For matching, the search space is discretized and an exhaustive search is conducted for each limb one by one. Kakadiaris and Metaxas [KM96] establish correspondences between the projected contour and the silhouette contour to track an arm. Other approaches estimate the human pose from the visual hull or voxel data which is obtained from the silhouettes [CKBH00, MTHC01]. Bălan et al. [BBHS07] consider not only silhouettes for human pose estimation, but also cast shadows to gain some additional information in monocular sequences.

In a controlled environment, the silhouettes can be efficiently extracted by background subtraction. More general methods rely on different intensity distributions in the foreground and background region and take the object model as a shape constraint into account. This leads to a fusion of pose estimation and silhouette extraction, which is performed by level-set segmentation [BRW05, RBS$^+$06, RBW07] or graph-cut segmentation [BKT06]. The computational costs for these methods are higher than with edge-based approaches. On the other hand segmentation can better deal with low contrast edges and noise. Although there are usually fewer local optima than in the edge-based approach, local optima are still a significant problem, as they prohibit tracking in case of large transformations from frame to frame. Another problem is ambiguous solutions. For instance, the pose of a sphere cannot be uniquely determined from its silhouette.

2.2.3 Appearance

Instead of taking only the contour or edges into account, the appearance of the surface can be modeled in a more general manner. To this end, the appearance like color needs to be acquired either in a pre-processing step or is initialized at the first frame and optionally updated during tracking. Having an appearance model for the surface, one seeks for the pose such that the appearance of the surface is consistent with the image data measured at the projected surface. Wachter and Nagel [WN99] have extended an edge-based approach with an appearance model that contains the gray values of some surface points. They assume that the gray values of projected surface points remain constant in consecutive frames and optimize the pose such that the gray values measured at the projected surface points resemble their values in the appearance model. Wren et al. [WADP97] cluster pixels into regions with similar image properties such as color and spatial similarity. Particularly in the context of 2D tracking, various appearance models have been proposed like color histograms [Bir98, CRM00], view-based subspace models of appearance [BJ98, GB98], which are learned from training data, or more sophisticated methods that combine a slowly adapting template, a fast adapting template, and an outlier process [JFEM03]. The latter has also been adapted to human motion capture where the appearance is modeled by a mixture of Gaussians [BB06].

In contrast to 2D appearance models, texture-based approaches map the texture onto the 3D surface and synthesize images by projecting the textured model onto the image plane. Li et al. [LRF93] have proposed such an approach for head tracking. They use optical flow to match the synthesized image with the original image and iterate the processes of synthesizing and matching until it converges. Lerasle et al. [LRD99] have generated a fully textured model of a leg from several camera views where they assume that the subject wears richly textured tights. For comparison, the normalized cross-correlation is used. In order to handle illumination differences between synthesized images and original images, illumination templates have been proposed for head tracking [CSA00]. Leptit et al. [LPF04] have formulated 3D tracking of rigid objects as a detection problem. They store patches of a textured model from different viewpoints in a preprocessing step and match each frame to one of the key frames. Although there is a real-time implementation that uses randomized trees [LLF05], it is not suitable for articulated objects since the large number of degrees of freedom requires a large number of keyframes.

Color and texture provide more information than geometric features like silhouettes and edges, but they need to be acquired a priori or online from the video sequence. In addition, the appearance model needs to be updated during tracking in order to deal with changes of the illumination, for instance. Updating, however, is problematic since it is usually sensitive to pose estimation errors and occlusions. Another disadvantage of appearance models over silhouettes is the more complex matching which relies either on homogeneous surfaces using local or global statistics, like histograms, or structured surfaces using optical flow or patch-based matching. The latter two matching methods are also commonly used for motion cues.

2.2.4 Motion

Since the movement in the 3D space involves image motion between two consecutive frames, it is a convenient cue for pose estimation. In general, 2D correspondences between successive frames are established by flow-based or patch-based techniques. Under the assumption that the pose in the previous frame is well estimated, the 2D correspondences on the projected surface indicate the 2D movement of the subject and are used to estimate the pose for the current frame.

Optical flow methods assume brightness constancy between pairs of adjacent frames [HS81, LK81]. The success of these approaches depends considerably on the chosen optical flow method. Most methods are restricted to small pixel displacements and rely on parametric flow models that might be too restrictive, for instance, in case of human motion estimation. Moreover, optical flow estimation is usually very sensitive even to small brightness changes. These problems are better handled by current variational methods like [BBPW04], which can be implemented in real-time using a fast multi-grid solver [BW05]. Optical flow has been used for human motion capture in [PH91, JBY96, BM98].

Patch-based methods locate interest points or regions in the image that are invariant under certain transformations [MTS$^+$05]. The so-called keypoints are encoded by local descriptors [MS03], which are distinctive representations of the keypoints' neighborhoods. Correspondences between two images are then established by matching the descriptors of the keypoints. Among the most popular patch-based approaches are the KLT tracker [ST94] and a tracker based on the recently developed SIFT features [Low99, Low04]. Especially the SIFT tracker can deal with small frame rates and fast motion, as it is invariant with respect to scaling, image rotation, and moderate lighting changes. The features, however, might not be well distributed on the object's surface such that the pose estimation becomes inaccurate, whereas optical flow provides a dense field of correspondences.

The main drawbacks of patch-based and flow-based trackers in general are their need for sufficiently textured objects and the accumulation of errors over time, which results in a drift away from the object. The latter is caused by the assumption of knowing the correct pose in the previous frame.

2.2.5 Multi-cue

Since all these cues come along with inherent drawbacks, it makes sense to combine complementary cues. This has been suggested in [DM00], where optical flow is incorporated as a hard constraint in an edge-based method to face tracking. In this method, the optical flow dominates the tracking. In contrast, the work in [MBCM99] uses the optical flow in order to predict the pose parameters in a new frame, which serve as initialization for an edge-based method. The idea in [MBCM99] is that a multi-resolution optical flow method captures large displacements of the object and thus helps the edge-based method to hit better local optima. In addition, the importance of edges can be weighted according to the motion boundaries extracted from the optical flow [ST03]. Optical flow has also been used for constructing a 3D flow field [TCMS04] to refine the human pose after estimating the pose from silhouettes. This approach requires accurate silhouettes and a relatively large number of cameras to get a stable 3D flow field. Brox et al. [BRCS06] improve the shape prior for the segmentation by predicting the pose of a rigid object from optical flow. The final pose is then estimated iteratively by combining correspondences from region matching and optic flow. Vacchetti et al. [VLF04a] propose the combination of a patch-based tracker and an edge-based method. The latter aims at preventing the accumulation of errors of the patch-based tracker. However, they show that the edge-based method tends to degrade results, despite the close initialization by patch-based tracking, since there are still local optima in the vicinity of this initialization. A patch-based tracker for rigid objects has been combined with an appearance model consisting of keyframes [VLF04b]. Even though the keyframes prevent an error accumulation and help the tracker to recover after significant tracking errors, the keyframe matching is not suitable for articulated objects and objects with homogeneous surfaces. Instead of combining several cues manually, Sidenbladh and Black [SB03] learn a multi-cue likelihood for a Bayesian tracking framework from a large set of training data. They present tracking results for an arm where the learned likelihood combines steered filter responses corresponding to edges, ridges, and motion-compensated temporal differences. Other combinations, which have been proposed for human motion capture, are silhouettes and edges [DBR00] or silhouettes and stereo data [PF03].

2.3 Optimization and Filtering

2.3.1 Optimization

Local optimization has been widely used for 3D human motion capture, e.g. [RK94, GD96, KM96, BM98, WN99, DC01, CTMS03, BMP04, CBK05, KBG05, RBS^{+}06, MCA07, BC08, KRH08]. While Rehg and Kanade [RK94] model the hand by quaternions and apply a Levenberg-Marquardt method to the resulting nonlinear least-squares problem, Bregler and Malik [BM98] represent the kinematic chain by twists and solve the least squares problem by a Newton-Raphson method. Stochastic meta descent for local optimization has been used in [KBG05]. Gavrila and Davis [GD96] propose a search space decomposition where the pose of each limb is estimated in a hierarchical manner according to the kinematic chain. Starting with the torso and keeping the parameters of the other limbs fixed, the pose of each limb is estimated in a low-dimensional search space one after another. The local search is performed by discretization of the continuous space around the previous pose. This approach not only limits the accuracy by the discretization, but also propagates errors through the kinematic chain such that the extremities suffer from estimation errors of preceding limbs. The latter is addressed by Drummond and Cipolla [DC01]. They iteratively propagate the distributions of the motion parameters for the limbs through the kinematic chain to obtain the maximum a posteriori pose for the entire chain subject to the articulation constraints.

Local optimization methods provide very accurate results provided that the state vector is initialized near the global optimum. Since they search only for the locally best solution, they usually cannot recover from errors and require an initialization. Without additional prior information, the tracking often fails in case of fast motions and ambiguities. The optimization for pose estimation has recently been coupled with level-set segmentation [RBS$^+$06] and graph-cut segmentation [BKT06] where the estimated pose serves as shape prior for segmentation. Even though the shape prior yields better segmentation results and can be applied more generally than background subtraction, it introduces a local term for energy minimization that depends on the previous estimate. Hence, these approaches are not able to recover from errors since a wrong estimate results in a wrong shape prior and a wrong segmentation for the next frame.

To overcome the problem of local minima, fast simulated annealing [SH87] has been proposed for human motion capture [CMC$^+$06]. The time constraints for tracking, however, limit the number of iterations for each frame such that the global optimization needs to be aborted before the global optimum is reached. In general, optimization methods cannot handle ambiguities since they provide only a single value or a single hypothesis for the pose. An estimation error, e.g. caused by occlusions or noisy image data, results in a poor initialization for the next frame such that the search for the global optimum becomes very expansive in case of global optimization or impossible in case of local optimization. In contrast to optimization methods, filtering approaches represent the solution by a distribution and take noise and ambiguities into account.

2.3.2 Filtering and Smoothing

Filtering approaches estimate the unknown true state x_t from some noisy observations y_t, e.g. images. In general, the estimation is called prediction, filtering, or smoothing if observations before frame t, including t, or also after t are taken into account. The filtering problem is typically solved by Kalman filtering [Kal60] or particle filtering [GSS93] where it is assumed that the underlying stochastic processes

$$x_{t+1} = f_t\left(x_t\right) + v_t, \tag{2.1}$$

$$y_t = h_t\left(x_t\right) + w_t \tag{2.2}$$

with noise v_t and w_t are known. While f_t models the transition of the state from time step t to $t+1$, the mapping from the state space to the observation space is given by h_t. Isard and Blake [IB96] have applied a particle filter to 2D tracking and have extended it to a two-pass smoothing algorithm [IB98]. For 3D human motion capture, Sidenbladh et al. [SBF00] have combined a particle filter with very strong motion priors to resolve the ambiguities from monocular sequences. Motion priors have been also proposed for a Rao-Blackwellised particle filter [XL07]. In [WR06] various variants of particle filters like the unscented particle filter [MDFW00] have been evaluated for human motion capture. The most modifications aim to improve the distribution of the particles in the high-dimensional space to obtain a better approximation of the posterior. In [CF01] hybrid Monte Carlo filtering has been applied where a Markov chain Monte Carlo technique is used within a particle filter to get better samples from the posterior. Another approach follows the idea of search space decomposition where the space is divided into independent low-dimensional subspaces [MI00]. When human models are represented as graphical models, nonparametric belief propagation [LN06, SBR$^+$04] has been proposed, which allows inference over arbitrary graphs rather than a simple chain.

Pentland and Horowitza [PH91] combine a Kalman filter with a finite element method for tracking non-rigid and articulated objects. An extended Kalman filter has been applied by Goncalves et al. [GBUP95] to track a human arm even though the noise w_t for the observations is not Gaussian according to their measurements, which are given by comparing the brightness between the real and the predicted image at several sample points. Rohr [Roh97] has suggested a Kalman filter framework for human motion capture where the measurements y_t are obtained by a local grid search and a constant velocity is assumed for prediction. The Kalman filter not only provides a better initialization for the local optimization, but also filters the noisy pose estimates from the local search. A Kalman filter has been also integrated into a more complex framework with multiple abstraction levels of the human dynamics [Bre97]. Since a Kalman filter provides only a single hypothesis, Cham and Rehg combined several Kalman filters to track multiple hypotheses [CR99].

Even though filtering approaches exploit temporal coherence, handle noise and are able to recover from errors, they are usually too imprecise for motion analysis in high dimensional spaces. Since accurate models for f_t and h_t are rarely available, the model's weakness is compensated by overestimating the noise vectors v_t and w_t at the expense of poor performance. For this reason, some heuristics based on particle filters have been developed to combine local optimization with filtering. Sminchisescu and Triggs [ST03] propose covariance scaled sampling to guide the particles to the local maxima of a posterior distribution. To find the local maxima, the particles are broadly spread in the search space by inflating the covariance of the dynamic prior and refined by a local optimization with respect to the likelihood. The posterior is then modeled by a mixture of Gaussians where the means and covariance matrices are given by the detected local maxima and their Hessians. Smart particle filtering [BKMG07] combines a particle filter with the stochastic meta descent [Sch99] for local optimization. Since the optimization of the particles changes the approximated distribution, a correction factor is used to compensate for the additional set of particles. The factor, however, depends on the unknown posterior distribution. Hence, a regularization [DFG01, Chapter 12], which introduces an error, is performed to estimate the continuous posterior distribution from the finite set of particles before the optimization step. Particularly, the low number of particles makes an accurate estimation of the correction factor infeasible. Deutscher et al. propose an annealed particle filter [DBR00, DR05] that follows the idea of annealing to guide the particles to the global maximum of the likelihood. To this end, the shape of the likelihood is gradually changed and the sampling is repeated. The approach does not perform annealing in the classical sense where the temperature is monotonically decreased, but relies on the fluctuating survival rate of the particles. Hence, the annealed particle filter is not suitable for global optimization and requires an additional technique for initialization like the other approaches that combine local optimization with particle filter. Although it has been shown that these heuristics work well tracking hands or humans, there is no evidence that they converge to the optimal solution of the filtering problem as stated in Equations (2.1) and (2.2) in contrast to Kalman or particle filtering.

2.4 Prior Knowledge

Besides a known surface model, other prior knowledge has been proposed for human motion capture like anatomical constraints, motion, and appearance priors. In particular, the use of prior poses or motion patterns learned from a motion database has become very popular in order to achieve robust tracking also in difficult and ambiguous scenarios [SBF00, SBS02, UF04, RBS07]. By learning the mapping between the image space and the pose space, the pose can be directly recovered from silhouettes and image features [GSD03, AT06, LE07, SKM07]. In [TDDS06] pose estimation is formulated as inference in a conditional random field model where the observation potential function is learned from a large set of trainings data. Gaussian process dynamical models [MP06, UFF06] have been used for embedding motion in a low-dimensional latent space. In [LYST06] locally linear coordination is proposed for dimensionality reduction. Although these learning strategies allow for tracking even in monocular video sequences, they impose strong assumptions on the tracked motion. The restriction to a small subset of human motion patterns limits their application in practice. A prior on anatomical constraints is independent of the motion and can be learned [BRKC06], but the used training data also introduces some bias. When, for example, the movement of a person with an artificial hip joint is measured using training data from persons with natural hip joints, the estimates are likely to be biased towards the movement of a person with natural hip joints, i.e., one eliminates exactly the information that is important for the medical application.

Tracking-by-detection approaches [SVD03, LESC04, MM06, ARS08] rely on a learned template model and require a large training set. Since the detection is usually limited to canonical poses like lateral walking, the human poses are only detected on a subset of frames. A second step is therefore required to interpolate or track between the detected frames. The tracking or refinement is usually done offline since the detected poses are also used to learn a subject specific appearance model [FDLF07, RFZ07]. A more detailed description of various learning approaches can be found in the survey [Pop07].

3

Preliminaries

3.1 Model Representation

There are several ways to represent the pose of an object, e.g., Euler angles, quaternions [Gol80], twists [MLS94], or the axis-angle representation. Stochastic approaches like particle filter or ISA require from the representation that primarily the mean but also the variance can be at least well approximated. For this purpose, we have chosen the axis-angle representation of the absolute rigid body motion M given by the 6D vector $(\theta\omega, t)$ with

$$\omega = (\omega_1, \omega_2, \omega_3), \quad \|\omega\|_2 = 1 \quad \text{and} \quad t = (t_1, t_2, t_3).$$

Using the exponential, M is expressed by

$$M = \begin{pmatrix} \exp(\theta\hat{\omega}) & t \\ 0 & 1 \end{pmatrix}, \qquad \hat{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{3.1}$$

While t is the absolute position in the world coordinate system, the rotation vector $\theta\omega$ describes a rotation by an angle $\theta \in \mathbb{R}$ about the rotation axis ω. The function $\exp(\theta\hat{\omega})$ can be efficiently computed by the Rodriguez formula [MLS94].

Given a rigid body motion defined by a rotation matrix $R \in SO(3)$ and a translation vector $t \in \mathbb{R}^3$, the rotation vector is constructed according to [MLS94] as follows: When R is the identity matrix, θ is set to 0. For the other case, θ and the rotation axis ω are given by

$$\theta = \cos^{-1}\left(\frac{trace(R) - 1}{2}\right), \qquad \omega = \frac{1}{2\sin(\theta)} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}. \tag{3.2}$$

We write $\log(R)$ for the inverse mapping of the exponential.

The mean of a set of rotations r_i in the axis-angle representation can be computed by using the exponential and the logarithm as described in [PA98, Pen98]. The idea is to find a geodesic on the Riemannian manifold determined by the set of 3D rotations. When the geodesic starting from the mean rotation in the manifold is mapped by the logarithm onto the tangent space at the mean, it is a straight line starting at the origin. The tangent space is called *exponential chart*. Hence, using the notations

$$r_2 \star r_1 = \log\left(\exp(r_2) \cdot \exp(r_1)\right), \qquad r_1^{-1} = \log\left(\exp(r_1)^T\right)$$

for the rotation vectors r_1 and r_2, the mean rotation $\bar{r}$ satisfies

$$\sum_i \left(\bar{r}^{-1} \star r_i\right) = 0. \tag{3.3}$$

Weighting each rotation with $\sum_i \pi_i = 1$, yields the least squares problem:

$$\frac{1}{2} \sum_i \pi_i \left\|\bar{r}^{-1} \star r_i\right\|_2^2 \rightarrow min. \tag{3.4}$$

The weighted mean can thus be estimated by

$$\hat{r}_{t+1} = \hat{r}_t \star \left(\frac{\sum_i \pi_i \left(\hat{r}_t^{-1} \star r_i\right)}{\sum_i \pi_i} \right). \tag{3.5}$$

The gradient descent method takes about 5 iterations until it converges. The variance and the normal density on a Riemannian manifold can also be approximated, cf. [Pen06].
The twist representation used in [BM98, BMP04, RBS$^+$06] is quite similar. Instead of a separation between the translation t and the rotation r, it describes a screw motion where the motion velocity θ also affects the translation. A twist $\theta\hat{\xi} \in se(3)$ is represented by

$$\theta\hat{\xi} = \theta \begin{pmatrix} \hat{\omega} & v \\ 0 & 0 \end{pmatrix}, \tag{3.6}$$

where $\exp(\theta\hat{\xi})$ is a rigid body motion. The logarithm of a rigid body motion $M \in SE(3)$ is the following transformation:

$$\theta\omega = \log(R), \qquad v = A^{-1}t, \tag{3.7}$$

where

$$A = (I - \exp(\theta\hat{\omega}))\hat{\omega} + \omega\omega^T\theta \tag{3.8}$$

is obtained from the Rodriguez formula. This follows from the fact, that the two matrices which comprise A have mutually orthogonal null spaces when $\theta \neq 0$. Hence, $Av = 0 \Leftrightarrow v = 0$.
Since the position of a joint is constrained by the skeleton, a joint j can be modeled as a rotation around a given axis, i.e., the joint motion depends only on the rotation angle θ_j. We remark that the twist and angle-axis representation are identical in this case. Hence, we write M_{RBM} for the rigid body motion and $M(\theta_j)$ for the joints. Furthermore, we have to consider the kinematic chain of the skeleton. Let X_i be a point on the limb k_i whose position is influenced by n_{k_i} joints in a certain order. The inverse order of these joints is then given by the mapping ι_{k_i}; for example, a point on the left shank is influenced by the left knee joint $\iota_{k_i}(4)$ and by the three joints of the left hip $\iota_{k_i}(3)$, $\iota_{k_i}(2)$, and $\iota_{k_i}(1)$. Using homogeneous coordinates, the transformation of X_i is given by

$$X_i' = M_{RBM} M(\theta_{\iota_{k_i}(1)}) \ldots M(\theta_{\iota_{k_i}(n_{k_i})}) X_i. \tag{3.9}$$

Since the body parts of humans are not rigid, a skeletal subspace deformation (SSD) can be performed to interpolate around the joints. Instead of associating each vertex X_i of the mesh with only one bone and thus obtaining one transformation $X_i' = T_{k_i} X_i$, SSD [MTLT88, LCF00] linearly interpolates between the vertex transformations with respect to several bones. The influence of a bone k on a vertex X_i is given by $w_{k,i}$ where $\sum_k w_{k,i} = 1$. Equation (3.9) then becomes

$$X_i' = \sum_k w_{k,i} T_k X_i. \tag{3.10}$$

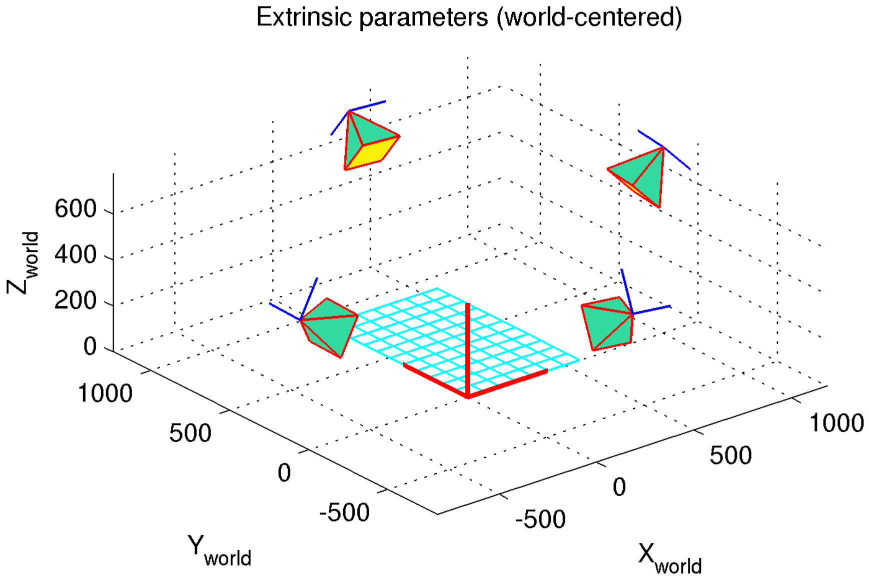


Figure 3.1: The positions and orientations of the cameras with respect to the world coordinate system are specified by the extrinsic parameters.

Even though SSD is very fast, it has several well-known drawbacks. When the joints are rotated to extreme angles, a loss of volume can be observed since the linear interpolation of the transformation matrices is not equivalent to the linear interpolation of their rotations. Furthermore, the variety of shapes is limited to the linear subspace of transformations such that subject specific deformations like bulging of muscles cannot be modeled. In order to achieve more realistic character animations, a variety of methods like [LCF00, ACP02, ASK$^+$05, MMG06, KCvO07, WSLG07, YBS07] have been proposed. For human motion capture, however, the fast computation time of SSD usually outweights the artifacts which affect only very few pixels. When the motion capture data is used for animation, a more sophisticated deformation scheme in combination with a high resolution mesh can be applied after pose estimation to achieve the necessary visual quality.

3.2 Camera Calibration

In order to fuse the information form several camera views, a common world coordinate system is required. This is obtained by calibrating each camera using a 2D or 3D rig as shown in Figure 1.4 b). For calibration, we use Jean-Yves Bouguet's camera calibration toolbox [Bou08] that determines the intrinsic and extrinsic parameters for each camera. While the intrinsic parameters describe the camera model, the extrinsic parameters determine the position and orientation of a camera according to a common world coordinate system. Detailed descriptions of various camera models and calibration methods can be found, for example, in [Bro71, Tsa87, HS97, SM99, Zha99]. When the radial and tangential distortions are removed, the projection matrix P can be written as

$$P = K\begin{pmatrix} R & t \end{pmatrix}, \quad \text{where} \quad K = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.11}$$

and the extrinsic parameters are given by the rotation matrix R and the translation vector t. The parameters (u_0, v_0) are the coordinates of the principal point, α and β are scale factors in image axes, and γ describes the skewness of the two image axes. Knowing P and using homogeneous coordinates, the projection from a 3D point X in the world coordinate system onto the image plane is given by $x = \Pi(P\ X)$ where Π denotes the projection from homogeneous to non-homogeneous coordinates. The world coordinate system and camera projections are illustrated in Figures 1.5 and 3.1.

3.3 Level-Set Segmentation

Assuming that only one object is to be segmented, level-set segmentation splits the image domain Ω into two disjoint regions, namely foreground Ω_1 and background Ω_2. The probability that a pixel x belongs to the region Ω_1 or Ω_2 is modeled by the probability density functions p_1 and p_2. The contour is given by the zero-line of a level-set function $\Phi : \Omega \to \mathbb{R}$, where $\Phi(x) > 0$ if $x \in \Omega_1$ and $\Phi(x) < 0$ otherwise. We generally constrain Φ to be the signed distance image of the contour. This means the absolute value of $\Phi(x)$ is the minimum distance of x to the contour. The segmentation problem can be formulated as an energy minimization problem [MS89, CV01, PD02, BRDW03]:

$$E(\Phi, p_1, p_2) = -\int_\Omega H(\Phi)\ln p_1 + (1 - H(\Phi))\ln p_2 \, dx + \vartheta \int_\Omega |\nabla H(\Phi)| \, dx, \tag{3.12}$$

where H is a regularized version of the Heaviside step function with $\lim_{s\to-\infty} H(s) = 0$, $\lim_{s\to\infty} H(s) = 1$, and $H(0) = 0.5$. While the first term maximizes the likelihood, the second term, weighted by the fixed parameter ϑ, regulates the smoothness of the contour. The minimization of Equation (3.12) with respect to Φ and p_i is achieved by gradient descent. Having an initial level-set function Φ, the densities p_i can be estimated. At the same time step, Φ is updated by

$$\Phi^{k+1} = \Phi^k + H'(\Phi^k)\left(\ln\frac{p_1^k}{p_2^k} + \vartheta \mathrm{div}\left(\frac{\nabla\Phi^k}{|\nabla\Phi^k|}\right)\right) \tag{3.13}$$

with iteration index k. The steps are iterated until the segmentation process converges to a local minimum.

There are various ways to model the probability densities p_1 and p_2. The most simple choice is the approximation of each region by its mean [CV01]. However, this would restrict the segmentation to homogeneous objects with homogeneous background. Other choices are Gaussians, mixture of Gaussians [PD02], or nonparametric Parzen density estimates [KFY$^+$05]. In order to keep the region model manageable, the color or feature channels u_j are assumed to be uncorrelated. Hence, the joint probability density function for a region Ω_i can be written as

$$p_i(x) = \prod_j p_{ij}(u_j(x)). \tag{3.14}$$

Using the notation $H_1 = H(\Phi)$ and $H_2 = 1 - H(\Phi)$, the Gaussian distribution

$$p_{ij}(u_j(x)) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp\left(\frac{(u_j(x) - \mu_{ij})^2}{2\sigma_{ij}^2}\right). \tag{3.15}$$

is given by the mean and variance

$$\mu_{ij} = \frac{\int_\Omega u_j(\zeta) H_i(\zeta)\, d\zeta}{\int_\Omega H_i(\zeta)\, d\zeta} \tag{3.16}$$

$$\sigma_{ij}^2 = \frac{\int_\Omega (u_j(\zeta) - \mu_{ij})^2 H_i(\zeta)\, d\zeta}{\int_\Omega H_i(\zeta)\, d\zeta}. \tag{3.17}$$

The nonparametric Parzen density estimate is computed by smoothing the histograms for each region i and channel j by a Gausian Kernel K_ρ with standard deviation ρ:

$$p_{ij}(u_j(x)) = K_\rho * \frac{\int_\Omega \delta_{u_j(\zeta)}(u_j(x)) H_i(\zeta)\, d\zeta}{\int_\Omega H_i(\zeta)\, d\zeta}, \tag{3.18}$$

with $\delta_x(y) = 1$ if $x = y$ and 0 otherwise. In comparison to the Gaussian approximation, the nonparametric probability density functions describe the region statistics better but they adapt more specifically to the given data which results in more local minima in the energy function (3.12). In contrast to conventional distributions that assume homogeneous regions, local distributions relax this assumption to be satisfied only locally [BRW05]. At each spatial position x, a separate probability density function is estimated from its local neighborhood. In the case of local Gaussian distributions, the regions are modelled by

$$p_{ij}(u_j(x), x) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2(x)}} \exp\left(\frac{(u_j(x) - \mu_{ij}(x))^2}{2\sigma_{ij}^2(x)}\right). \tag{3.19}$$

Estimation of the parameters $\mu_{ij}(x)$ and $\sigma_{ij}(x)$ can be achieved using a window function, e.g. a Gaussian K_ρ with standard deviation ρ, and restricting the estimation only to points within this window:

$$\mu_{ij}(x) = \frac{\int_\Omega K_\rho(\zeta - x) u_j(\zeta) H_i(\zeta)\, d\zeta}{\int_\Omega K_\rho(\zeta - x) H_i(\zeta)\, d\zeta} \tag{3.20}$$

$$\sigma_{ij}^2(x) = \frac{\int_\Omega K_\rho(\zeta - x)(u_j(\zeta) - \mu_{ij}(x))^2 H_i(\zeta)\, d\zeta}{\int_\Omega K_\rho(\zeta - x) H_i(\zeta)\, d\zeta}. \tag{3.21}$$

The window function implies spatial smoothness of the densities. The amount of smoothness is steered by the parameter ρ. Obviously, the local model converges to the corresponding homogeneous distribution model for $\rho \rightarrow \infty$.

In general, a bunch of channels u_j can be used to obtain discriminative region models. Typical channels are intensity, color, and texture information like Gabor filters [Gab46] or texture features [BW06a]. Since the number of channels and the complexity of the statistical model affect the computation time, the optimal model for segmenting an entire sequence is the simplest one that still has the ability to distinguish between fore- and background. Hence, the optimal model depends on the image data when the computation time is considered. For a more comprehensive survey on region-based segmentation schemes, we refer to [CRD07].

3.4 Particle Filter

Particle filters are designed to solve a filtering problem outlined by Equations (2.1) and (2.2). In this section, we summarize the fundamentals of particle filters and discuss convergence results under various assumptions as well as their impact on applications, particularly on motion capturing. Before a basic particle filter is presented as solution to the filtering problem described in [DFG01, Ch. 2] and [CG99], some basic notations are introduced.

3.4.1 Notations

Let (E, τ) be a topological space, and let $\mathcal{B}(E)$ denote its Borel σ-algebra. $B(E)$, $C_b(E)$, and $\mathcal{P}(E)$ denote the set of bounded measurable functions, bounded continuous functions, and probability measures, respectively. δ_x is the Dirac measure concentrated in $x \in E$, i.e. $\delta_x(\{x\}) = 1$ and $\delta_x(\complement\{x\}) = 0$. A Markov kernel K is a function $K : E \times \mathcal{B}(E) \to [0, \infty]$ such that $K(\cdot, B)$ is $\mathcal{B}(E)$-measurable $\forall B$ and $K(x, \cdot)$ is a probability measure $\forall x$. An example of a Markov kernel is given in Equation (3.30). For a thorough introduction to measure and probability theory, we refer to [Bau90, Bau91, Bau96, Bil95]. The supremum norm is denoted by $\|\cdot\|_\infty$.
Throughout this paper, we use the following compact notation for integrals:

$$\langle \mu, f\rangle = \int_E f(x)\, \mu(dx), \tag{3.22}$$

$$\langle K, f\rangle(x) = \int_E f(y)\, K(x, dy) \quad \text{for} \quad x \in E, \tag{3.23}$$

$$\langle \mu, K\rangle(B) = \int_E K(x, B)\, \mu(dx) \quad \text{for} \quad B \in \mathcal{B}(E), \tag{3.24}$$

where $f \in B(E)$, $\mu \in \mathcal{P}(E)$, and K is a Markov kernel on E.

3.4.2 Filtering

Let $X = (X_t)_{t\in\mathbb{N}_0}$ be an $\mathbb{R}^d$-valued Markov process, called *signal process*, with initial distribution η_0 and a family of transition kernels $(K_t)_{t\in\mathbb{N}_0}$ satisfying the Feller property [RW01], i.e. $\langle K_t, f\rangle \in C_b(E)$ for all t and $f \in C_b(E)$. Let $Y = (Y_t)_{t\in\mathbb{N}_0}$ be an $\mathbb{R}^m$-valued stochastic process, called *observation process*, defined as

$$Y_t = h_t(X_t) + W_t \qquad \text{for } t > 0, \qquad Y_0 = 0, \tag{3.25}$$

where, for each $t \in \mathbb{N}$, $h_t : \mathbb{R}^d \to \mathbb{R}^m$ is a continuous function, $(W_t,\ t \in \mathbb{N})$ are independent m-dimensional random vectors and their distributions possess densities $g_t \in C_b(\mathbb{R}^m)$. The filtering problem consists in computing the conditional distribution

$$\eta_t(B) := P\left(X_t \in B \mid Y_t, \ldots, Y_0\right), \tag{3.26}$$

for all $B \in \mathcal{B}(\mathbb{R}^d)$ or, alternatively, $\langle \eta_t, \varphi\rangle = E\left[\varphi(X_t) \mid Y_t, \ldots, Y_0\right]$ for all $\varphi \in B(\mathbb{R}^d)$.
The *generic particle filter (GPF)* is a commonly used particle filter for the solution of the filtering problem, which provides a basis for further developments and modifications for other applications. The algorithm consists of the four steps “Initialization”, “Prediction”, “Updating”, and “Resampling”. During the initialization, we sample n times from the initial distribution η_0. By saying that we sample $x^{(i)}$ from a distribution μ, we mean that we simulate n independent

random samples, also named particles, according to μ. Hence, the n random variables $(X_0^{(i)})$ are independent and identically distributed (i.i.d.) according to η_0. Afterwards, the values of the particles are predicted for the next time step according to the dynamics of the signal process. During the "Updating" step, each predicted particle is weighted by the likelihood function $g_t(y_t - h_t(\cdot))$ which depends on the observation y_t. The resampling is done by drawing n times with replacement from the set $(\bar{x}_{t+1}^{(j)})_{j=1\ldots n}$ according to the probabilities $\pi_{t+1}^{(j)}$.

Algorithm 1 Generic Particle Filter

```
Requires: number of particles $n$, $\eta_0$, $(K_t)_{t\in\mathbb{N}_0}$, $(g_t)_{t\in\mathbb{N}}$, $(h_t)_{t\in\mathbb{N}}$, and observations $(y_t)_{t\in\mathbb{N}}$
  1. Initialization
       • Sample $x_0^{(i)}$ from $\eta_0$ for all $i$
  2. Prediction
       • Sample $\bar{x}_{t+1}^{(i)}$ from $K_t(x_t^{(i)}, \cdot)$ for all $i$
  3. Updating
       • Set $\pi_{t+1}^{(i)} \leftarrow g_{t+1}(y_{t+1} - h_{t+1}(\bar{x}_{t+1}^{(i)}))$ for all $i$
  4. Resampling
       • Set $x_{t+1}^{(i)} \leftarrow \bar{x}_{t+1}^{(j)}$ with probability $\frac{\pi^{(j)}}{\sum_{k=1}^{n} \pi^{(k)}}$ for all $i$ and go to step 2
```

Also other "Resampling" steps than the one described in Algorithm 1 have been employed for the particle filter, e.g. branching procedures [CG99, CML99, DFG01]. A detailed discussion can be found in [Mor04, Ch. 11.8]. The particle system is also called *interacting particle system* [Mor98] since the particles are (obviously) not independent after resampling.

For the case of a one-dimensional signal process, the operation of the algorithm is illustrated in Fig. 3.2, where the grey circles represent the unweighted particles after the "Prediction" step and the black circles represent the weighted particles after the "Updating" step. While the horizontal positions of the particles indicate their values in the state space of the signal process, the diameters of the black circles indicate the particle weights, that is the larger the diameter the greater the weight. As illustrated, the particles with large weight generate more offsprings than particles with lower weight during the "Resampling" step. In order to discuss the mathematical properties of the algorithm, we use the following notions (cf. also [Mac00]).

Definition 3.4.1. A *weighted particle* is a pair (x, π) where $x \in \mathbb{R}^d$ and $\pi \in [0, 1]$. A *weighted particle set* S is a sequence of finite sets of random variables whose values are weighted particles: the nth member of the sequence is a set of n random variables $S^{(n)} = \{(X^{(1)}, \Pi^{(1)}), \ldots, (X^{(n)}, \Pi^{(n)})\}$, where $\sum_{i=1}^{n} \Pi_i^{(n)} = 1$.

It is clear that every weighted particle set determines a sequence of random probability measures by

$$\sum_{i=1}^{n} \Pi^{(i)} \delta_{X^{(i)}} \qquad \text{for } n \in \mathbb{N}.$$

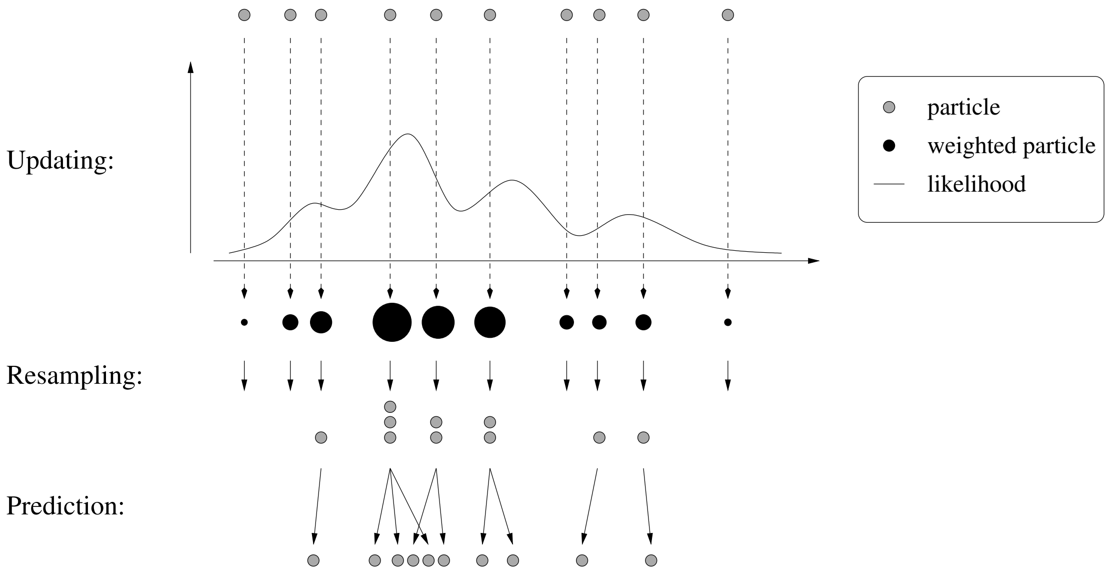


Figure 3.2: Operation of the generic particle filter. The predicted particles (*gray circles*) are weighted by the likelihood. The weighted particles (*black circles*) are resampled and predicted for the next time step.

The idea now is to approximate the conditional distribution η_t (3.26) by the distribution of an appropriate weighted particle set. We note that each step of the generic particle filter defines a particle set and consequently a random probability measure:

$$\hat{\eta}_t^n := \frac{1}{n}\sum_{i=1}^n \delta_{\bar{X}_t^{(i)}}; \quad \bar{\eta}_t^n := \sum_{i=1}^n \Pi_t^{(i)} \delta_{\bar{X}_t^{(i)}}; \quad \eta_t^n := \frac{1}{n}\sum_{i=1}^n \delta_{X_t^{(i)}}.$$

With this notation, the algorithm is illustrated by the three separate steps

$$\eta_t^n \xrightarrow{Prediction} \hat{\eta}_{t+1}^n \xrightarrow{Updating} \bar{\eta}_{t+1}^n \xrightarrow{Resampling} \eta_{t+1}^n. \tag{3.27}$$

3.4.3 Convergence

The proof of the following convergence result can be found in [Mor96].

Theorem 3.4.2. *For all* $t \in \mathbb{N}_0$, *there exists* c_t *independent of* n *such that*

$$E\left[\left(\langle \eta_t^n, \varphi\rangle - \langle \eta_t, \varphi\rangle\right)^2\right] \leq c_t \frac{\|\varphi\|_\infty^2}{n} \quad \forall \varphi \in B(\mathbb{R}^d). \tag{3.28}$$

Inequality (3.28) shows that the rate of convergence of the mean square error is of order $1/n$. However, c_t depends on t and, without any additional assumption, c_t actually increases over time. This is not very satisfactory in applications as this implies that one needs an increasingly larger number of particles as time t increases to ensure a given precision. We will state below a recent convergence result (Theorem 3.4.6) which is uniform in time under additional assumptions on the filtering problem. The idea of preventing an increasing error is to ensure that any error is forgotten fast enough. For this purpose, we define a so-called mixing condition in accordance with [MG01b] and [GO04].

Definition 3.4.3. A kernel on E is called *mixing* if there exists a constant $0 < \varepsilon \leq 1$ and a measure μ on E such that

$$\varepsilon\mu(B) \leq K(x,B) \leq \frac{1}{\varepsilon}\,\mu(B) \quad \forall x \in E, B \in \mathcal{B}(E). \tag{3.29}$$

This strong assumption means that the measure $K(x,\cdot)$ depends only "weakly" on x. It can typically only be established when $E \subset \mathbb{R}^d$ is a bounded subset which is the case in many applications like human motion capturing. For example, the (bounded) Gaussian distribution on E

$$K(x,B) := \frac{1}{Z}\int_B \exp\left(-\frac{1}{2}\,(x-y)^T\,\Sigma^{-1}\,(x-y)\right)\,dy \tag{3.30}$$

with $Z := \int_E \exp(-\frac{1}{2}\,(x-y)^T\,\Sigma^{-1}\,(x-y)\,dy$ is mixing if and only if E is bounded. Moreover, a Gaussian with a high variance satisfies the mixing condition with a larger ε than a Gaussian with lower variance. We give two examples where the kernels are not mixing.

Example 3.4.4. Let $E = \{a,b\}$ and $K(x,B) := \delta_x(B)$. Assume that K is mixing. From inequality (3.29) we get the following contradiction

$$\begin{aligned} K(a,\{b\}) = \delta_a(\{b\}) = 0 \quad &\Rightarrow \quad \mu(\{b\}) = 0, \\ K(b,\{b\}) = \delta_b(\{b\}) = 1 \quad &\Rightarrow \quad \mu(\{b\}) > 0. \end{aligned}$$

Example 3.4.5. Let $E = \mathbb{R}$ and

$$K(x,B) := \frac{1}{\sqrt{2\pi}}\int_B \exp\left(\frac{-(x-y)^2}{2}\right)\,dy.$$

Suppose there exists an $\varepsilon > 0$ and a measure μ such that the inequality (3.29) is satisfied. Note that for all $x \in \mathbb{R}$ and all intervals $I = [a,b]$, $a < b$, we have $K(x,I) > 0$. Our assumption entails that $\mu(I) > 0$. But then $\varepsilon\mu(I) < K(x,I)$ cannot hold for all $x \in \mathbb{R}$, since $K(x,I) \rightarrow 0$ as $|x| \rightarrow +\infty$.

The uniform convergence of the generic particle filter with respect to the time parameter was first proved by Del Moral and Miclo [MM00] assuming that the mixing condition for $(K_t)_{t\in\mathbb{N}_0}$ is satisfied. Le Gland and Oudjane [GO04] showed also the uniform convergence (Theorem 3.4.6) by using the mixing condition for the family of random kernels

$$R_t(x,B) := \int_B g_{t+1}(Y_{t+1} - h_{t+1}(y))\,K_t(x,dy).$$

Theorem 3.4.6. *If the family of random kernels $(R_t)_{t\in\mathbb{N}_0}$ is mixing with $\varepsilon_t \geq \varepsilon > 0$, then there exists a constant $c(\varepsilon)$ independent of n such that*

$$E\left[\left(\langle\eta_t^n,\varphi\rangle - \langle\eta_t,\varphi\rangle\right)^2\right] \leq c(\varepsilon)\,\frac{\|\varphi\|_\infty^2}{n} \quad \forall t \in \mathbb{N}_0,\, \varphi \in B(\mathbb{R}^d).$$

This means that as long as the mixing condition (3.29) is satisfied there exists an upper bound of the error that is independent of the time parameter. Hence, the number of particles, that ensures a given precision in an application, does not increase over time. The mixing condition can furthermore be relaxed such that the density $dK(x,\cdot)/d\mu$ is not μ-almost surely greater than or equal to $\varepsilon > 0$ but may vanish on a part of the state space, as shown in [CL04].

4

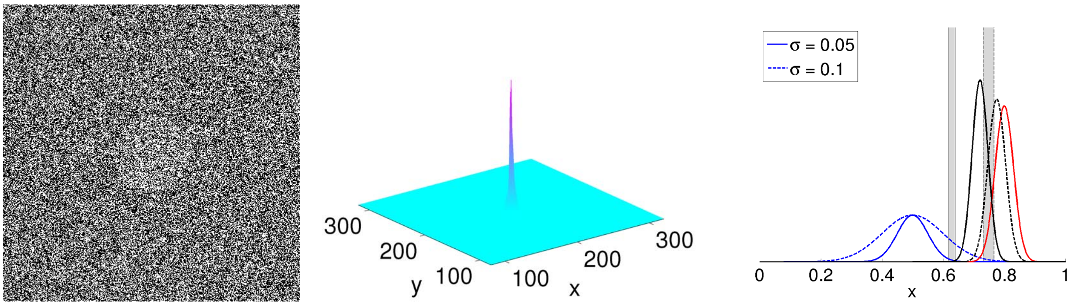
Filtering

This chapter mainly discusses the modeling problem of filtering approaches in the context of model-based human motion capture. Furthermore, a prior on physical restrictions on the kinematic chain is proposed to constrain the state space and to improve weak models of the human dynamics within the framework of particle filters.

4.1 Modeling Problem

It is important to note that the convergence results stated in Section 3.4.3 are only valid when the signal and the observation process are known and satisfy the respective assumptions. Since this is rarely the case for applications, good approximations are needed. In applications like motion capture, it is very difficult to model the transition kernels and the noise of the observation process in an appropriate way. The modeling problem can be addressed by learning techniques [SBS02, SB03], but even for large datasets the models are usually not general enough such that they cover the large space of human motion and human appearance. For applications like motion analysis, the bias introduced by the training data, particularly in terms of motion priors, is actually a negative side effect.

In order to illustrate the modeling problem for model-based human motion capture, we continue the synthetic example from Section 1.4. The example uses a simple model, namely the disc, and the observation y_t is a silhouette image distorted by Gaussian pixel noise as shown in Figure 4.1 a). Hence, the observation process (3.25) is known: $h_t : \mathbb{R}^2 \rightarrow \mathbb{R}^{400^2}$ is the projection of the disc and g_t is a multivariate Gaussian probability density function. The weighting function $g_t(y_t - h_t(\cdot))$ is plotted in Figure 4.1 b). Despite of the severe image distortion, the likelihood is similar to Figure 1.3 c). It indicates that the impact of observation noise is very small when silhouettes and surface models are used as it is common for model-based human motion capture. Without strong motion priors, modeling the human dynamics by Markov kernels K_t, which rely only on the previous state, gives just a rough approximation of the real dynamical model. To overcome this problem, the state vector x_t can be extended by taking velocity and acceleration into account under the assumption that the framerate is known. While this works fine for low dimensional tracking problems, it requires estimating a nearly 90-dimensional signal for human motion capture. When the dynamics cannot be well approximated and the impact of the observation noise is assumed to be small, the generic particle filter is expected to perform poorly. The weak prediction causes most of the particles to be far away from the observation and to have marginal weights. This yields a poor approximation of the mean of η_t as it is illustrated in Figure 4.1 c), where 100 particles have been used. An overestimation of the signal noise improves the estimate but it also changes η_t. The problem of a very peaked likelihood is also addressed by modifications of the generic particle filter like extended Kalman particle filter [FNGD00],



(a) (b) (c)

Figure 4.1: Impact of the noise model. **From left to right: a)** The synthetic example from Figure 1.3 is continued. The image y_t is severely distorted by independent Gaussian pixel noise w_t. **b)** The correct likelihood function $g_t(y_t - h_t(\cdot))$ is not anymore the Dirac measure but still very spiky despite the image noise. Such weighting functions are problematic for a particle filter since the vast majority of the particles have marginal weights such that η_t is basically estimated by very few particles yielding a poor approximation. **c)** One dimensional example. When the noise of the observation process $g_t(y_t - h_t(\cdot))$ *(red)* is small and the prediction model in terms of K_t *(blue)* is weak, the mean of the approximated distribution η_t *(black)* is poorly estimated *(gray bar)*. In order to get more particles near to the observation and a better estimate of the mean, the noise of the signal process can be overestimated *(dashed)*. Note that η_t also becomes dominated by $g_t(y_t - h_t(\cdot))$.

unscented particle filter [MDFW00], auxiliary particle filter [PS99], and the regularized particle filter [DFG01, Chapter 12]. However, the overestimation of the signal noise also reflects the unreliability of a weak dynamical model such that the posterior is dominated by the likelihood.

4.2 Learning Constraints of the Skeleton

In contrast to dynamical priors or dimensionality reduction techniques, a prior on anatomical constraints is independent of motion patterns and can also be learned from training data [BRKC06] as discussed in Section 2.4. The prior constrains the state space such that anatomical limits of the joints and unrealistic self-intersections are considered for the transition kernels K_t. Instead of modeling the physical restrictions as hard constraints, we allow for the simplification and approximation of the kinematic model by integrating this prior knowledge as soft constraints. To this end, a probability density function p_{pose} models the probability of a skeleton configuration in the space of human poses and is combined with a common dynamical model K_t^{pred}. The modified transition kernel is given by

$$K_t(x_t, B) = \int_B \frac{1}{Z(x_t)} p_{pose}(x_{t+1}) K_t^{pred}(x_t, dx_{t+1}), \tag{4.1}$$

where $Z(x_t) = \langle K_t^{pred}, p_{pose} \rangle (x_t)$. As it is often expensive to sample from the corresponding distribution, we show that it is possible to integrate p_{pose} in the updating step, where we write $g_{t+1}^Y(x_{t+1}) = g_{t+1}(y_{t+1} - h_{t+1}(x_{t+1}))$:

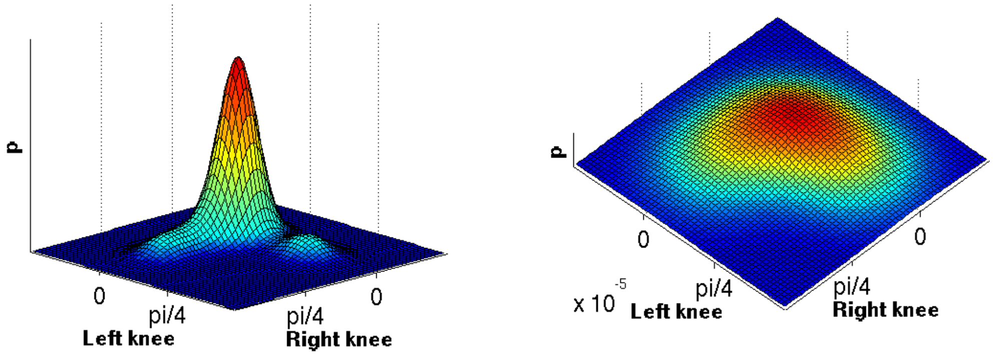


Figure 4.2: The Parzen estimate subject to the angles of the knee joints. **From left to right:** **a)** Using the Euclidean distance leads to a domination of the knee joints. The density rapidly declines to zero as the values differ from the data. **b)** The influence of the knees is reduced by the weighted Euclidean distance.

$$\eta_{t+1}(dx_{t+1}) = \frac{1}{\langle\langle \eta_t, K_t\rangle, g^Y_{t+1}\rangle} \overbrace{g^Y_{t+1}(x_{t+1})}^{\text{Updating}} \int_E \overbrace{K_t(x_t, dx_{t+1})}^{\text{Prediction}} \eta_t(dx_t)$$

$$= \frac{1}{\langle\langle \eta_t, \frac{1}{Z} K^{pred}_t\rangle, g^Y_{t+1} p_{pose}\rangle} \overbrace{g^Y_{t+1}(x_{t+1}) p_{pose}(x_{t+1})}^{\text{Updating}} \int_E \overbrace{\frac{1}{Z(x_t)} K^{pred}_t(x_t, dx_{t+1})}^{\text{Prediction}} \eta_t(dx_t)$$

Note that sampling from the distribution $K^{pred}_t(x_t, \cdot)/Z(x_t)$ is equivalent to sample from $K^{pred}_t(x_t, \cdot)$ for a given x_t. Hence, the prediction step of the particle filter remains unchanged whereas the particles are weighted by the product $g_{t+1}(y_{t+1} - h_{t+1}(x_{t+1}))\, p_{pose}(x_{t+1})$ during updating.

4.2.1 Probability of a Pose

Only in rare cases, we are able to give an analytical expression for p_{pose}. Instead, we suggest learning the probability of the various poses from a finite set of training samples. For a non-parametric estimate of the density, we use a Parzen-Rosenblatt estimator [Ros56, Par62] with a Gaussian kernel

$$p_{pose}(x) = \frac{1}{(2\,\pi\,\sigma^2)^{d/2}\,N} \sum_{i=1}^{N} \exp\left(-\frac{d(x, x_i)}{2\,\sigma^2}\right) \tag{4.2}$$

to deal with the complexity of the distribution, where N denotes the number of training samples and the function d is a distance measure in E. The estimate depends on the window size σ that is necessary to be chosen in an appropriate way. While a small value of σ forces the particles to stick to the training data, a greater value of σ approximates the density smoother. In order to cope with this, we chose σ as the maximum second nearest neighbor distance between all training samples, i.e. the two neighbors of a sample are at least within a standard deviation. Other values for the window size are discussed in detail in [Sil86].

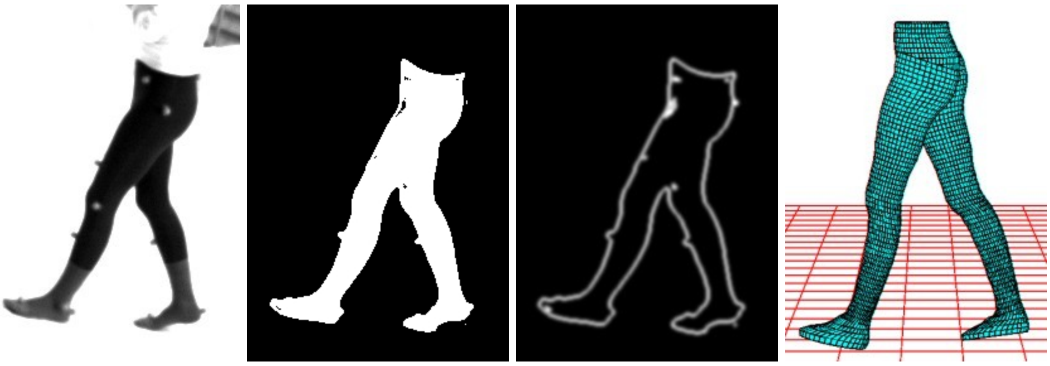


Figure 4.3: **From left to right:** (**a**) Original image. (**b**) Extracted silhouette. (**c**) The smoothed contour is slightly deformed by the markers needed for the marker-based system. (**d**) 3D model with 18 DOF used for tracking.

There are also several ways to specify the norm for evaluating the distance between a training sample x_i and a value x in the d-dimensional state space E for Equation (4.2). The commonly used Euclidean distance weights all dimensions of the state space uniformly. This means in the context of human motion estimation that a discrepancy of the knee contributes to the measured distance in the same matter as a discrepancy of the ankle. As illustrated in Figure 4.2, this involves a dominated measure by joints with a relatively large anatomical range as the knee in comparison to joints with a small range as the ankle. A weighted Euclidean distance measure incorporates the variance of the various joints, i.e.

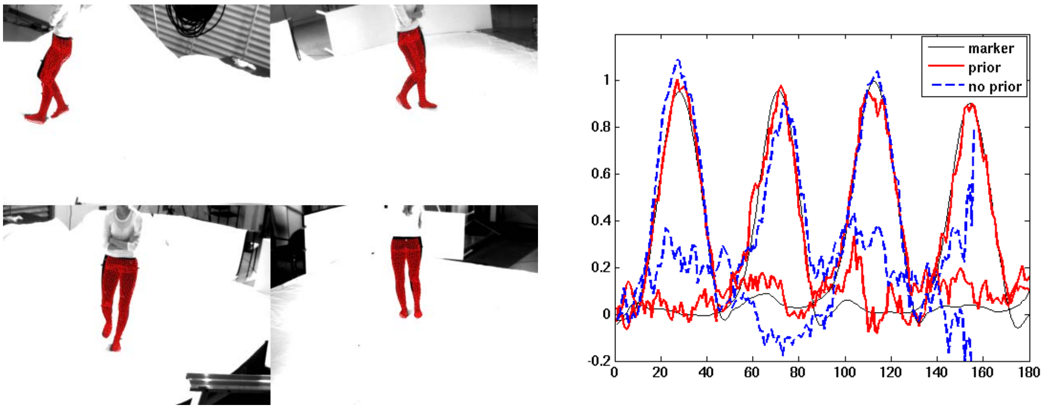
$$d(x, x_i) := \sqrt{\sum_{k=1}^{d} \frac{((x)_k - (x_i)_k)^2}{\rho_k}}, \qquad \rho_k := \frac{\sum_{i=1}^{N} \left((x_i)_k - \overline{(x)_k}\right)^2}{N-1}, \tag{4.3}$$

where $\overline{(x)_k}$ denotes the arithmetic mean of the samples in the kth dimension. This distance is generally applied in image analysis [MR98] and is equivalent to a Mahalanobis distance in the case that the covariance matrix is diagonal. A full covariance matrix significantly increases the computation in high dimensional spaces. We remark that p_{pose} takes only the skeleton configurations, i.e. joint angles, into account without the position and orientation of the human such that the prior is spatially invariant.

4.2.2 Experiments

To illustrate the impact of p_{pose}, we have used an annealed particle filter (APF) [DR05] for tracking the lower part of a human body. The annealed particle filter is actually a heuristic based on the generic particle filter as discussed in Section 2.3.2, but it suits our purpose. A more comprehensive discussion on the annealed particle filter is presented in [GRBS06] and will be given in Section 6.1.5. In our experiments we use four calibrated and synchronized cameras. The sequences are simultaneously captured by a commercial marker-based system from Motion Analysis [Mot08] allowing a quantitative error analysis. The black leg suit and the attached retroflective markers are required by the marker-based system, see Figure 4.3 a).

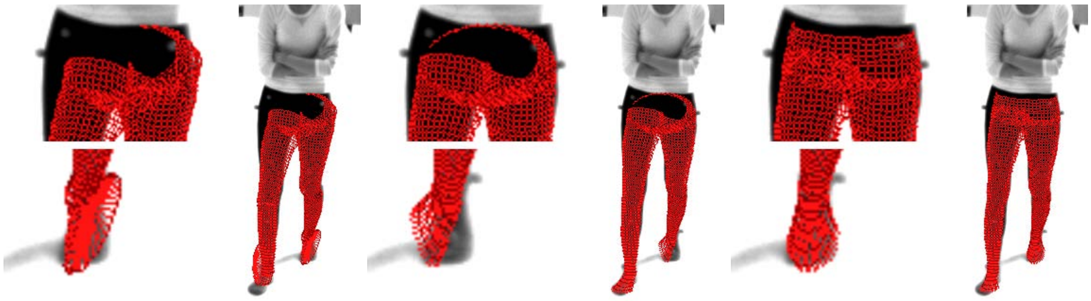
The likelihood $g_t^Y(x)$ is calculated pixel-wise similar to [DR05]. Each particle $x \in E$ determines a pose of our 3D model. The projected surface of the model onto the image plane gives a set of



(a)

(b)

Figure 4.4: **From left to right: a)** Results for a walking sequence captured by four cameras. **b)** The joint angles of the right and left knee. *Solid (thin):* Marker-based system. *Solid (thick):* Prior with weighted distance. *Dashed:* Without prior (Tracking fails).



(a)

(b)

(c)

Figure 4.5: Visual comparison of results. **From left to right:** (**a**) Without prior. (**b**) Without weighted distance. (**c**) With weighted distance.

silhouette points $P_i^S(x)$ and a set of contour points $P_i^C(x)$ for each view $i = 1, \ldots, r$ where a set contains all pixels $p \in \mathbb{R}^2$ of the silhouette and the contour, respectively. The silhouette S_i^Y of the observed object is obtained by a level set segmentation where $S_i^Y(p) = 1$ if p belongs to the foreground and $S_i^Y(p) = 0$, otherwise. The contour C_i^Y is just the boundary of the silhouette smoothed by a Gaussian filter and normalized between 0 and 1 as shown in Figures 4.3 b) and c). The likelihood is approximated by

$$g_t^Y(x) \approx \exp\left(-\sum_{i=1}^{r}\left(\Sigma_S^Y(x,i) + \Sigma_C^Y(x,i)\right)\right), \tag{4.4}$$

where

$$\Sigma_L^Y(x,i) := \frac{1}{\left|P_i^L(x)\right|}\sum_{p\in P_i^L(x)}\left(1 - L_i^Y(p)\right)^2 \tag{4.5}$$

for $L \in \{S, C\}$. According to Section 4.2, the prior is integrated by weighting the particles with $g_t^Y(x)p_{pose}(x)$.

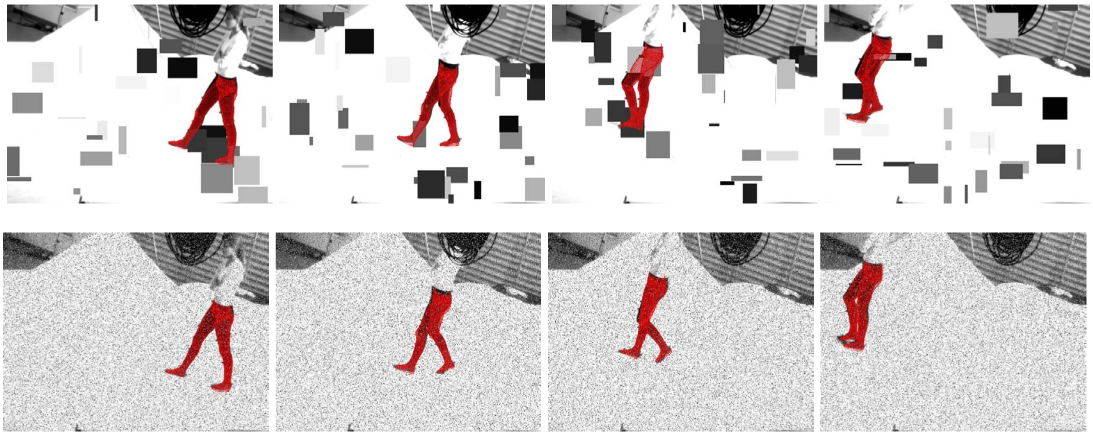


Figure 4.6: Results for distorted sequences (4 of 181 frames). Only one camera view is shown. **Top:** Occlusions by 30 random rectangles. **Bottom:** 25% pixel noise.

The training data used for learning p_{pose} consists of 480 samples obtained from walking sequences of the same person. The data was captured by the commercial system before recording the test sequences. The simple dynamical model K_t^{pred} is modeled by a zero-mean Gaussian distribution with covariance matrix determined by $0.1\,\rho_k$; see Equation (4.3). The initial distribution is the Dirac measure of the initial pose that is assumed to be known. Our implementation of the APF with 250 particles and 10 layers has taken several minutes for processing one frame. Figure 4.4 visualizes results of a walking sequence that is not contained in the training data. For the sake of comparison, the results of the APF without using prior knowledge at all are also visualized in Figure 4.5. The estimated angles of the left and the right knee are shown in the diagram in Figure 4.4 where the values acquired from the marker based system provide a ground truth with an accuracy of about 3 degrees. The root mean square (RMS) error for both knees is 6.2 degrees (red line). While tracking with 100 particles failed, our method also succeeded using 150 and 200 particles with RMS errors 15.3 and 8.8 degrees, respectively.
Figure 4.6 shows the robustness in the presence of noise and occlusions. Each frame has been independently distorted by 25% pixel noise and by occluding rectangles of random size, position, and gray value. The legs are tracked over the whole sequence with RMS errors 8.2 and 9.0 degrees, respectively. Finally, we have applied the method to a sequence with scissor jumps as shown in Figure 4.7. This demonstrates that our approach is not restricted to the motion patterns that have been used for training as it is when learning the patterns instead of the poses. However, the 7th image also highlights the limitations of the prior. Since our training data are walking sequences, the probability that both knees are bended is almost zero, cf. Figure 4.2. Therefore, a more probable pose is selected with less bended knees, which yields a higher hip of the 3D model than in the image. Overall, the RMS error is 8.4 degrees. A similar error can be observed for the feet since they are more bended for jumping as for walking. Nevertheless, the result is much better than without using any prior.

4.3 Summary

The main problem for applying filtering techniques to human motion capture is the need for accurate models. In particular, modeling the human dynamics with Markov kernels is very challenging. While many learning techniques provide solutions only for certain motion patterns, and

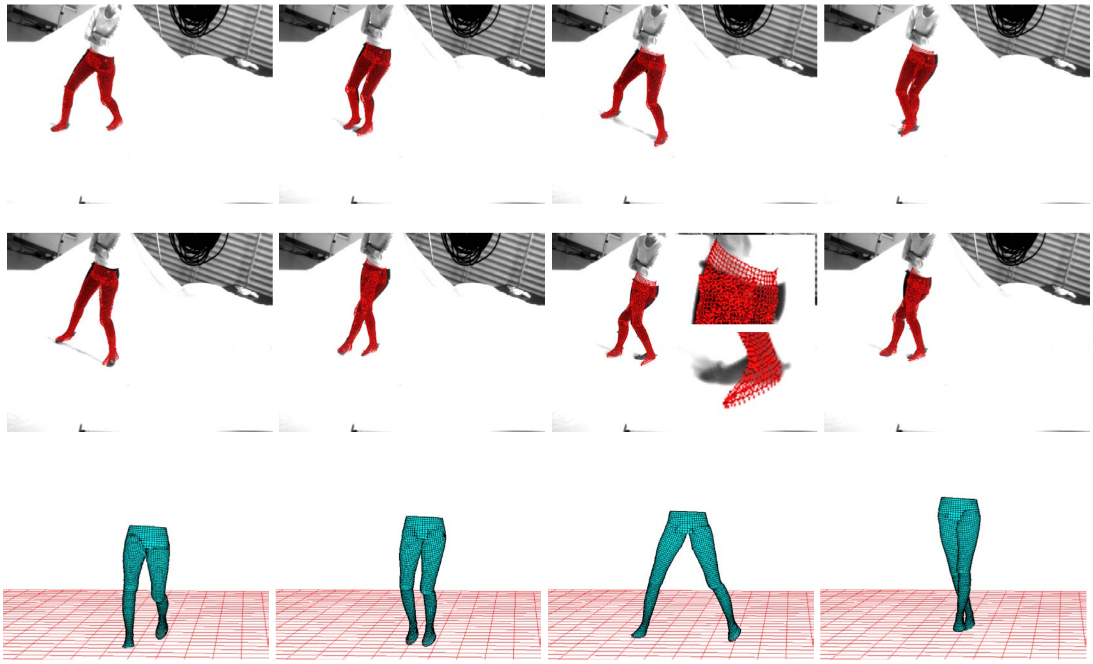


Figure 4.7: **Rows** 1-2**:** Results for a sequence with scissor jumps (8 of 141 frames). **Row** 3**:** The 3D models for the 4 poses on the left hand side of rows 1 and 2 are shown from a different viewpoint.

therefore very application specific solutions, a general motion model is a crucial step towards commercial applications. We have proposed a motion pattern independent prior that compensates at least to some extent for the weak dynamical model. The prior ensures that particles representing a familiar pose are favored such that the state space becomes more constrained. Even though it improves particle filter approaches, it is not a replacement for a good dynamical model. Without such model, accurate results for human motion capture in reasonable time are barely to achieve. Hence, it raises the question of whether filtering is the best formulation of model-based human motion capture. The main advantages of filtering approaches over optimization approaches are basically the ability to handle noise and to resolve ambiguities over time. The example given in Figure 4.1, however, indicates that pixel noise is not a serious problem in this context. Therefore, it is an important issue of whether optimization techniques can handle model-based human motion capture better.

5

Local Optimization

Local optimization has been successfully applied to human pose estimation as discussed in Section 2.3.1. In this chapter, we focus on the local optimization approach based on twists that has been proposed by Bregler and Malik [BM98] and that is also used in [RBS$^+$06]. It is motivated by the field of robotics where twists are commonly used for modeling the kinematics of articulated robots [MLS94]. Letting the optimization scheme fixed, we address the question *"What are good cues for human motion capture?"* from Section 1.5.1. Before discussing the advantages and disadvantages of various cues, we summarize the used local optimization in Section 5.1. Finally, we introduce an analysis-by-synthesis framework that combines complementary cues to track a variety of objects. The potential of the framework is demonstrated on various sequences and on a challenging real-world problem, namely crash test video analysis.

5.1 Pose Estimation

The pose of a rigid object or a human is represented by a set of twists $\theta_j\hat{\xi}_j$ as discussed in Section 3.1. A transformation of a vertex X_i on the limb k_i influenced by n_{k_i} joints is given by

$$X'_i = \exp\left(\theta\hat{\xi}\right)\exp\left(\theta_{\iota_{k_i}(1)}\hat{\xi}_{\iota_{k_i}(1)}\right)\ldots\exp\left(\theta_{\iota_{k_i}(n_{k_i})}\hat{\xi}_{\iota_{k_i}(n_{k_i})}\right)X_i, \tag{5.1}$$

where the mapping ι_{k_i} represents the order of the joints in the kinematic chain. Since the joint motion depends only on the joint angle θ_j, the state of a kinematic chain is defined by a parameter vector $\chi := (\theta\xi, \Theta)$ that consists of the six parameters for the global twist $\theta\hat{\xi}$ and the joint angles $\Theta := (\theta_1, \ldots, \theta_N)$. For estimating the parameters χ, a sufficient set of 3D-2D point correspondences is needed. How such correspondences are obtained, is subject of the next sections. For the moment, we assume that such a set of correspondences (X_i, x_i) with $X_i \in \mathbb{R}^3$ and $x_i \in \mathbb{R}^2$ is already given and represented in homogeneous coordinates.

As the camera parameters are known as explained in Section 3.2, the projection rays can be reconstructed from the 2D points x_i. 3D lines can be represented implicitly by so-called *Plücker lines* [Sto91, She98]. A Plücker line $L_i = (n_i, m_i)$ is determined by a unit vector n_i and a moment m_i, where $m_i = x \times n_i$ for a point x on the line. Thereby, × denotes the cross product. This implicit representation allows to conveniently determine the distance of a 3D point to this line. Consequently, the distance of a pair (X'_i, x_i) is given by the norm of the perpendicular vector between the line L_i and the point X'_i:

$$\|\Pi\left(X'_i\right) \times n_i - m_i\|_2, \tag{5.2}$$

where Π denotes the projection from homogeneous coordinates to non-homogeneous coordinates.

Provided the 3D-2D point correspondences are correct, the transformed 3D points must be on the projection rays reconstructed from their corresponding 2D points. In practice the correspondences are not exact for various reasons, yet we can seek to minimize the above distance. In particular, we seek a transformation $\chi = (\theta\xi, \Theta)$ applied to all points X_i such that the total distance over all correspondences is minimized in the least squares sense:

$$\underset{\chi}{\operatorname{argmin}} \frac{1}{2} \sum_i \left\| \Pi \left(\exp\left(\theta\hat{\xi}\right) \prod_{j=1}^{n_{k_i}} \exp\left(\theta_{\iota_{k_i}(j)} \hat{\xi}_{\iota_{k_i}(j)}\right) X_i \right) \times n_i - m_i \right\|_2^2 . \tag{5.3}$$

It is worth noting that minimizing the distance to the 3D ray and minimizing the 2D re-projection error can be made equivalent by appropriate rescaling of each error vector [RBW07].
Equation (5.3) states a nonlinear least squares problem due to the exponential form of the transformation matrices. In order to solve for the parameters, we use the Gauß-Newton method, i.e., the transformation matrix is linearized and the parameter estimation is iterated. With the identity matrix I and

$$\exp(\theta\hat{\xi}) = \sum_{k=0}^{\infty} \frac{(\theta\hat{\xi})^k}{k!} \approx I + \theta\hat{\xi}, \tag{5.4}$$

we can approximate Equation (5.3) as the linear least squares problem

$$\underset{\chi}{\operatorname{argmin}} \frac{1}{2} \sum_i \left\| \Pi \left(\left(I + \theta\hat{\xi} + \sum_{j=1}^{n_{k_i}} \theta_{\iota_{k_i}(j)} \hat{\xi}_{\iota_{k_i}(j)} \right) X_i \right) \times n_i - m_i \right\|_2^2 , \tag{5.5}$$

which can be solved, for instance, with the Householder algorithm. To this end, Equation (5.5) can be written as linear system:

$$\begin{pmatrix} A_1 \\ A_2 \\ \cdot \\ A_r \end{pmatrix} \chi^T = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ b_r \end{pmatrix}, \tag{5.6}$$

where $A_i \in \mathbb{R}^{3\times(6+N)}$ and $b_i \in \mathbb{R}^3$. For convenience, we write

$$A_i = \left(\left(A_i^1\right)_{3\times3} \left(A_i^2\right)_{3\times3} \left(A_i^{j_1}\right)_{3\times1} \cdots \left(A_i^{j_N}\right)_{3\times1} \right)_{3\times(6+N)} . \tag{5.7}$$

The matrices A_i^1 and A_i^2 for the global transformation are given by

$$A_i^1 = \begin{pmatrix} 0 & n_{i,3} & -n_{i,2} \\ -n_{i,3} & 0 & n_{i,1} \\ n_{i,2} & -n_{i,1} & 0 \end{pmatrix} \tag{5.8}$$

and

$$A_i^2 = \begin{pmatrix} -X_{i,3}n_{i,3} - X_{i,2}n_{i,2} & X_{i,1}n_{i,2} & X_{i,1}n_{i,3} \\ X_{i,2}n_{i,1} & -X_{i,1}n_{i,1} - X_{i,3}n_{i,3} & X_{i,2}n_{i,3} \\ X_{i,3}n_{i,1} & X_{i,3}n_{i,2} & -X_{i,2}n_{i,2} - X_{i,1}n_{i,1} \end{pmatrix} . \tag{5.9}$$

The matrix A_i^j for the joint j is $(0\ 0\ 0)^T$ if the vertex X_i is not influenced by joint j. Otherwise,

$$A_i^j = \begin{pmatrix} (X_{i,1}n_{j,3} - X_{i,3}n_{j,1} + m_{j,2})\, n_{i,3} - (X_{i,2}n_{j,1} - X_{i,1}n_{j,2} + m_{j,3})\, n_{i,2} \\ (X_{i,2}n_{j,1} - X_{i,1}n_{j,2} + m_{j,3})\, n_{i,1} - (X_{i,3}n_{j,2} - X_{i,2}n_{j,3} + m_{j,1})\, n_{i,3} \\ (X_{i,3}n_{j,2} - X_{i,2}n_{j,3} + m_{j,1})\, n_{i,2} - (X_{i,1}n_{j,3} - X_{i,3}n_{j,1} + m_{j,2})\, n_{i,1} \end{pmatrix}, \tag{5.10}$$

where (n_j, m_j) is the Plücker line representation for the rotation axis of the joint j. The vector $b_i \in \mathbb{R}^3$ is given by

$$b_i = \begin{pmatrix} X_{i,3}n_{i,2} - X_{i,2}n_{i,3} + m_{i,1} \\ X_{i,1}n_{i,3} - X_{i,3}n_{i,1} + m_{i,2} \\ X_{i,2}n_{i,1} - X_{i,1}n_{i,2} + m_{i,3} \end{pmatrix}. \tag{5.11}$$

Each point correspondence yields three equations of rank two, i.e., at least three correspondences are required for a unique solution in case of a rigid object. For each limb, at least one additional correspondence for a point on this limb is needed. If the limb has three degrees of freedoms, two correspondences are required. The acquisition of such a set of correspondences (X_i, x_i) from image data is discussed in the following Sections 5.2, 5.3, and 5.4.

5.2 Region-based Tracking

One of the most popular cues for human motion capture is silhouettes or the silhouette contours. In the simplest case, they can be extracted by background subtraction. A more general procedure couples level-set segmentation with pose estimation [RBW07]. It has the distinct advantage that it does not require a static background which makes it suitable for outdoor scenes and multiple moving objects.

Since the energy functional (3.12) in Section 3.3 does not take the knowledge of the 3D shape into account, it tends to separate other regions than the object regions. Hence, additional constraints are required in order to restrict the sought contour to stay close to the object shape. To this end, the functional (3.12) is extended by an additional term. This term implements the model assumption that the shape in the image should be close to the projection of the surface model. The extended energy reads:

$$\begin{aligned} E(\Phi, p_1, p_2, \chi) = & \underbrace{-\int_\Omega H(\Phi) \ln p_1 + (1 - H(\Phi)) \ln p_2 \, dx + \vartheta \int_\Omega |\nabla H(\Phi)| \, dx}_{\text{segmentation}} \\ & \underbrace{+ \lambda \int_\Omega (\Phi - \Phi_\chi)^2 \, dx,}_{\text{shape distance}} \end{aligned} \tag{5.12}$$

where Φ_χ denotes the shape of the projected surface model with pose parameters χ. In order to compare the object shape Φ_χ with the level-set function Φ, the shape is represented by the signed Euclidean distance function. To this end, the surface is transformed according to Equation (5.1) and projected onto the image plane as described in Section 3.2. After applying a signed distance transform [FH04] to the binary image of the projected surface $\widetilde{\Phi}_\chi$, one obtains

$$\Phi_\chi(x) = \begin{cases} \operatorname{dist}(x, C) & \text{if } \widetilde{\Phi}_\chi(x) > 0, \\ -\operatorname{dist}(x, C) & \text{otherwise,} \end{cases} \tag{5.13}$$

where $\text{dist}(x, C)$ is the shortest Euclidean distance of point x to the contour of $\widetilde{\Phi}_\chi$. It needs to be emphasized that any skinning method described in Section 3.1 can be used for transforming the surface model according to the pose parameters χ. In general, we use the notation of the articulated model given by Equation (5.1) as a synonym for any kind of skeleton-based mesh transformation throughout the paper.

The last term of Equation (5.12) couples the segmentation model and the pose parameters as it enforces the projected surface model to match the object region. This has two effects: firstly, the pose parameters are adapted such that the projection fits the region extracted by the segmentation part. Secondly, the segmentation is constrained by the shape of the surface model and is not allowed to deviate too much from this shape. The tolerated amount of deviation depends on the clarity of the image data and the choice of λ.

The energy function (5.12) can be minimized with respect to Φ and p_i by gradient descent. Having an initialization of Φ by the projected object surface Φ_χ, we can estimate p_i as described in Section 3.3. From this we can update Φ by

$$\Phi^{k+1} = \Phi^k + H'(\Phi^k)\left(\ln\frac{p_1^k}{p_2^k} + \vartheta \text{div}\left(\frac{\nabla\Phi^k}{|\nabla\Phi^k|}\right)\right) + 2\,\lambda(\Phi_\chi - \Phi^k) \tag{5.14}$$

with iteration index k. When moving on to a new frame, it makes sense to run a few iterations with the densities from the previous frame before adapting p_i. This allows the contour to capture the new position of the object boundary. We assume the distribution to be sufficiently smooth for being valid also for the displaced regions in the new frame. This smoothness is ensured by the large window $\rho = 12$ for the Gaussian kernel K_ρ (3.18).

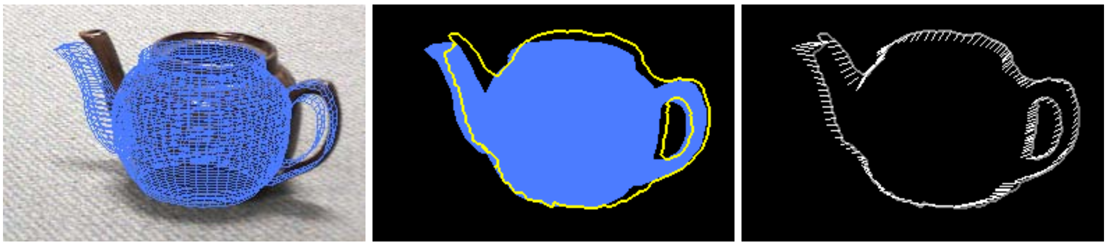

(a) (b) (c)

Figure 5.1: Closest point correspondences. **a)** One seeks 2D-3D correspondences between the image and the 3D mesh model. **b)** Projected object surface in blue and the extracted object contour in yellow. **c)** Correspondences between the silhouette of the blue area and the yellow contour.

The shape distance in (5.12)

$$\int_\Omega (\Phi - \Phi_\chi)^2\, dx \tag{5.15}$$

relates the pose parameters χ to the region represented by Φ. To estimate the pose parameters for a given Φ, we need 2D-3D point correspondences. Since Φ_χ is the projection of the object model, corresponding 3D points on the model are known. Hence, 2D-3D correspondences can be derived by matching the 2D shapes Φ and Φ_χ. Towards this end, we seek the displacement vector field $(u(x), v(x))$ that minimizes

$$\int_\Omega (\Phi(x, y) - \Phi_\chi(x + u, y + v))^2\, dx. \tag{5.16}$$

In practice, we are only interested in correspondences for points along the contours.
Numerous methods on 2D shape matching can be found in the literature. We are interested in a method that can deal with shape deformations in order to handle projective distortion and articulated objects. Moreover, we can assume that the transformation between the shapes is limited. Thus a local method is most effective. A suitable and simple method is closest point search. It can be computed efficiently, if the two contours Φ and Φ_χ are represented by distance images, i.e., the value of $\Phi(x)$ is the minimum distance of x to the contour. A very efficient method for computing the minimum Euclidean distance in linear time has been proposed in [FH04]. A more robust shape matching can be achieved by optical flow [RBCS06], but it is also more expensive.
The pose parameters, and consequently the shape prior Φ_χ, and the level set function Φ are optimized in an iterative, alternating scheme that is initialized with the pose from the previous frame. In summary, the steps

1. Compute shape prior Φ_χ
2. Estimate contour Φ (5.14)
3. Shape matching between Φ_χ and Φ
4. Estimate pose parameters χ (5.5)

are iterated until the pose converges or the maximum number of iterations is reached.
Although the described region-based approach performs well even in case of occlusions and noise [RBW07], it requires many iterations until convergence which makes it very expensive. Particularly for large transformations from frame to frame, the segmentation and consequently the pose estimation usually get stuck in a local optimum. Another problem is ambiguous solutions. For instance, the pose of a sphere cannot be uniquely determined from its silhouette. Hence, more than one feature is needed for robust tracking.

5.3 Motion-based Tracking

In order to track also objects in case of fast motions and large deformations, frame to frame correspondences are required. In this section, we consider two methods that compute 2D correspondences between successive frames t and $t+1$: optical flow and SIFT tracking. We assume the pose parameters of the model in frame t to be known. Therefore, it is known, how 3D model points project into this frame. Finding the new positions of the projected points in frame $t+1$ by either optical flow or the SIFT tracker yields 2D-3D point correspondences at $t+1$. From these the new pose of the object can be estimated using the technique described in Section 5.1.
Such a procedure obviously accumulates errors over time. This is due to the assumption that the pose in the previous frame is known and is exact. As a consequence, even the smallest estimation errors are propagated from frame to frame. Therefore it is crucial to combine motion-based correspondences with region-based ones.

5.3.1 Optical Flow

Optical flow is the common name for the displacement field $w(x) := (u(x), v(x), 1)$ between two images of an image sequence $I(x)$, where $x := (x, y, t)$. Numerous optical flow estimation methods can be found in the literature. Variational methods currently mark the state-of-the-art and yield dense flow fields. Since we are interested in capturing large displacements, we further focus on multi-resolution methods. Building upon the method in [BBPW04, BW05], we seek the optical flow as the minimizer of

$$\begin{aligned} E(u,v) = &\int_{\Omega_1} r(x) \cdot \Psi_1\big(|I(x+w) - I(x)|^2\big)\, dx \\ &+\gamma \int_{\Omega_1} r(x) \cdot \Psi_1\big(|\nabla I(x+w) - \nabla I(x)|^2\big)\, dx \\ &+\alpha \int_{\Omega_1} \Psi_2\left(|\nabla u|^2 + |\nabla v|^2\right) dx. \end{aligned} \tag{5.17}$$

The energy consists of two parts. The first part states the gray value and the gradient constancy assumption, both weighted relatively to each other by the parameter $\gamma = 5$. This part is usually called data term. It is weighted locally by $r(x)$, which will be explained later. The second term introduces the assumption of a smooth flow field. It is weighted by the parameter $\alpha \geq 0$. $\Psi_1(s^2)$ and $\Psi_2(s^2)$ are so-called robust penalizer functions [BA96, MP96]. In [BBPW04], $\Psi_1(s^2) = \Psi_2(s^2) = \sqrt{s^2 + \epsilon^2}$ with $\epsilon = 0.001$. Such a penalizer allows for outliers in the data (e.g. due to noise, specularities, or occlusions) and in the smoothness assumptions (due to motion discontinuities). We adopt the same functions for tracking articulated objects and choose $\alpha = 50$.

In case of rigid objects, the model can be simplified by setting $\Psi_2(s^2) = s^2$ and $\alpha = 800$, which leads to a linear term in the Euler-Lagrange equations of the smoothness term. This simplification results in a faster implementation. It becomes possible because in contrast to [BBPW04] the energy is only integrated inside the object region Ω_1. The object region is a byproduct of model-based tracking and beneficial as it already determines most of the relevant motion discontinuities. In case of rigid objects that are far enough from the camera, it even captures all relevant motion discontinuities. This is different for articulated objects. One could imagine, e.g., the case of two legs next to each other where one leg partially occludes the other. The legs can move in opposite direction, hence creating a motion discontinuity within the object region.

The model in [BBPW04] can be further extended by the explicit, local weighting $r(x)$ of the data term. This weighting is for integrating the result of an occlusion detection, which is described in [BRGC09]. The weights are set to

$$r(x) = \begin{cases} 0 & \text{if } x \text{ occluded} \\ 1 & \text{else.} \end{cases} \tag{5.18}$$

At occluded pixels the data term is ignored and only the smoothness term determines the estimated flow. This yields a smooth interpolation of the flow field in areas, where the data does not reflect the motion of the object.

The minimizer of (5.17) can be computed with a continuous optimization method in a multi-resolution setting. After discretization of the Euler-Lagrange equations, we obtain a nonlinear system of equations that can be solved via two nested fixed point iteration loops and a solver for sparse linear systems. For details we refer to [BBPW04]. With a fast multi-grid solver, the optical flow can be computed in real-time [BW05].

Since we are interested in an adaptive weighting of optical flow correspondences versus correspondences from other cues, we need some measure that tells us something about the local confidence of the computed optical flow. A standard confidence measure is the gradient magnitude of the image $c_{\text{grad}}(x) = |\nabla I(x)|$ or some similar expression [BFB94]. However, this measure does not perform well in case of contemporary, variational optical flow methods, as pointed out in [BW06b]. Instead, it was proposed in [BW06b] to employ the local energy of variational methods as a confidence measure. We adopt this idea and use

$$\begin{aligned} c_{\text{Energy}}(x) &= \beta\left(1 + e(x)\right)^{-1} \\ e(x) &:= \Psi_1\left(|I(x+w) - I(x)|^2\right) \\ &\quad +\gamma\Psi_1\left(|\nabla I(x+w) - \nabla I(x)|^2\right) \\ &\quad +\alpha\Psi_2\left(|\nabla u|^2 + |\nabla v|^2\right) \end{aligned} \tag{5.19}$$

according to the energy stated in (5.17). This confidence measure is small in areas, where the assumptions stated in the energy functional cannot be fulfilled. Consequently, it indicates areas where optical flow computation is difficult and not reliable. Point correspondences derived from the optical flow are weighted by this confidence value. The factor β normalizes the confidence, such that $c_{\text{Energy}} = 1$, if the optical flow computation works reasonably well. If the confidence is larger, the correspondence obtains more influence than average, and if it is smaller, its relative influence is decreased. Empirical evaluation resulted in $\beta = 12$ for $\Psi_2(s^2) = s^2$ and $\beta = 3$ for $\Psi_2(s^2) = \sqrt{s^2 + \epsilon^2}$.

5.3.2 SIFT

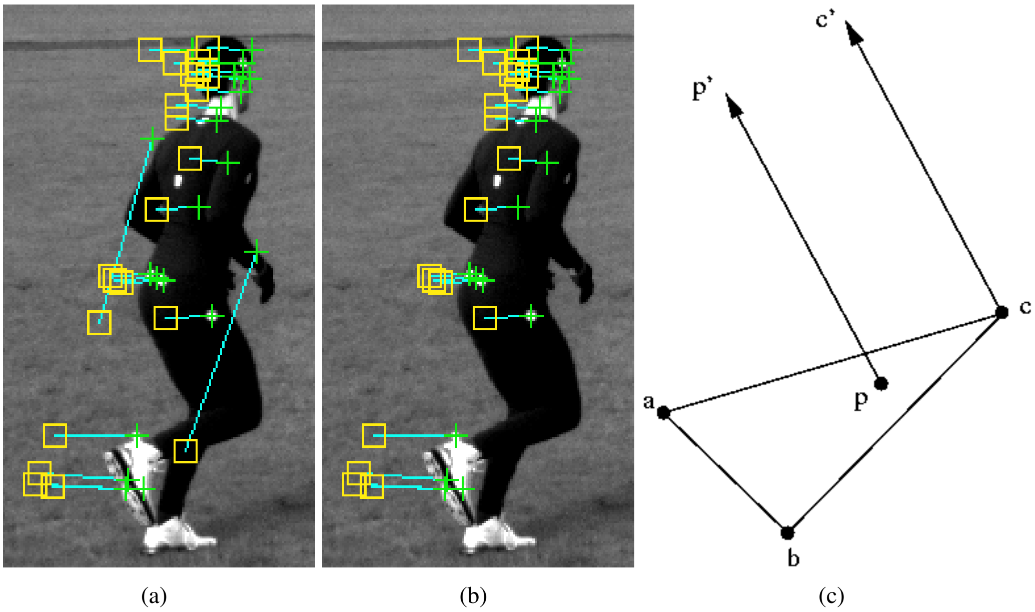


Figure 5.2: **From left to right: a)** Matches between previous frame (squares) and current frame (crosses). **b)** The outliers are removed after filtering. **c)** When a matched SIFT keypoint p does not coincide with a projected mesh vertex, the 2D translation vector $p' - p$ is added to the closest vertex (here c). For the new 2D-2D correspondence c - c', the 2D-3D counterpart is available.

The scale invariant feature transform and its corresponding region descriptor [Low04] currently belong to the most reliable techniques for sparse matching [MS05]. Matching is restricted to keypoints which correspond to local extrema in scale-space. Each keypoint is described by orientation histograms computed in its neighborhood [Low04]. Correspondences between successive images are then established by nearest neighbor distance ratio matching [MS05] where

conflicting correspondences are deleted. We used the distance ratio threshold of 0.6. Only keypoints that belong to the object region and are not occluded are considered.
As shown in Figure 5.2, the matching produces reliable point correspondences but also some outliers that need to be eliminated. The rudest mismatches for each pair of images are removed by discarding correspondences with a Euclidean distance that exceeds the average by a multiple. When the average is above a threshold, we also delete corresponding features with the same location since the match in frame $t + 1$ usually belongs to a static object in the background in this case. Such pre-selection increases the inliers-to-outliers ratio, though it does not restrict the applicability to static backgrounds, as demonstrated in Figure 5.6. After deriving the 2D-3D correspondences, a preliminary pose is estimated and the new 3D correspondences are projected back in order to detect the remaining outliers.
In contrast to dense optical flow where a point correspondence is available for each projected mesh vertex, SIFT keypoints do not necessarily coincide with the projected mesh points. However, if the mesh is fine enough, we can assume the closest projected mesh point to undergo approximately the same 2D translation between two successive images as the SIFT keypoint. This is illustrated on the right hand side of Figure 5.2.
Thanks to the outlier detection and the high overall robustness of SIFT matching, a separate confidence measure like in case of the optical flow is not needed. The influence of SIFT correspondences automatically increases with the number of successful matches. In case of poorly structured objects, the number of these matches, and thus the influence of SIFT, will be low.

5.3.3 Overview

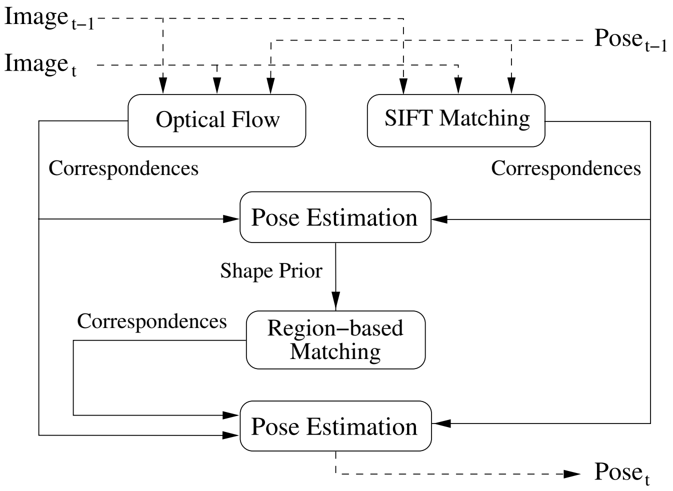


Figure 5.3: Optical flow or SIFT features are used to establish correspondences between the successive frames $t - 1$ and t. Knowing the pose for frame $t - 1$, the pose is estimated from these correspondences. The pose is further refined by region-based matching. To this end, the estimated pose is used for computing the shape prior Φ_χ. After segmentation, the shapes Φ and Φ_χ are matched. The final pose is then estimated from correspondences established by motion-based and region-based cues.

Figure 5.3 depicts an overview of the tracking system that combines region-based and motion-based cues. While optical flow or SIFT features estimate large transformations between two successive frames, the region-based pose estimation compensates for the errors of motion-based tracking. Towards this end, 2D-2D correspondences between two successive frames are established by matching SIFT features (Section 5.3.2) or optical flow (Section 5.3.1). Since the pose χ from the previous frame is known, 3D-2D correspondences can be derived to estimate the pose (Section 5.1). This gives a better estimate of the shape prior Φ_χ than the pose from the previous frame. Finally, the three steps from region-based pose estimation (Section 5.2) are performed: estimation of the contour Φ (5.14), shape matching between Φ_χ and Φ, and estimation of the pose parameters χ (5.5). For the pose estimation, however, correspondences between 2D image points and 3D model points are established from different cues, namely by matching the projected model to the object region in the image, by matching image points in successive frames via optical flow, and by matching SIFT keypoints of successive frames.

5.3.4 Fusion

Correspondences from different cues can be easily combined in the least-squares framework by considering all of them in the sum of Equation (5.3). Nonlinear optimization with the Gauß-Newton method yields the optimum pose considering all constraints in the least-squares sense, which is related to the assumption of a Gaussian error distribution.

If there is a way to estimate the expected deviation of the matched points, for instance through a confidence measure, this can be incorporated by means of the variance of the Gaussian distribution. The sums in (5.3) and (5.5) can be replaced by weighted sums $\sum_i w_i \| \cdot \|$, where w_i corresponds to the inverse variance of the Gaussian distribution. This leads to the well-known weighted least-squares setting.

The relative influence of a certain type of correspondence, i.e. flow, SIFT, or region-based, depends on the sum of all weights over each set. Obviously, there are large differences in the size of the sets. While optical flow provides several thousand correspondences, usually only few SIFT keypoints are detected in the object region. On the other hand, a matched SIFT keypoint tends to be more reliable than a point correspondence established by the optical flow. The different numbers of points can be balanced by normalizing the weights w_i with the size of the respective point set.

The main idea is to reduce the relative influence of correspondences, which probably contain large errors. In case of the optical flow this is achieved through a confidence measure based on the local energy, which allows weighting each correspondence individually. In case of SIFT keypoints there is no such measure, yet the number of keypoints is a good indicator for the appropriateness of SIFT in a certain scene. The confidence of the region-based cue is directly integrated in the interleaved segmentation and pose estimation. In areas where the segmentation is evident, i.e. the difference of the log-likelihoods of foreground and background is large, the image driven part of the segmentation energy dominates the shape prior and the contour can deviate much from the projected model. Vice-versa, if the foreground and background distributions fit almost equally well, the segmentation will stay close to the shape prior, which reflects the pose estimated with flow and SIFT based correspondences. Hence, the confidence of the contour-based correspondences is already reflected by the distance of the projected mesh points to the corresponding points on the contour.

These considerations suggest the following weighting strategy. Let n_{C} and n_{OF} denote the number of contour- and flow-based correspondences, respectively. We take the contour-based correspondences as reference and assign all of them $w_{\mathrm{C}} = 1$. SIFT correspondences are all assigned the weight $w_{\mathrm{SIFT}} = 0.002 \cdot n_{\mathrm{C}}$. Optical flow correspondences are weighted individually by means of the confidence measure described in Section 5.3.1. For a correspondence i with

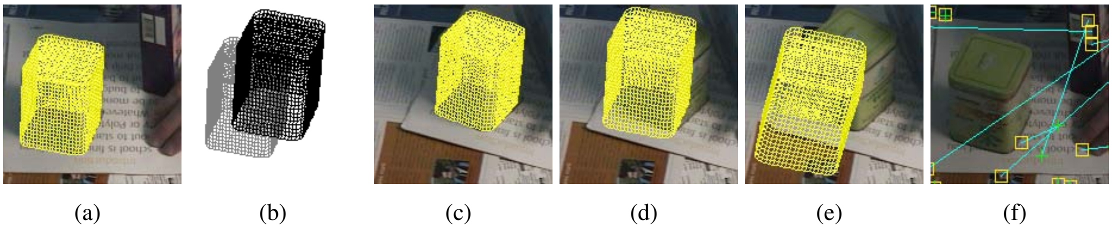


Figure 5.4: Combining motion- and region-based tracking allows capturing the large motion of a tea box. **From left to right: a)** Object pose at frame 1. **b)** Object motion due to the estimated optical flow between frame 1 and frame 2. Gray: pose from frame 1. Black: pose prediction for frame 2. **c)** Estimated pose at frame 2 using combined motion- and region-based tracking. **d)** Bad pose just using motion-based tracking. **e)** Bad pose just using region-based tracking. **f)** Not enough distinctive SIFT features are located to allow for a proper prediction. Flow-, region-, or SIFT-based tracking alone cannot handle this situation.

confidence c_i, we assign the weight $w_i = c_i \frac{n_\mathrm{C}}{n_\mathrm{OF}}$. For the pose prediction, where no contour-based cues are available, the factor n_C is replaced by n_OF, respectively.
If all cues can be extracted in an equally reliable manner, they are all weighted equally. As soon as one of them is more reliable, its relative influence is increased. Effectively, this kind of adaptive weighting automatically chooses the most reliable cue and ensures independence of the sampling of the mesh. For this reason, it is possible to fix this weighting procedure and run the method on different data preferring different cues.

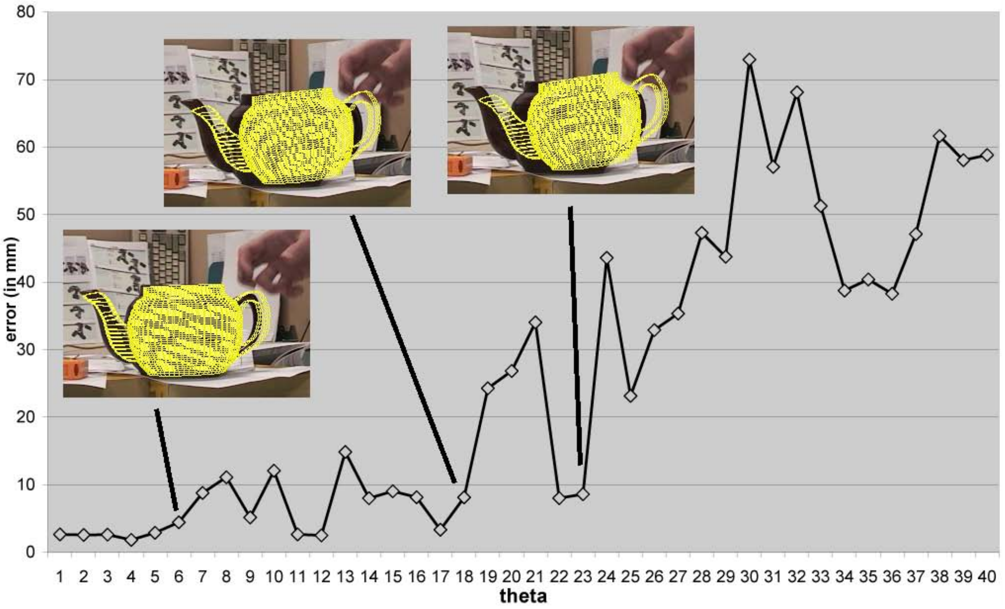


Figure 5.5: Sensitivity of the region based method on the initial pose. The diagram shows the average error of the mesh points depending on the amount of disturbance from the correct pose. Three key initial poses are depicted in the images.

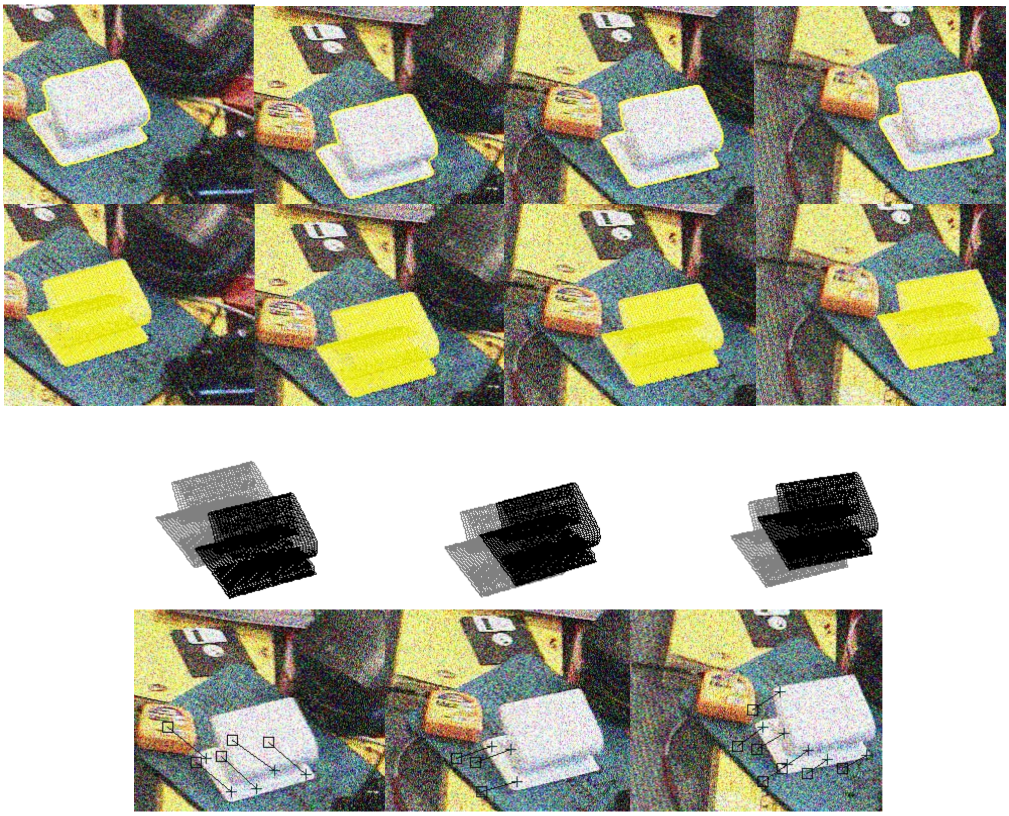


Figure 5.6: Four successive frames from a sequence with the camera moving and Gaussian noise with standard deviation 60 added (140 frames, 8fps). **First row:** Extracted contour. **Second row:** Estimated pose. **Third row:** Object motion due to optical flow correspondences. Gray: pose from previous frame. Black: pose prediction at current frame. **Last row:** A very similar result is obtained with the SIFT tracker.

5.3.5 Experiments

In order to demonstrate the ability of the tracking system to deal with a number of challenges, we applied it to numerous tracking scenes. These scenes contain homogeneous as well as textured objects, large transformations, noisy images, partial occlusions, and articulated human motion. With the experiments we aim at showing that, due to the combination of complementary cues and their adaptive weighting, the tracking system can handle all these scenes without the need of manual adaptations.

Rigid objects

Figure 5.4 depicts an experiment where a tea box has been moved by about 30 pixels between two frames including a rotation. As the transformation is quite large, the computed optical flow vectors contain errors. This can be seen from the pose prediction in Figures 5.4 b) and d). However, thanks to the additional region-based correspondences, the final pose result is good (Figure 5.4 c)). Conversely, the pose estimation also fails if only the region-based correspondences are used. This is shown in Figure 5.4 e). Figure 5.4 f) reveals that in this scene there are not enough SIFT keypoints on the object (only one, to be precise) for tracking the tea box. This

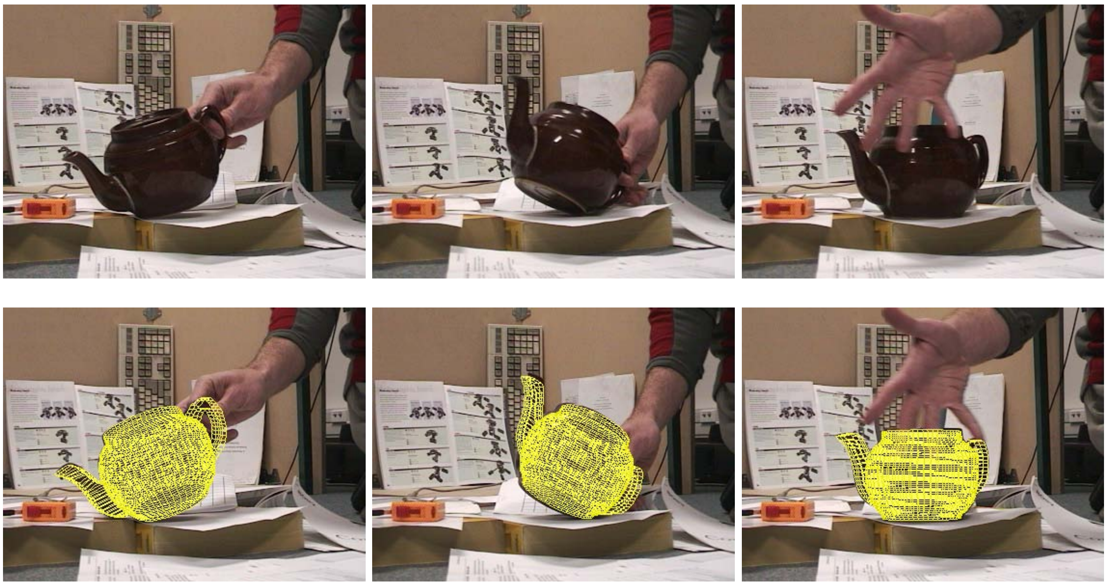


Figure 5.7: **Top row:** Frames 97, 116, and 188 of a stereo sequence used for the experiments in Table 5.1. **Bottom row:** Tracking results.

noise level	0	20	40	60	80
region	124	115	95	85	5
region+flow	tracked	115	115	75	5
region+SIFT	tracked	110	100	25	5
region+flow+SIFT	tracked	115	115	85	5

Table 5.1: Sensitivity to noise in the input images. The table indicates the frame number where tracking failed. The sequence contains 196 images. Some of them are shown in Figure 5.7.

experiment demonstrates two things. Firstly, there are scenes where none of the cues alone is able to correctly track the object. Taking region- and motion-based cues together, on the other hand, leads to a successful tracking. Secondly, there is clearly a difference between the usage of correspondences from optical flow and SIFT. While the estimated flow might not be exact in difficult situations, it provides at least enough correspondences for a unique approximate solution. SIFT correspondences are usually more reliable, but their number is sometimes not sufficient to estimate the pose.

In order to evaluate the sensitivity of the region based pose estimation on the initialization, we added increasing perturbations to the correct pose. This kind of experiment is also commonly used in the scope of active appearance models [MB04]. The perturbing twists were $0.01\theta(10, 10, 10, 0.5, 0.5, 0.5)^\top$ for increasing θ. The remaining average deviation of all mesh points is depicted in Figure 5.5 together with the initial poses for three θ. Clearly, the method can deal very well with small perturbations, and the pose estimates are still quite good with medium perturbations. The reason for some smaller perturbation leading to inferior results than a larger perturbation is due to different ways from the initialization to the next optimum. Already very small structures can be the reason for a local minimum. Initializations that are too far away lead

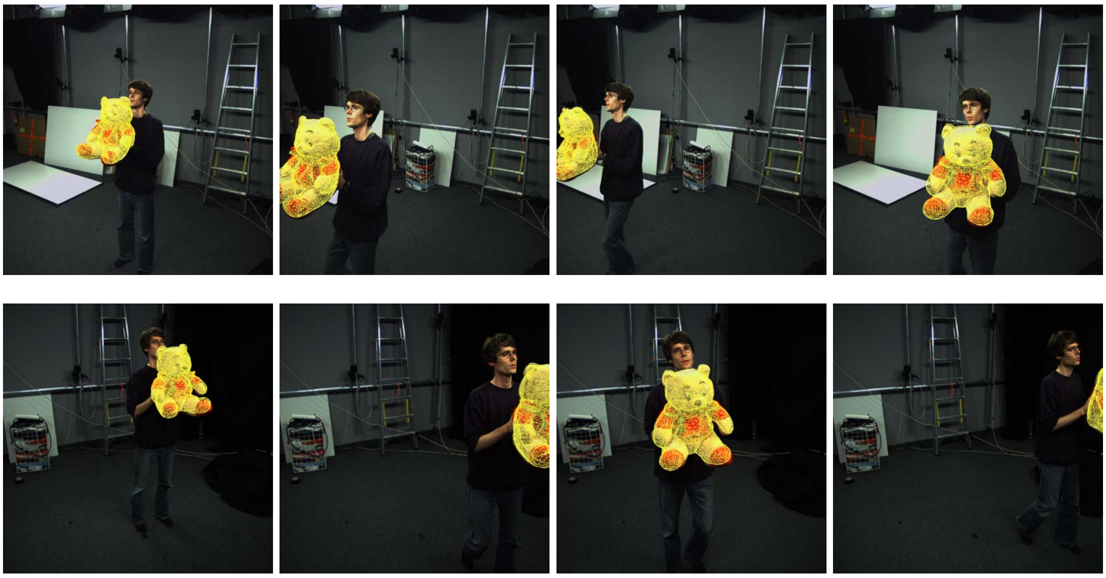


Figure 5.8: Tracking result for another rigid object. Two out of three camera views **(rows)** are shown for four frames.

to local minima that correspond to very bad poses. For this reason, motion based cues are needed to handle fast motion at low frame rates.

In Figure 5.6 displacements between successive frames are almost of the size of the object. Without a motion based prediction, region based pose estimation will fail to track this object. Surprisingly, although the object is homogeneous in large parts and there is a very high amount of noise added to the input images, multi-resolution optical flow is still able to capture its motion by means of its coarse-scale structure. The SIFT descriptor works fine as well, though there are only few SIFT regions on the puncher. When further decreasing the frame rate by skipping every second image, optical flow fails as the motion is larger than the tracked structure itself. For the SIFT tracker, the larger transformation is not a problem. The accumulation of inaccuracies is prevented by the region-based matching. Once the projected object model covers larger parts of the object region, the segmentation can robustly determine the exact location of the object contour, thanks to the homogeneity of the object region. As a consequence, it can correct errors of the motion-based prediction. This experiment shows that the system can deal with homogeneous objects, even if there are large displacements and substantial degradation by noise.

Figure 5.7 shows a slightly more difficult sequence, which we used to quantitatively determine the sensitivity to noise in the input images. We added increasing amounts of noise to the images and observed the frame number when tracking failed. The results are shown in Table 5.1. Without additional noise, the combined system can track the sequence completely. The tracking fails earlier in the sequence when the amount of noise is increased. Using the combined system, successful tracking is possible for a larger number of frames.

We performed two further experiments with quantitative results, as depicted in Figure 5.9 and Figure 5.10. Ground truth has been provided by placing the tracked objects on a turntable and reading the true pose from the turntable controller. The tracking curves reveal a very accurate tracking of the objects. In case of the tea pot, the average error is only 2.3 degree. The error increased to 4.6 degree replacing 50% of the pixel in the input images by uniform noise. In case

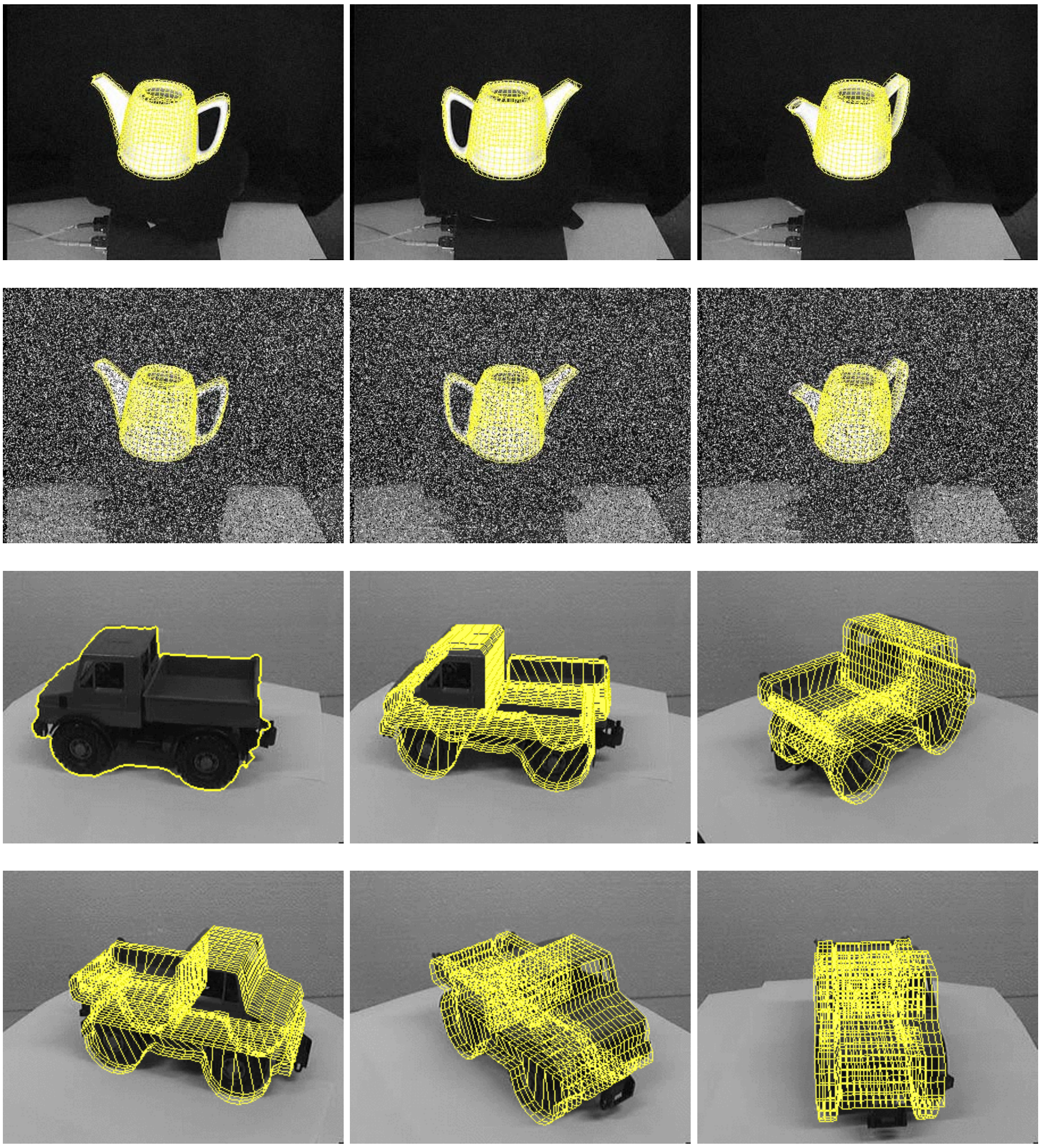
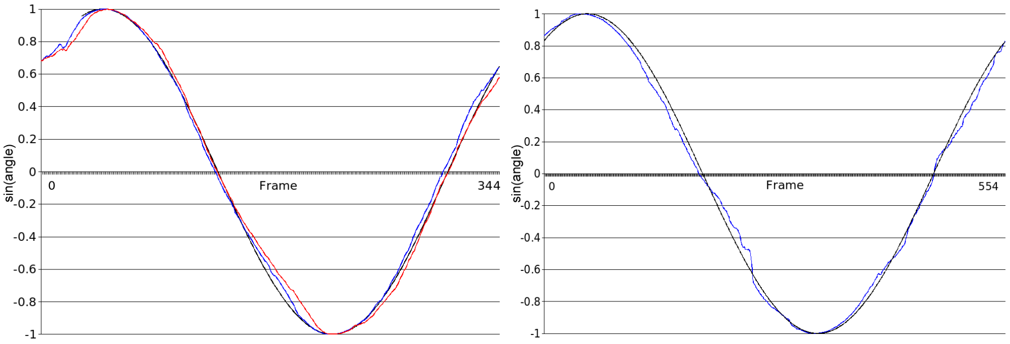


Figure 5.9: **Top row:** Tracking results of a tea pot on a turntable. **Second row:** Tracking results with 50% of the pixels in the input image replaced by a uniformly distributed random value. **Bottom:** Input image with estimated contour and tracking results of a toy car on a turntable. A quantitative error analysis is given in Figure 5.10.



(a) (b)

Figure 5.10: Quantitative error analysis of the sequences shown in Figure 5.9. **From left to right: a)** Tea pot. Comparison of the estimated pose *(blue)* versus the true motion *(black)*. The red curve shows the result on the noisy input images. **b)** Toy car. Comparison of the estimated pose *(blue)* versus the true motion *(black)*.

of the car, the average error is 2 degree.

Human motion tracking

In another set of experiments, we applied the system to the tracking of articulated objects, in particular to human motion tracking. Besides the global rigid motion, the joint angles of the body model represent further degrees of freedom that have to be estimated.

Due to the relatively small size and fast motion of limbs, it is very likely that region-based tracking gets stuck in local optima and tracking fails. Hence, the predicted pose due to optical flow and SIFT matches is particularly important for human motion tracking. This is demonstrated by the experiment in Figure 5.11 where the upper body of a person waving his arms is tracked. Without a good pose prediction, the arm movement is clearly underestimated as the contour extraction gets stuck in a local optimum. Optical flow and SIFT together allow for good predictions. SIFT alone is not sufficient since the number of keypoints is often too small for a unique estimate. Provided a good prediction, the region-based cues ensure a precise final pose estimate without accumulating the errors from motion-based tracking.

The experiment in Figure 5.12 shows the outcome of a full-body outdoor running sequence. The body model has 26 degrees of freedom and the image data was captured with four Basler gray-scale cameras and 120 frames per second. Ground-truth data was obtained for this sequence through parallel tracking of the person with a marker-based system. Bad marker correspondences have been corrected manually.

Thanks to combined cues, even fast motion can be tracked, as illustrated in Figure 5.14. The image in the top left corner depicts the start pose. The second image shows the predicted motion in the next frame using optical flow. The third image shows the tracked SIFT-features. Due to the black body suit, not enough features are detected to allow for a proper prediction using SIFT tracking alone. Tracking fails even with regularized equations since limb movements are not properly predicted. The left and center image in the bottom row depict the outcome of the combined optical flow and SIFT tracker. It is superior to the results of the separate motion predictors. The estimate is further refined when region-based tracking is involved. Compare the right hand of the person, for instance.

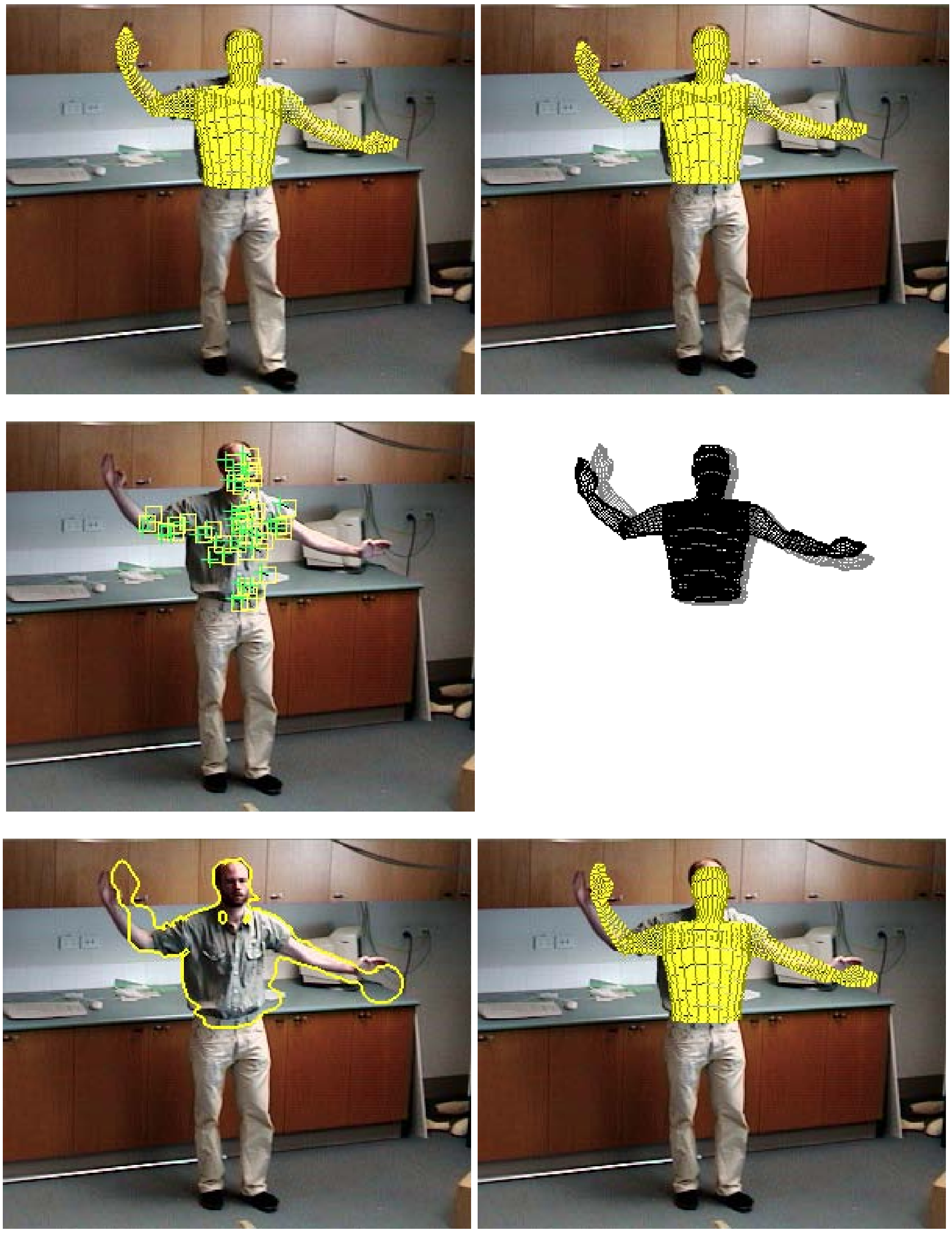


Figure 5.11: Combining motion- and region-based tracking allows capturing fast upper body motion. **Top row:** Initialization with the pose from the previous frame (left), and the estimated pose in the new frame when combining all available cues (right). **Middle row:** Matched SIFT keypoints (left). Yellow rectangles indicate keypoints in the previous frame, green crosses keypoints in the new frame. In this frame, successfully matched keypoints are available at the main body but missing at the hands. Right: motion prediction by optical flow and SIFT. **Bottom row:** The same situation with region-based cues only. Lacking a sufficiently close initialization, contour extraction fails (left) and leads to an inaccurate pose estimation (right).

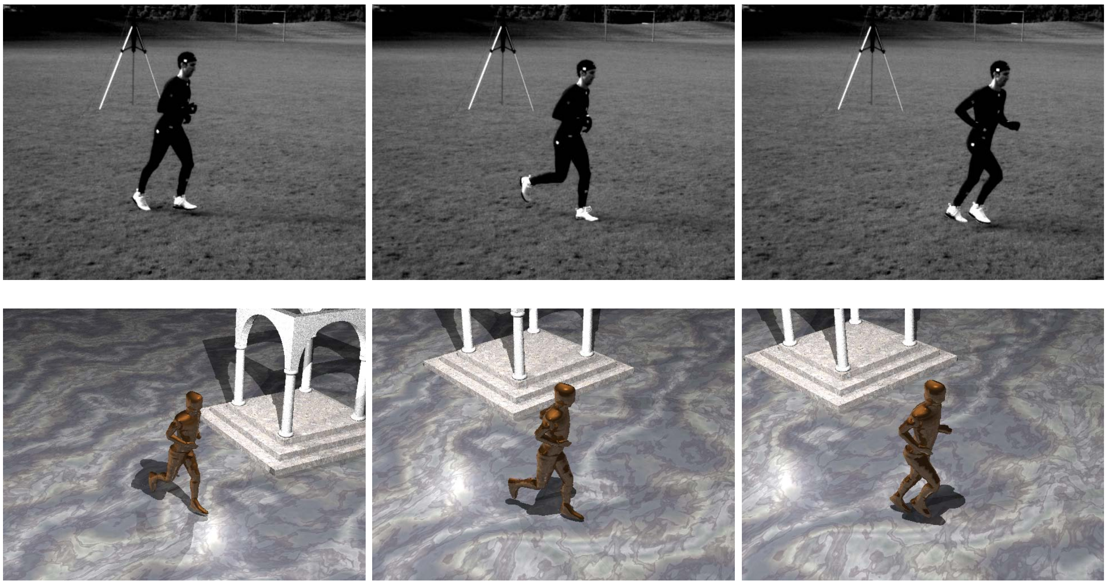


Figure 5.12: Full body tracking in a sequence with ground truth data. **Top row:** Input frames from one out of four camera views. **Bottom row:** Synthesized images from the tracked 3D pose. A different viewpoint than in the input images is depicted. Further results are shown in Figures 5.13, 5.14, 5.15, and Table 5.2.

	flow only	region only	region+SIFT	region+flow	region+SIFT+flow
120 fps	- (30)	4.29 ± 3.42	4.35 ± 3.31	4.42 ± 3.38	4.46 ± 3.38
40 fps	- (30)	- (165)	4.35 ± 3.34	4.31 ± 3.43	4.29 ± 3.38
30 fps	- (33)	- (118)	4.86 ± 4.29	4.47 ± 3.94	4.73 ± 3.99
24 fps	- (21)	- (33)	- (33)	- (25)	5.83 ± 4.91

Table 5.2: Comparison of cue combinations at various frame rates corresponding to the experiment in Figure 5.12. The table shows the average error of the knee and elbow joint angles over all 180 frames in degrees. The second value indicates the standard deviation. Tracking failures are marked by '-' and a number that indicates the frame where tracking failed (one bad limb).

Table 5.2 shows quantitative results for the most interesting cue combinations. Clearly, the combination of correspondences improves the robustness of tracking when the frame rate is reduced. When tracking is successful, the results are very precise with average errors of about 5 degrees. Figure 5.13 depicts corresponding tracking curves for the elbow and knee angles. The system with the combined cues is close to the ground truth even when the frame rate is small, whereas tracking with the purely region-based system fails (red curves).

Tracking with purely motion-based cues always fails due to accumulation of errors. Figure 5.15 illustrates the corresponding drift. Although the estimated optical flow is extremely precise, as indicated by the successful tracking of the torso over 150 frames, even smallest errors accumulate over time especially at limbs with few correspondences. Such drift can be reduced by region-based correspondences, which are based on matching the image directly to the model and are less sensitive to small errors in previous frames.

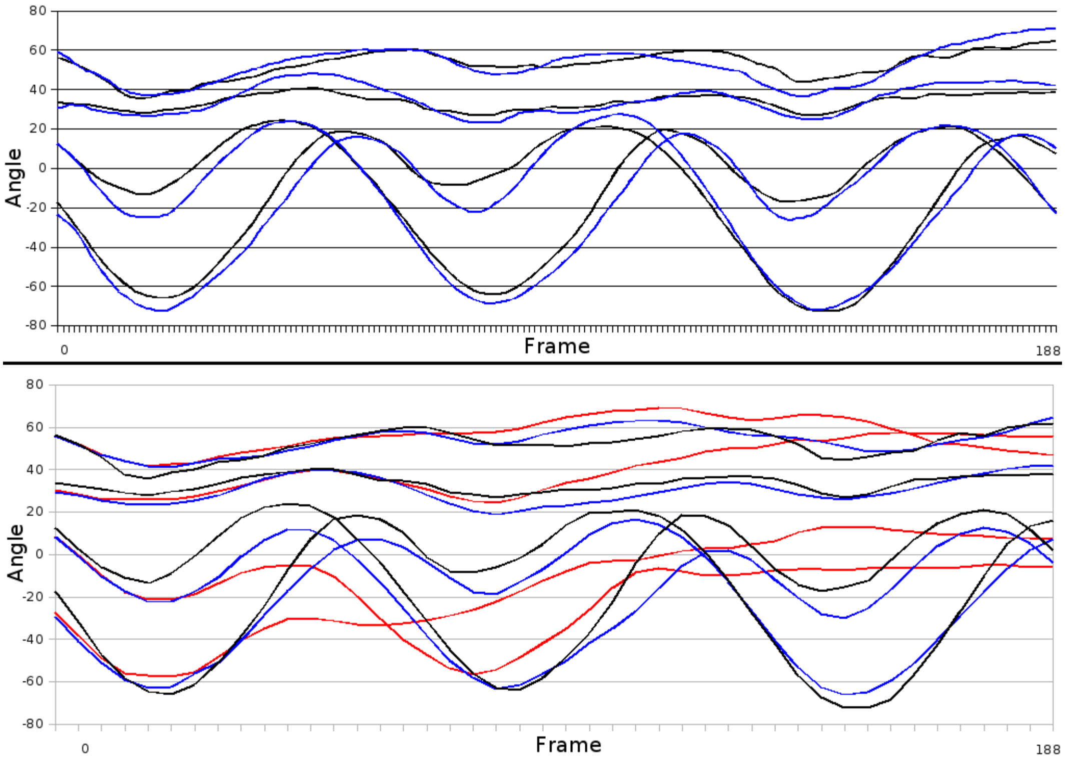


Figure 5.13: Tracking diagram for the sequence in Figure 5.12. The curves show the angles of the two elbow and the two knee joints. **Top:** Comparison of the proposed system (blue) to the ground truth (black). **Bottom:** Comparison of the combined system (blue) to the purely region-based system (red) for a reduced frame rate of 24fps. The black curve shows again the ground truth. The tracking failure of the single-cue system is clearly visible. See Table 5.2 for average errors.

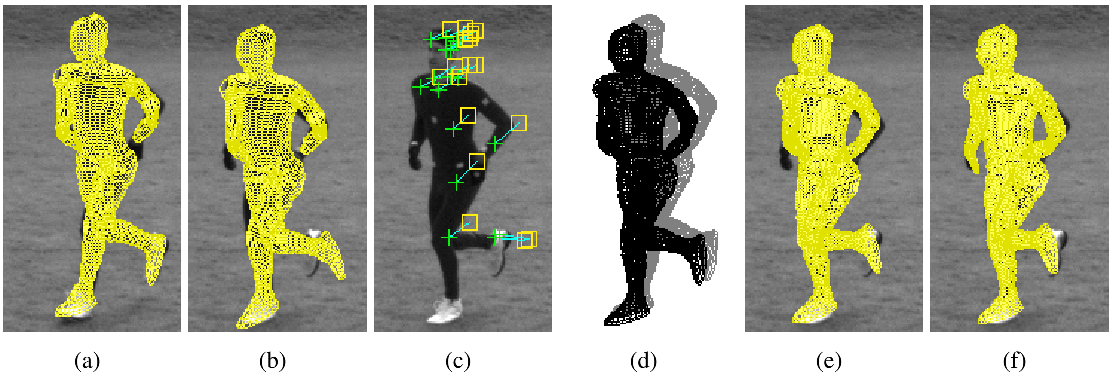


Figure 5.14: Combining motion- and region-based tracking allows capturing the fast motion of a jogging person. **From left to right: a)** Object pose at frame 1. **b)** Pose at frame 2 estimated from optical flow correspondences only. **c)** Tracked SIFT features: not enough features are located to ensure a proper prediction. **d)** Estimated prediction at frame 2 using combined optical flow and SIFT information. Gray: pose in frame 1. Black: prediction for frame 2. **e)** Prediction from (d) overlaid with the image. The outcome is much better than the result in (b). **f)** Final outcome for motion- and region-based tracking.

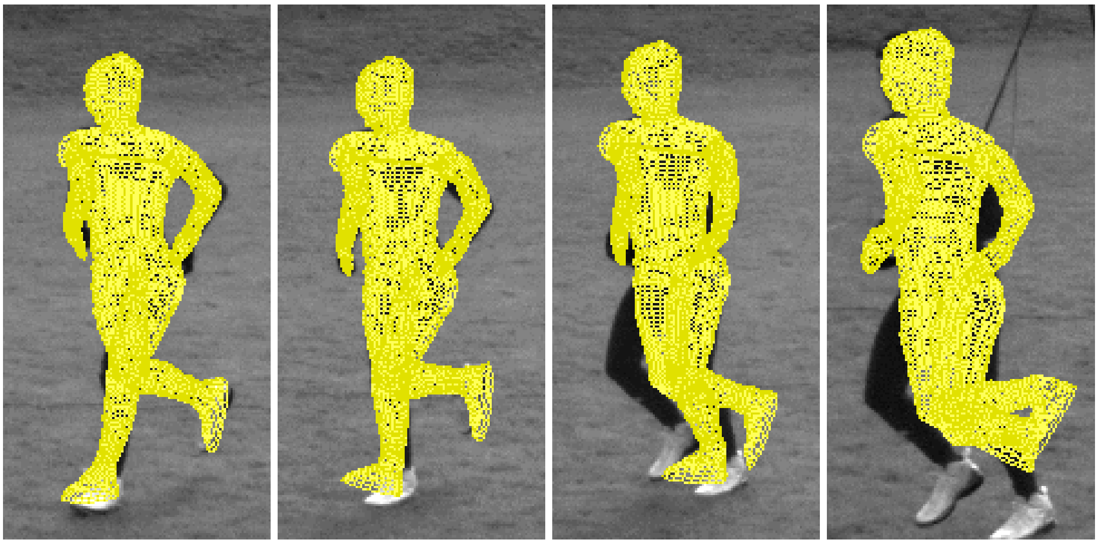


Figure 5.15: Illustration of the drift when only flow-based correspondences are used for tracking. **From left to right:** Result at frames 1, 10, 30, and 150. The optical flow yields good results for the first frames, which indicates its suitability for predicting the pose in successive frames. However, errors accumulate over time and are the reason for tracking failures of the limbs.

The computation time for tracking the full body model with four camera views was around 4 minutes per frame. Tracking the upper body model with two camera views took approximately 80 seconds per frame. The rather large computation time is mainly due to the iterative region-based tracking and the involved local region statistics including a texture feature space.

5.3.6 Summary

In this section, the combination of surface-region matching, optical flow, and SIFT tracking has been proposed for 3D motion capture of rigid and articulated objects. The system is designed in a way that all involved cues can incorporate their strong aspects, while weaknesses are sought to be suppressed. This is achieved as the system adaptively weights cues according to their reliability. This results in a very generally applicable tracking system. We have demonstrated this by a number of experiments in very different scenarios, where we obtained stable tracking results although the parameters of the system were not manually adapted when changing the scene. In particular, the system is able to capture large transformations, it can track textured as well as homogeneous objects, and it can deal with noise and partial occlusions. Furthermore, we have demonstrated that the system can be applied to human motion tracking, even when prior knowledge about typical human movements is missing.

5.4 Analysis-by-Synthesis Tracking

Although the combination of region- and motion-based cues performs well, it assumes that the initial pose is known or well approximated as demonstrated in Figure 5.5. It also implies that the system cannot recover from a significant tracking error. A natural approach to overcome these limitations is to exploit available prior knowledge of the object. So far only the 3D shape was used (Figures 5.16 a) and b)) and combined with motion cues that match the 2D image texture between successive frames (Section 5.3). However, the 3D surface model can be extended by acquiring the surface texture which usually resolves ambiguities for objects with symmetric shapes as shown in Figure 5.16. Since the estimating process relies on correspondences between some 2D features in the images and their counterparts on the 3D model, the additional information allows extracting more reliable correspondences that makes the pose estimation more robust. Moreover, it allows detecting the initial pose automatically.

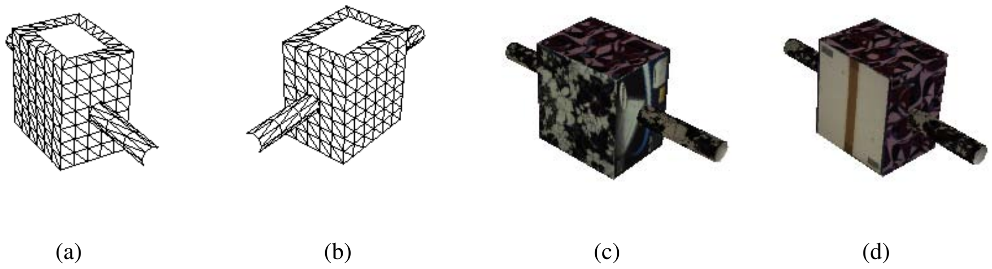


Figure 5.16: 3D mesh and rendered textured model used for tracking.

5.4.1 Overview

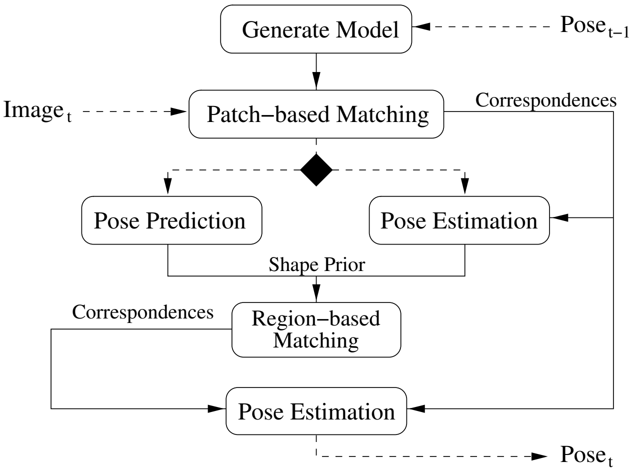


Figure 5.17: Correspondences extracted by matching patches between the synthesized image and the original image and correspondences between the contour of the projected 3D model and the contour obtained by segmentation are used for pose estimation. If not enough keypoints are detected by patch-based matching, an autoregression is performed to predict the pose for the next frame.

The approach for pose estimation is illustrated by the flow chart in Figure 5.17. Knowing the pose of the object for frame $t-1$, we generate a 3D textured model in the same world coordinate system used for the calibration of the cameras. Synthesized images of the model are obtained by projecting the model onto the image plane according to the calibration matrix for each camera. In a second step, the patch-based features, namely PCA-SIFT [KS04], are extracted from the synthesized images and from the new images of frame t. The features are used for establishing correspondences between the 3D model and the 2D images for each view. For estimating the pose, we use the least-squares approach of the previous sections. If not enough correspondences are extracted by PCA-SIFT, the pose is predicted by an autoregression. Finally, region-based matching is performed where the predicted pose is used as shape prior (Section 5.2). The pose for frame t is then estimated from correspondences obtained by patch-based and region-based matching.

5.4.2 Analysis-by-Synthesis

Synthesis

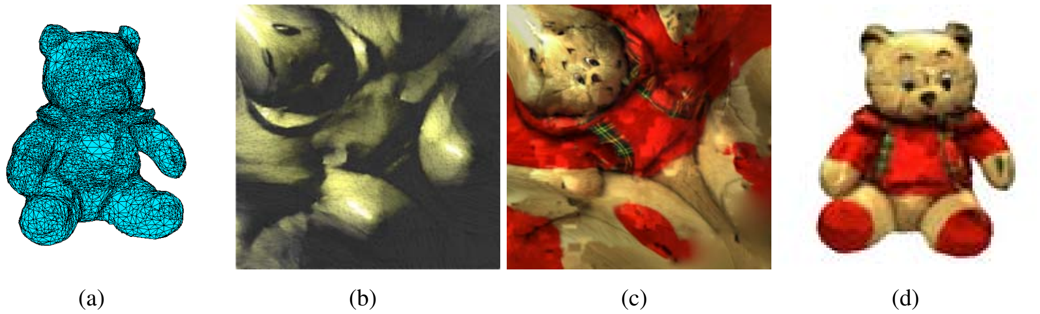


Figure 5.18: **From left to right: a)** Triangulation. **b)** Parameterization. **c)** Texture map. **d)** Textured model.

For synthesizing images, the texture of the model needs to be acquired. As in the previous sections, we assume that a triangulated 3D model is available as shown in Figure 5.18 a), which might be obtained by any 3D acquisition or modeling technique as discussed in Section 1.6. Since the image domain is only 2D, a parameterization of the 3D surface is necessary. For this purpose, the mesh is manually cut and mapped to a square where unavoidable distortions of the triangles are reduced by a quasi-harmonic map [ZRS05], see Figure 5.18 b). The images for the texture can be either acquired directly from the tracking sequence or in a preprocessing step by capturing the object from different viewpoints with a calibrated camera. Having images of the object from different views, the silhouettes are extracted by background subtraction and the pose of the model is estimated from the silhouettes [GRS07]. This step is described in-depth in Section 6.2. The object is then mapped onto the squared texture map for each camera and the visible parts are fused by multiresolution splines [BA83] in order to remove seams between triangles from different views. Invisible triangles are filled up by linear interpolation. A resulting texture map is shown in Figure 5.18 c), which can be used to render the model in any pose. The texture acquisition for articulated models is the same as for rigid models.
Since we do not require that the textures are extracted from the tracking sequences, the modeling process is done only once and the model can be reused for any sequence provided that the texture remains unchanged. In order to render the 3D model in the same coordinate system as used for camera calibration, the calibration matrices are converted to the modelview and projection matrix representation of OpenGL. Since OpenGL cannot handle lens distortions directly, the image sequences are undistorted beforehand. However, the step could also be efficiently implemented by a look-up table. In a preprocessing step, PCA-SIFT is trained for the object by building the

patch eigenspace from the object textures. Moreover, we render some initial views of the 3D model by rotating and store the extracted keypoints, strictly speaking the PCA-SIFT descriptors of the keypoints, with the corresponding RBM. From the data, our system automatically detects the pose in the first frame.

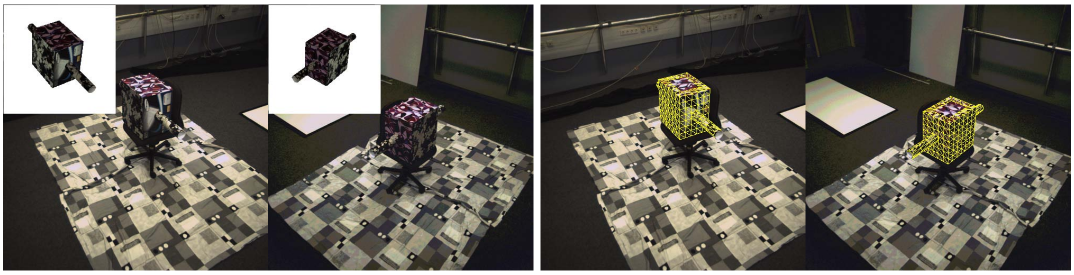

(a) (b)

Figure 5.19: Initialization. **From left to right: a)** Both camera views of the first frame. Best initial view for initialization is shown in top left corner. **b)** Estimated pose after initialization.

Matching

Since lighting conditions between the object and its textured model are inhomogeneous, we use local descriptors that provide robust matching under changes in viewpoint and illumination. A comparison of local descriptors [MS03] revealed that SIFT [Low04], PCA-SIFT [KS04], and GLOH [MS03] perform best. The descriptors build a distinctive representation of a so-called keypoint in an image from a patch of pixels in its neighborhood. The keypoints are localized by an interest point detector. We use the detector proposed by Lowe [Low99] based on local 3D extrema in the scale-space pyramid built with difference-of-Gaussian filters. It has the advantage that it runs faster than other detectors [MS04] like the slower Harris-Affine detector [MS02]. The DoG representation, however, is not affine invariant. Hence, we cannot use GLOH that requires an affine-invariant detector. Therefore, we use PCA-SIFT that reduces the dimension of the descriptor by principal component analysis. This speeds up the matching process and produces less outliers than SIFT but also fewer correspondences.

After the 3D model is rendered and projected onto the image plane for each camera view, the keypoints are extracted by PCA-SIFT. The keypoints are also extracted from the captured images. The effort is reduced by bounding cubes for each component of the 3D model. Projecting the corners of the cubes provides a 2D bounding box for each image. Since we track an object, we can assume that the object is near the bounding box except for the first frame. Hence, the detector is only performed on a subimage. 2D-2D correspondences are then established by nearest neighbor distance ratio matching [MS03] with the additional constraint that two different located points cannot correspond to points with the same position. Since the set of correspondences contains outliers, the rudest mismatches are removed by discarding correspondences with a Euclidean distance that exceeds the average by a multiple.

The 3D coordinate X of a 2D point x in the projected image plane of the model is obtained as follows: Each 2D point is inside or on the border of a projected triangle of the 3D mesh with vertices v_1, v_2, and v_3. The point can be expressed by barycentric coordinates, i.e., $x = \sum_i \alpha_i \, v_i$. Assuming an affine transformation, the 3D point is then given by $X = \sum_i \alpha_i \, V_i$ where V_i are the 3D vertices of the projections v_i. This gives a better estimate for X than

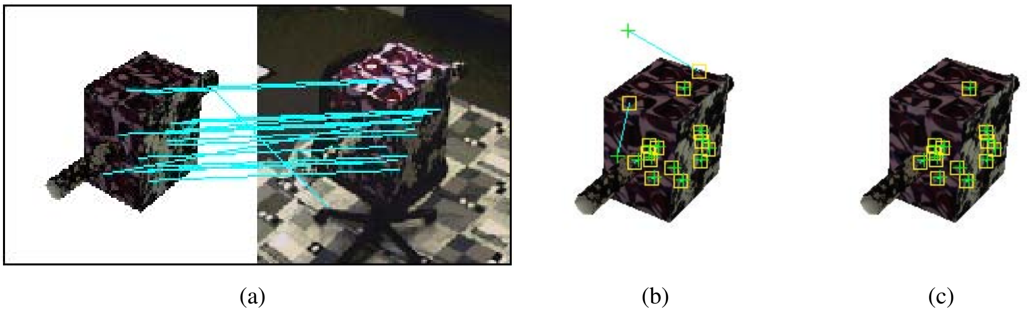

(a) (b) (c)

Figure 5.20: **From left to right:** **a)** Correspondences between projected model and image. **b)** Displaying the points of the projected model *(yellow squares)* corresponding to points in the image *(green crosses)*. Two outliers are in the set of correspondences. **c)** After filtering only the outliers are removed.

taking the nearest projected vertex as in Section 5.3.2. The corresponding triangle for a point can be efficiently determined by a look-up table containing the color index and vertices for each triangle. Afterwards, the pose is estimated from the resulting 2D-3D correspondences. In a second filtering process, the new 3D coordinates from the estimated pose are projected back and the last outliers are removed by thresholding the Euclidean distance between the 2D correspondences and the reprojected counterparts.

During initialization, the keypoints from the images are matched with the keypoints extracted from the initial views beforehand. According to the number of matches, a best initial view is selected and the pose is estimated from the obtained correspondences as shown in Figure 5.19.

Prediction

It is not straightforward to derive a formula for the velocity of a rigid body whose motion is given by $g(t)$, a curve parameterized by time t in $SE(3)$, since $SE(3)$ is not Euclidean. In particular, $\dot{g} \notin SE(3)$ and $\dot{g} \notin se(3)$. But by representing a rigid body motion as a screw action, the spatial velocity can be represented by the twist of the screw, see [MLS94] for details. This allows for motion interpolation, damping, and prediction.

For the pose prediction, an autoregression is employed that takes the global rigid body motions P_i of the last N frames into account. For this purpose, we use a set of twists $\xi_i = \log(P_i P_{i-1}^{-1})$ representing the relative motions and we make use of the adjoint transformation to represent a screw motion with respect to another coordinate system: If $\xi \in se(3)$ is a twist given in a coordinate frame A, then for any $G \in SE(3)$ which transforms a coordinate frame A to B, is $G\hat{\xi}G^{-1}$ a twist with the twist coordinates given in the coordinate frame B, see [MLS94] for details. The mapping $\hat{\xi} \longmapsto G\hat{\xi}G^{-1}$ is called the adjoint transformation associated with G.

Given a set of world positions and orientations P_i, see Figure 5.21, the twists ξ_i can be used to express the motion as local transformation in the current coordinate system M_1: Let $\xi_1 = \log(P_2 P_1^{-1})$ be the twist representing the relative motion from P_1 to P_2. This transformation can be expressed as local transformation in the current coordinate system M_1 by the adjoint transformation associated with $G = M_1 P_1^{-1}$. The new twist is then given by $\hat{\xi}_1' = G\hat{\xi}_1 G^{-1}$. The advantage of the twist representation is now that the twists can be scaled by a factor $0 \leq \lambda_i \leq 1$ to damp the local rigid body motion, i.e., $\hat{\xi}_1' = G\lambda_1\hat{\xi}_1 G^{-1}$. For given λ_i such that $\sum_i \lambda_i = 1$,

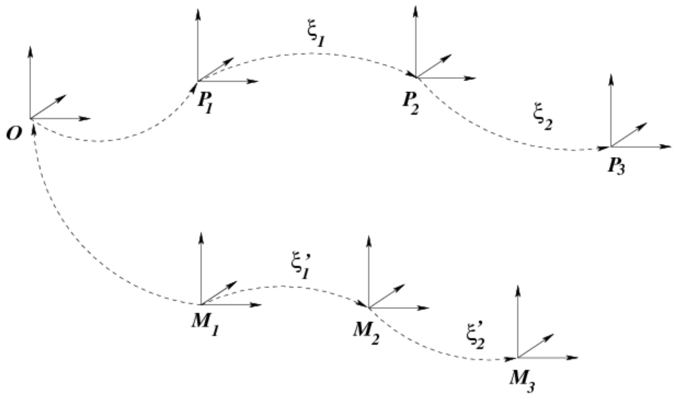


Figure 5.21: Transformation of rigid body motions from prior data P_i in a current world coordinate system M_i. A proper scaling of the twists results in a proper damping.

the predicted pose is obtained by the rigid body transformation

$$\exp(\hat{\xi}'_N)\exp(\hat{\xi}'_{N-1})\dots\exp(\hat{\xi}'_1). \tag{5.20}$$

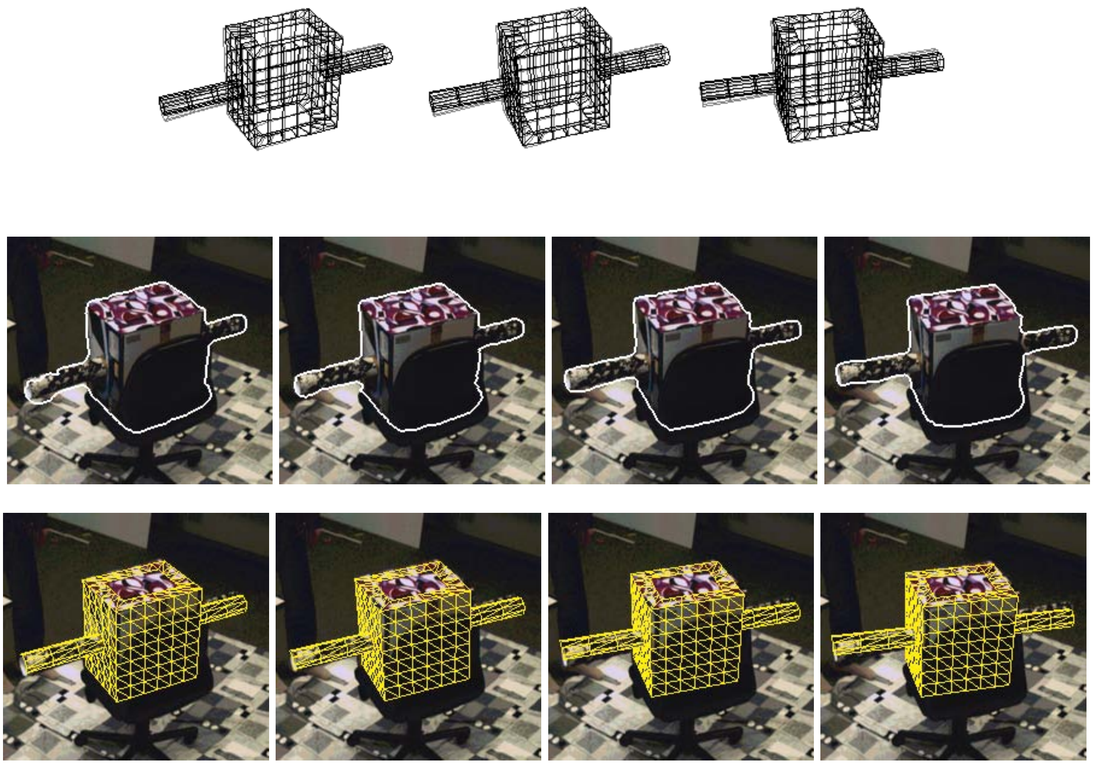
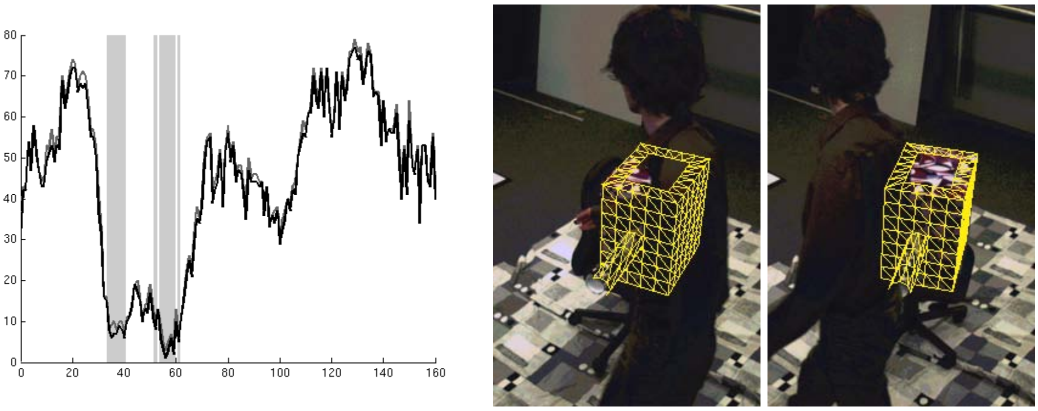


Figure 5.22: 4 successive frames of a rotation sequence (only one view is shown). **Top row:** Pose is predicted by autoregression for lack of PCA-SIFT matches. *Black:* Predicted pose. *Gray:* Previous pose. **Middle row:** Contour extracted by segmentation. **Bottom row:** Estimated pose.

Fusion

Although the segmentation as previously described is quite robust to clutter, shadows, reflections, and noise, a good shape prior is essential for tracking since both matching between the contours and the segmentation itself is prone to local optima. The predicted pose by an autoregression usually provides a better shape prior than the estimated pose in the previous frame. In situations, however, where the object region and the background region are difficult to distinguish, the error of the segmentation and the error of the prediction are accumulating after some time. The shortcoming is compensated by PCA-SIFT, but it is also clear that usually not enough keypoints are available in each frame. Hence, the correspondences from contour matching and from descriptor matching are added to one linear system for the pose estimation. Since the contour provides more correspondences, the sums in (5.3) and (5.5) are replaced by weighted sums as in Section 5.3.4. Let n_{C} denote the number of contour-based correspondences and w_i the weights. While the contour-based correspondences are taken as reference and all of them are assigned the weight $w_{\mathrm{C}} = 1$, PCA-SIFT correspondences are all weighted by $w_{\mathrm{PCA-SIFT}} = 0.2 \cdot n_{\mathrm{C}}$.



(a) (b) (c)

Figure 5.23: Rotation sequence with a moving person. **From left to right: a)** Number of matches from PCA-SIFT *(dark gray)*. After filtering the number of matches is only slightly reduced *(black)*. When the number is below a threshold, the pose is predicted by an autoregression *(gray bars)*. **b, c)** The rotating box is occluded by a moving person.

5.4.3 Experiments

For evaluating the performance of the analysis-by-synthesis approach, the 3D textured model shown in Figure 5.16 is used. The textures have been acquired from a video sequence that is not part of the evaluation sequences. The other sequences have been recorded by two synchronized and calibrated cameras under different lighting conditions. Although the size of the images is 502×502 pixels, the object is only about 100×100 pixels. The initial position has been automatically detected for each sequence as shown in Figure 5.19.

The tracked object is partially covered with two dissimilar fabrics and the printed side reflects the light. It is placed on a chair that occludes the back of the object. The background is rich textured and non-static. Shadows, dark patterns on the texture, and the black chair make contour extraction difficult even for the human eye. Furthermore, a person moves and occludes the object. These conditions make great demands on the method for pose estimation.

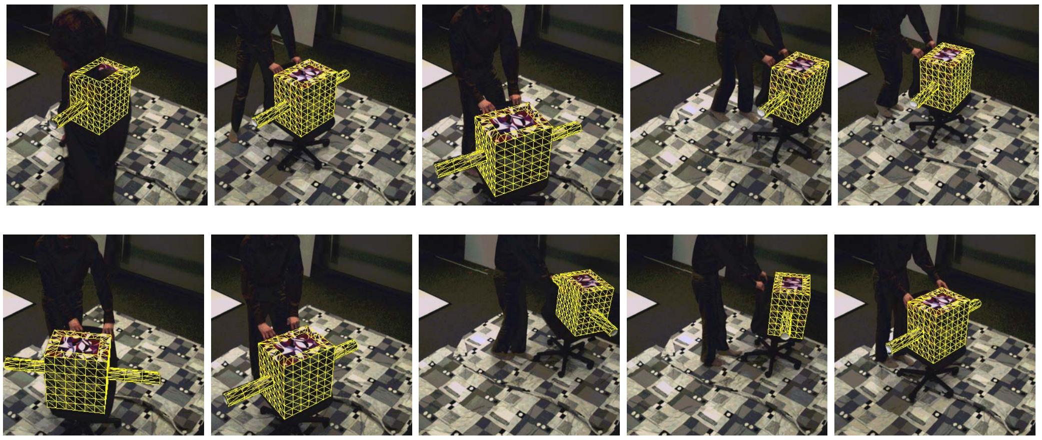
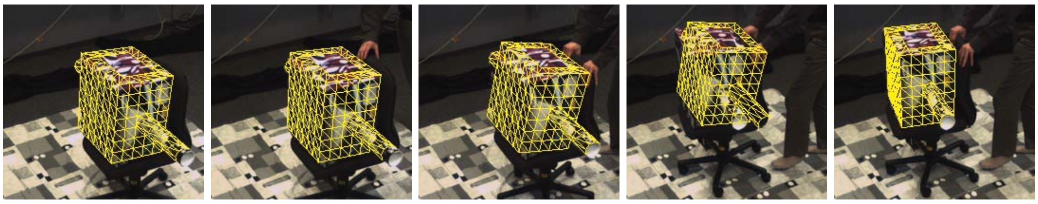


Figure 5.24: Pose estimates for 10 of 570 frames. The sequence contains several difficulties for tracking: a rich textured and non-static background, shadows, occlusions, and other moving objects. Only one camera view is shown.

In the first sequence, the chair with the object rotates clockwise. When the back of the chair occludes the object, there are not enough distinctive interest points for pose estimation. Therefore, the pose is predicted by an autoregression for the next frame as shown in Figure 5.22. Due to the shape prior, the segmentation is robust to the occlusion such that the estimates are still accurate. The number of matches from PCA-SIFT with respect to time is plotted in Figure 5.23 a). During the sequence, the object rotates counterclockwise while the person orbits the object clockwise. As we can see from the diagram, PCA-SIFT produces only few outliers that are removed after the filtering. The gray bars in the diagram indicate the frames where an autoregression was performed. Since the number of matches range from 1 to 77, it is clear that an approach based only on the descriptors would fail in this situation.

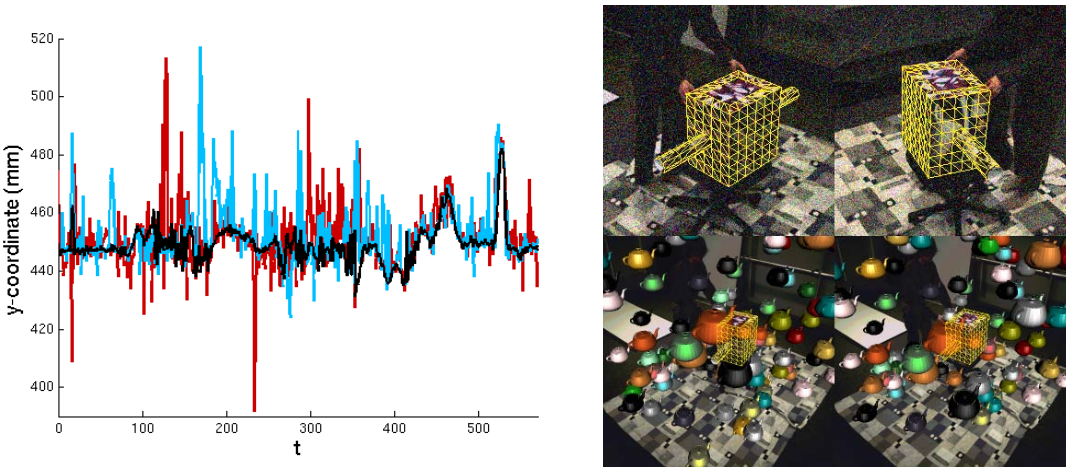


(a) (b)

Figure 5.25: Comparison with a contour-based method. **From left to right: a)** Pose estimates for frames 5, 50, 90, 110. **b)** Result of our method at frame 110.

Pose estimates for a third sequence including rotations and translations of the object are shown in Figure 5.24. When only the contour is used, the pose estimation is erroneous since both segmentation and contour matching are distracted by local optima, see Figure 5.25 a). For comparison, the result of our method is also given (Figure 5.25 b)).

Finally, noise on the sequence is simulated in order to obtain a quantitative error analysis. Since the object is placed on the chair, the y-coordinate of the pose is approximately constant. During the sequence, however, the object shifts slightly on the chair. The peak at frame 527 in the diagram of Figure 5.26 a) is caused by a relocation of the object. For one sequence, we added Gaussian noise with standard deviation 35 to each color channel of a pixel. Another sequence was disturbed by 80 teapots that were rendered in the 3D space of the tracked object. The teapots drop from the sky where the start positions, material properties, and velocities are random. Regarding the result for the undistorted sequence as some kind of ground truth, the diagram in Figure 5.26 a) shows the robustness of our approach. While an autoregression was performed only twice for the unmodified sequence and the average number of filtered matches per frame from PCA-SIFT was 50.9, the numbers fell down to 27.9 and 13.1 for the teapots sequence with 132 predictions and the noisy sequence with 361 predictions.



(a)

(b)

Figure 5.26: **From left to right: a)** Quantitative error analysis for a sequence with disturbances. *Black:* Undisturbed sequence. *Red:* Gaussian noise with standard deviation 35. *Blue:* 80 teapots dropping from the sky with random start position, material properties, and velocity. **b)** *Top:* Stereo frame 527 of the noisy sequence (image details). *Bottom:* Two successive frames of the teapot sequence.

5.4.4 Summary

The proposed analysis-by-synthesis approach exploits the shape *and* the texture of the surface. In contrast to motion-based cues, it is robust to partial occlusions, can recover from errors, and supports an automatic initialization. The additional region-based cue allows tracking even when the number of features is very low, e.g. in the case of occlusion, without the need of an additional re-initialization after some frames, when enough features are again detected [LPF04]. Another variant of keypoint matching, which allows real-time initialization and tracking, classifies the best initial view by a random forest [LLF05]. Although these approaches work well for rigid objects as we have shown, they are not suitable for articulated objects since the large number of degrees of freedom requires a large number of keyframes. In addition, small body parts like hands cannot be estimated by features alone. In order to obtain a robust tracking system even for articulated objects, the ideas from Sections 5.2, 5.3, and 5.4 need to be combined in a unique framework as it is proposed in the following section.

5.5 Drift-free Tracking

Motion-based tracking approaches as in Section 5.3 rely on image features that are tracked over time but the accumulation of small errors results in a drift away from the target object. In this section, we address the drift problem for the challenging task of human motion capture and tracking in the presence of multiple moving objects where the error accumulation becomes even more problematic due to occlusions. To this end, we propose an analysis-by-synthesis framework for articulated models. It combines the complementary concepts of patch-based and region-based matching to track both structured and homogeneous body parts.

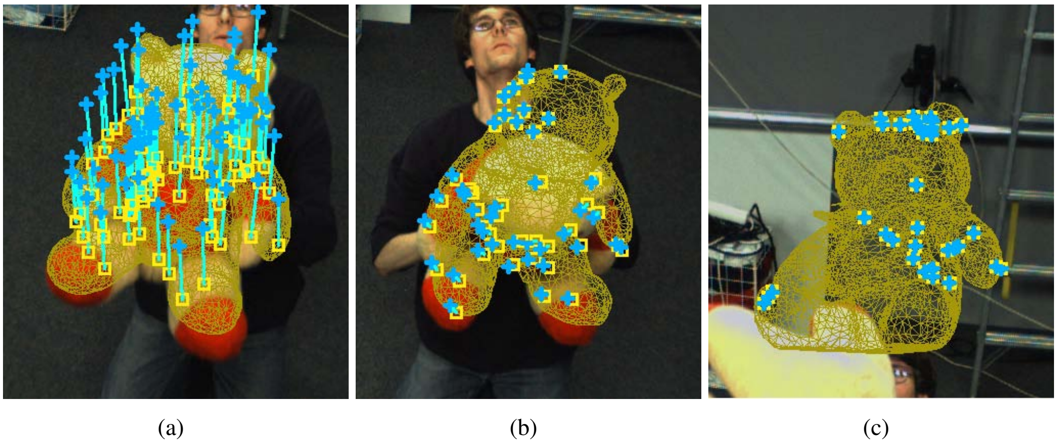


Figure 5.27: **Motivation.** When the pose of the target object *(projected mesh)* is known at the current frame, 2D correspondences between the current frame *(square)* and the next frame *(cross)* are often used to estimate the pose for the next frame. While this is sufficient at the beginning when the pose is well estimated **(a)**, small errors that accumulate over time result in a drift away from the target **(b)**. In the worst case, the object is completely lost **(c)**. The drift is even more problematic when occlusions occur, e.g. occlusions by other objects (Figure 5.30) or self-occlusions in the case of humans (Figure 5.34).

Image features can be tracked over time by flow-based methods [LK81, PBB$^+$06] (Section 5.3.1) or by patch-based 2D tracker like KLT [ST94] or interest point matching [MS03] (Section 5.3.2). Under the assumption that the pose is well estimated for the current frame, the 2D correspondences between the current frame and the next frame can be used to estimate the 3D pose for the next frame as illustrated in Figure 5.27 for a rigid object. The main drawback of these approaches is the error accumulation over time resulting in a drift away from the object. To overcome this limitation, the combination of multiple cues was proposed, e.g. optic flow and edges for face tracking [DM00] or optic flow and contour for rigid objects [BRCS06]. In [LRF93] an iterative analysis-by-synthesis approach was suggested for face tracking.

We go beyond the tracking of rigid objects and faces and propose a framework that combines the ideas of multi-cue integration and analysis-by-synthesis for the challenging task of human motion capture and tracking in the presence of multiple objects where drift becomes even more problematic due to occlusions as shown in Figures 5.30 and 5.34. To recover from errors and to detect occlusions, we propose the use of a synthesized image, which is generated with the predicted pose of the object and a static texture, as a reference image for each frame. For both prediction and correction by synthesis, patch-based matching is performed as outlined in Fig-

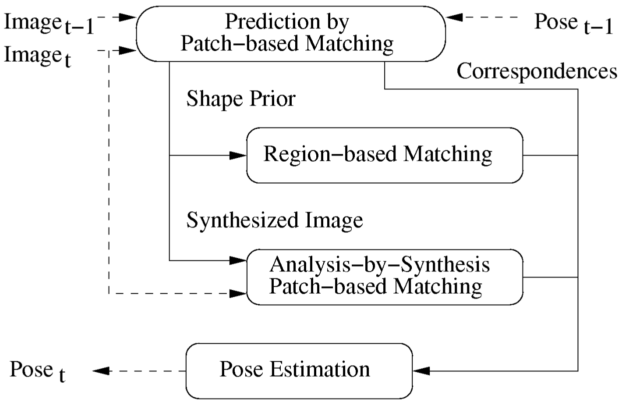


Figure 5.28: Having estimated the pose for time $t-1$, the pose for the next frame is predicted by matching patches between the images of frames $t-1$ and t. The predicted pose provides a shape prior for the region-based matching and defines the pose of the model for synthesis. The final pose for frame t is estimated from weighted correspondences emerging from the prediction, region-based matching, and analysis-by-synthesis, see also Figure 5.29.

ure 5.28. While the illumination properties between two successive frames are similar and therefore a large number of matches can be provided in the prediction step (see Figure 5.29), a static texture in the synthesis step provides correspondences that are not affected by error accumulation during tracking. Since the surfaces of human body parts are not always covered by patterns, which can be tracked well by patch-based matching, correspondences for homogeneous body parts are obtained by region-matching where the segmentation is improved by a shape prior from the predicted pose. In the following sections, we briefly recapitulate the cue extraction and the pose estimation for the sake of completeness, which has been described in-depth in the previous sections.

5.5.1 Cues

Region-based Matching

Region-based matching minimizes the difference between the projected surface of the model and the object region extracted in the image, see Figure 5.29 b). For this purpose, 2D-2D correspondences between the contour of the projected model and the segmented contour are established by a closest point algorithm [Zha94]. Since the projected points on the contour relate to 3D vertices of the mesh as shown in Figure 5.29 d), 3D-2D correspondences between the model and the image can be derived.

The silhouette of the object is extracted by a level-set segmentation that divides the image into fore- and background where the contour is given by the zero-line of a level-set function Φ. As proposed in [RBW07], the level-set function Φ is the minimum of the energy functional

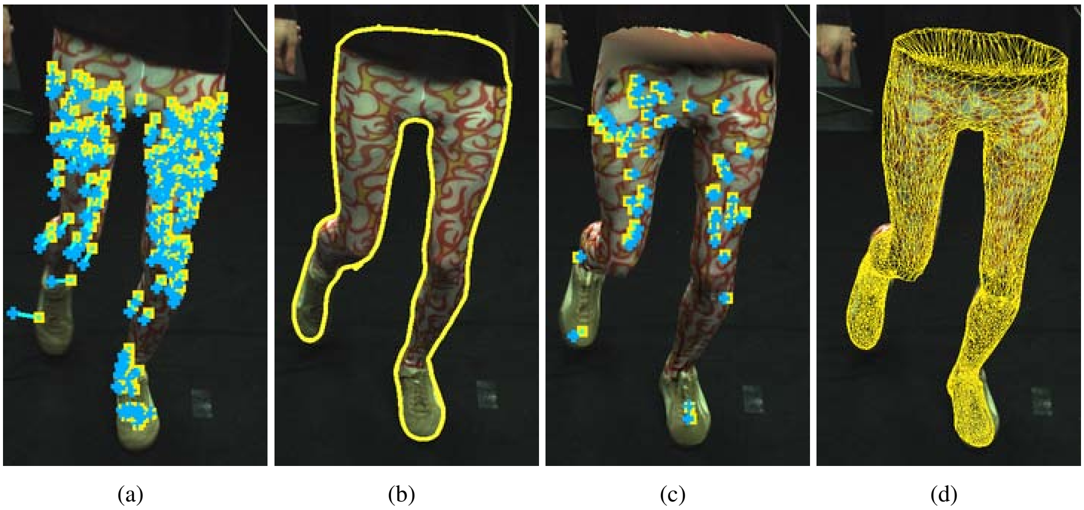


Figure 5.29: **From left to right:** **a)** Correspondences between current frame *(yellow square)* and next frame *(blue cross)*. **b)** The extracted contour for region-based matching also provides correspondences for homogeneous body parts like the right foot. **c)** Correspondences between the synthesized image and the original image. **d)** Projection of estimated pose.

$$E(\Phi, p_1, p_2, \chi) = \underbrace{-\int_\Omega H(\Phi) \ln p_1 + (1 - H(\Phi)) \ln p_2 \, dx + \vartheta \int_\Omega |\nabla H(\Phi)| \, dx}_{\text{segmentation}} + \underbrace{\lambda \int_\Omega (\Phi - \Phi_\chi)^2 \, dx}_{\text{shape distance}}, \tag{5.21}$$

where H is a regularized version of the step function, p_1 and p_2 are the densities of the fore- and background modeled by local Gaussian densities. While the first term maximizes the likelihood, the second term regulates the smoothness of the contour by parameter $\vartheta = 2$. The last term penalizes deviations from the projected surface of the predicted pose Φ_χ with $\lambda = 0.06$.

Patch-based Matching

Patch-based matching extracts correspondences between two successive frames for prediction and between the current image and a synthesized image for avoiding drift as outlined in Figure 5.28. The synthetic image is obtained by projecting the predicted textured model onto the current image as shown in Figure 5.29 c). For reducing the computation effort of the keypoint extraction [Low99], a region of interest is selected by determining the bounding box around the projection and adding fixed safety margins that compensate for the movement. To cope with the illumination differences between the synthetic and the current image, we apply PCA-SIFT [KS04] as local descriptor that is trained for the object by building the patch eigenspace from the object texture. 2D-2D correspondences are then established by nearest neighbor distance ratio matching [MS03] where the search is reduced to keypoints inside a local neighborhood, e.g. 100 × 100 pixels, to deal with repeating patterns on the surface. Since each 2D

keypoint x of the projected model is inside or on the border of a triangle with vertices v_1, v_2, and v_3, the 3D counterpart is approximated by $X = \sum_i \alpha_i V_i$ using barycentric coordinates $(\alpha_1, \alpha_2, \alpha_3)$.
The patch matching produces also outliers that need to be eliminated. In a first coarse filtering step, mismatches of the torso and each limb are removed by discarding 2D-2D correspondences with a Euclidean distance that exceed the average of the torso or limb by a multiple. After deriving the 3D-2D correspondences, the pose is estimated and the new 3D correspondences are projected back. By measuring the distance between the 2D correspondences and their reprojected counterparts, the remaining outliers are detected.
For the patch-based matching between two successive frames, only keypoints on the projected surface of the model are kept and the filtering thresholds for the limbs are given by predicting the largest 2D translation of the points of each limb. For this purpose, the joint configuration is predicted by an autoregression

$$\widehat{\Theta}_t = a_1\Theta_{t-1} + a_2\Theta_{t-2} + a_3\Theta_{t-3}, \tag{5.22}$$

where the coefficient vectors a_i are computed from a training sequence. The rigid body motion $\theta\xi$ is predicted as described in Section 5.4.2. By approximating each limb with a cuboid, the maximal translation can be efficiently calculated for each view. Since the projected surface depends on the previous estimated pose, parts of the correspondences might belong to the background as demonstrated in Figure 5.27. Hence, corresponding features of the torso or of a limb with the same location are deleted if the average is above a threshold. An inaccurate pose can also yield a wrong limb association of a keypoint when self-occlusions occur. The confidence of a correspondence is therefore significantly reduced if neighboring pixels of the keypoint belong to two unconnected limbs. Filtered 2D-2D correspondences are shown in Figures 5.29 a) and c).

5.5.2 Pose Estimation

For estimating the pose, we seek for the transformation $\chi = (\theta\xi, \Theta)$ that minimizes the error of extracted 3D-2D correspondences denoted by pairs (X_i, x_i) of homogeneous coordinates with weights w_i. According to Section 5.1, this is modeled by the weighted least squares problem:

$$\operatorname*{argmin}_{\chi} \frac{1}{2} \sum_i w_i \left\| \Pi \left(\exp\left(\theta\hat{\xi}\right) \prod_{j=1}^{n_{k_i}} \exp\left(\theta_{\iota_{k_i}(j)} \hat{\xi}_{\iota_{k_i}(j)}\right) X_i \right) \times n_i - m_i \right\|_2^2 . \tag{5.23}$$

The sought transformation is obtained iteratively by solving for each iteration step the linear system

$$\begin{pmatrix} w_1 A_1 \\ w_2 A_2 \\ \cdot \\ w_r A_r \end{pmatrix} \chi^T = \begin{pmatrix} w_1 b_1 \\ w_2 b_2 \\ \cdot \\ w_r b_r \end{pmatrix} \tag{5.24}$$

which is derived by linearizing the exponentials similar to Equation (5.6).

Tracking

After the prediction, the final pose is estimated from correspondences that are extracted by patch-based and region-based matching as outlined in Figure 5.28. Since the number of correspondences from the contour varies according to scale, shape, and triangulation of the object, we weight the summands in Equation (5.23) such that the influence between patches and silhouette is independent of the model.

We denote the set of correspondences from the original images, the synthetic image, and the contour by C_o, C_s, and C_c, respectively. The invariance is obtained by setting the weights for the equations for C_o and C_s in relation to C_c:

$$w_o = \alpha \frac{|C_c|}{|C_o|}, \quad w_c = 1, \quad w_s = \beta \, w_o. \tag{5.25}$$

While the influence of the image-based patches and the contour is controlled by α independent of the number of correspondences, the weight w_s reflects the confidence in the matched patches between the synthesized and original image that increases with the number of matches $|C_s|$ relative to $|C_o|$. Since illumination differences between the two images entail that $|C_s|$ is less than $|C_o|$, β compensates for the difference, cf. Figures 5.29 a) and c). For the experiments, we set $\alpha = 0.2$ and $\beta = 10.0$.

To avoid that the system of linear equations (5.24) becomes under-determined for small and homogeneous body parts, we add a low weighted regularization term that penalizes the deviation of a joint angle θ_j from the predicted pose $\widehat{\Theta}$ (5.22):

$$\gamma\theta_j = \gamma \left(\widehat{\theta}_j - \tilde{\theta}_j \right) \tag{5.26}$$

for each joint j, where $\tilde{\theta}_j$ is the previously estimated absolute joint angle. The parameter γ is set relative to the number of contour correspondences to achieve a constant weighting for each frame. In practice, we use $\gamma = 0.02 \cdot |C_c|$.

Self-intersections are prevented by learning the physical constraints of the human skeleton from training data Θ^k where the probability of a pose p_{pose} is estimated by a Parzen-Rosenblatt estimator with Gaussian kernels over a small set of skeleton configurations, see Section 4.2.1. In the local optimization setting, this can be integrated by minimizing the negative logarithm [BRKC06]:

$$E_{pose} = -\ln(p_{pose}(\Theta)). \tag{5.27}$$

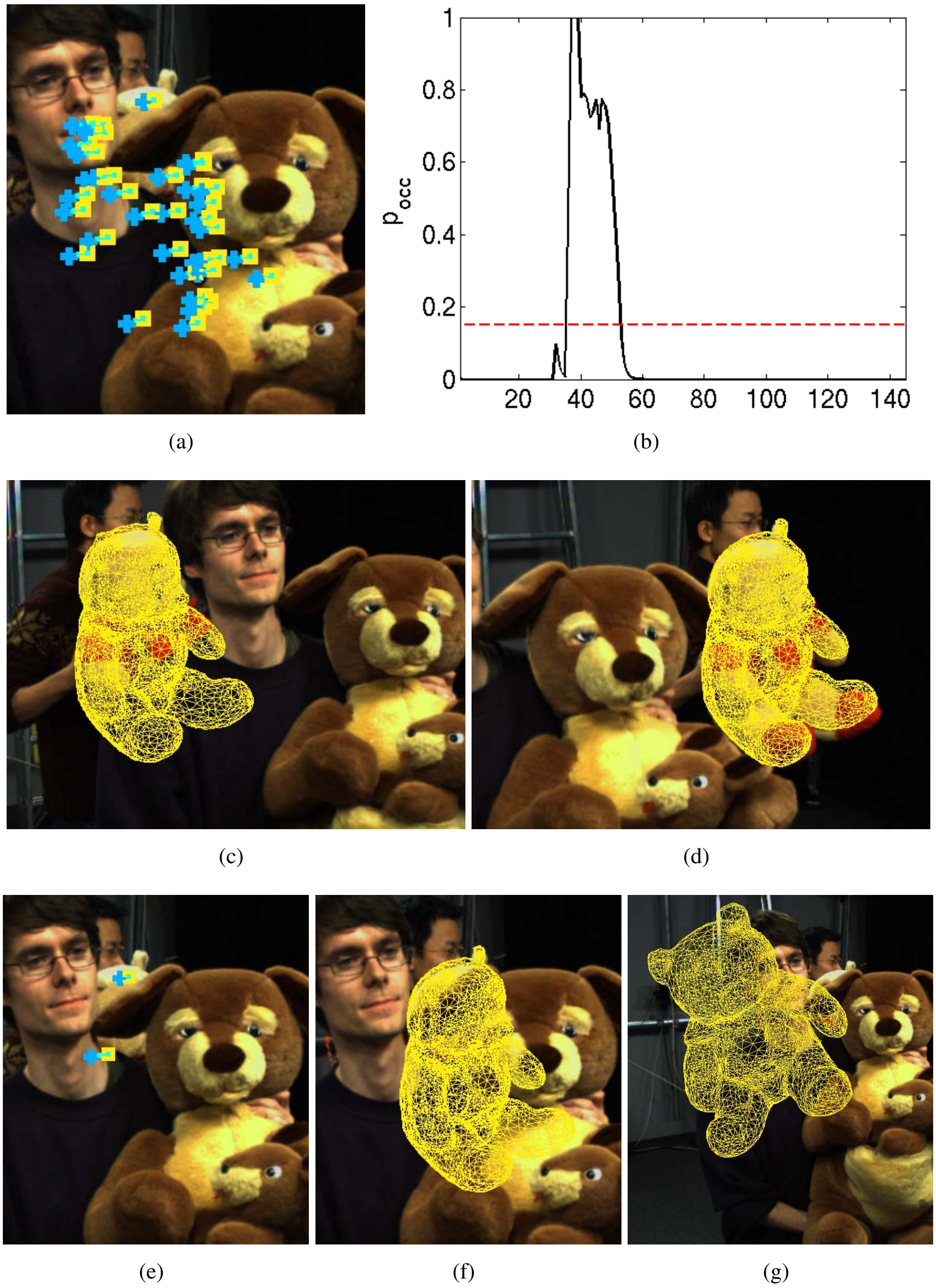
Using a gradient descent with step size τ, one obtains for each joint j an additional equation

$$\delta\theta_j = \delta \left(\tilde{\theta}_j + \tau \partial_t \tilde{\theta}_j \right) \quad \text{with} \quad \partial_t \tilde{\theta}_j = \frac{\sum_{k=1}^{N} \exp\left(-\frac{\|\Theta^k - \tilde{\Theta}\|^2}{2\sigma^2}\right) \left(\theta_j^k - \tilde{\theta}_j\right)}{\sigma^2 \sum_{k=1}^{N} \exp\left(-\frac{\|\Theta^k - \tilde{\Theta}\|^2}{2\sigma^2}\right)} \tag{5.28}$$

that is added to the system of linear equations (5.24). In our experiments, the parameters $\tau = 0.125\sigma^2$ and $\delta = 0.08 \cdot |C_c|$ yielded stable results. Since the dependency between the joints of the head, the upper, and the lower body is low, the sample size is reduced by splitting p_{pose} up into three independent probabilities p_{pose}^{head}, p_{pose}^{upper}, and p_{pose}^{lower}, respectively. Indeed, we use only 200 samples of the CMU motion database [CMU08].

Occlusion

Since patch-based matching between two successive frames is prone to occlusions, it requires the removal of correspondences not belonging to the target, see Figure 5.30 a). In our analysis-by-synthesis framework, occluded patches are detected by comparing the original image with the synthesized image. For this purpose, the patches are mapped into the CIELab color space that



(a) (b)

(c) (d)

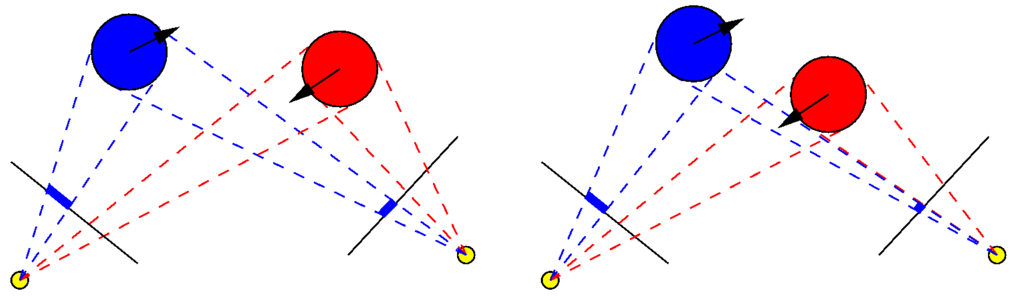
(e) (f) (g)

Figure 5.30: **From left to right. Row 1: a)** Most features belong to the occluding object (frame 44). The bear (target) is moving from left to right and the kangaroo from right to left. **b)** Probability of an occlusion in the shown view for a sequence with 150 frames. **Row 2: c)** The occlusion is correctly detected (frame 36). **d)** After the occlusion, the probability drops below the threshold 0.15 and the object is still correctly tracked (frame 52). **Row 3:** Frame 44. **e)** Almost all wrong matches are removed. **f)** Estimate after removing wrong matches. **g)** Estimate without occlusion handling.

mimics the human perception of color differences. To calculate the cross-correlation of color images [SE99], we represent each pixel with Lab color values as quaternion by $Li + aj + bk$. A correspondence is then labeled as occluded if the difference between the mean of the patch on the original image $P^o = \{p_1^o, \ldots, p_n^o\}$ and the patch on the synthesized image $P^s = \{p_1^s, \ldots, p_n^s\}$ is large or if the normalized cross correlation

$$NCC = \frac{\left|\sum_{i=1}^{n} \tilde{p}_i^o \overline{\tilde{p}_i^s}\right|}{\sqrt{\sum_{i=1}^{n} \tilde{p}_i^o \overline{\tilde{p}_i^o}} \sqrt{\sum_{i=1}^{n} \tilde{p}_i^s \overline{\tilde{p}_i^s}}} \tag{5.29}$$

is below a given threshold, where $\tilde{p}_i^o = p_i^o - \frac{1}{n}\sum_k p_k^o$ and $\overline{\tilde{p}_i^o}$ denotes the conjugate. An example for eliminating occluded patches is shown in Figures 5.30 a) and e).


(a)

(b)

Figure 5.31: Occlusions are detected by recognizing changes of the projected surfaces. **From left to right.** **a)** The scene contains two moving objects captured by two cameras. The target object *(blue)* is so far not occluded by the unknown object *(red)*. **b)** Since the target is now occluded in the right camera view, the visible area of the target is much smaller than in the previous frame. The ratio of the covered areas between the left and the right image plane has also changed.

To make the removal of patches more efficient, it is only performed when occlusions are detected for a camera view which is illustrated in Figure 5.31. When the target becomes occluded, the visible area of the projected surface gets smaller. By observing changes of the covered area for one view and the ratio between all views, occlusions can be detected. Since the visible areas of the projections cannot be measured, we use the number of matches as indicator, i.e. the difference of the absolute and relative number of matches between two successive frames for each view v:

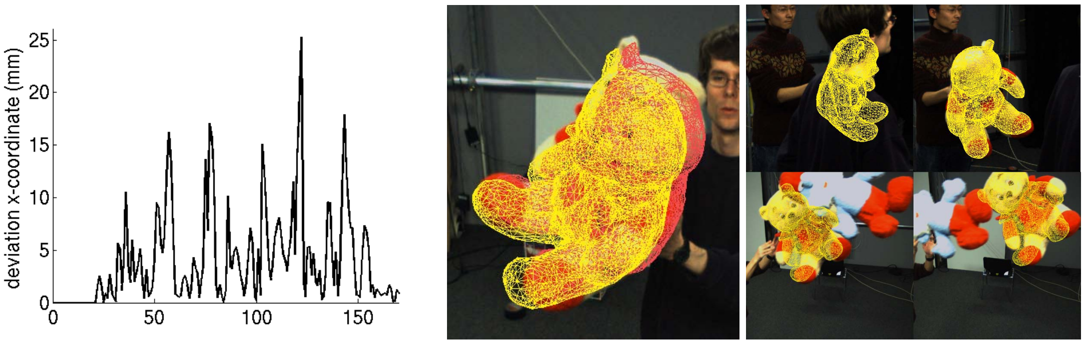
$$\Delta_{abs}^{v,t} = |C_o^{v,t}| - |C_o^{v,t-1}| \tag{5.30}$$

$$\Delta_{rel}^{v,t} = \frac{|C_o^{v,t}|}{\sum_u |C_o^{u,t}|} - \frac{|C_o^{v,t-1}|}{\sum_u |C_o^{u,t-1}|}. \tag{5.31}$$

While these numbers indicate the beginning of an occlusion, the occluded area is measured by the number of occluded patches $|C_{occ}^{v,t}|$ relative to all matches $|C_o^{v,t}|$. Based on these observations, we propose a recursive model for the probability of an occlusion at time t:

$$p_{occ}^{v,t} = \frac{|C_{occ}^{v,t}|}{|C_o^{v,t}|} - f(\Delta_{abs}^{v,t}) - f(\Delta_{rel}^{v,t}) + \frac{2}{5} p_{occ}^{v,t-1} - \frac{1}{2}, \tag{5.32}$$

where $p_{occ}^{v,t}$ is truncated to the interval $[0, 1]$ and $f(x) = x$ if $x < -0.2$ else zero. The function f ensures that only significant changes of at least 20% are taken into account. Using this model, the detection and removal of occluded patches is only performed when the probability is higher than 0.15, marked as the dashed line in Figure 5.30 b).



(a) (b) (c)

Figure 5.32: Comparison of the results with and without occlusion handling for a sequence containing no occlusions (170 frames). **From left to right: a)** Deviation of the x-coordinate. **b)** The frame (122) with the largest deviation between occlusion handling *(yellow)* and without detecting occlusions *(red)*. **c)** Two additional sequences with occlusions. *Top:* Object is occluded by a human. Frames 35 and 42 are shown. *Bottom:* Smurf and bear are thrown in opposite directions. Frames 35 and 38 are shown.

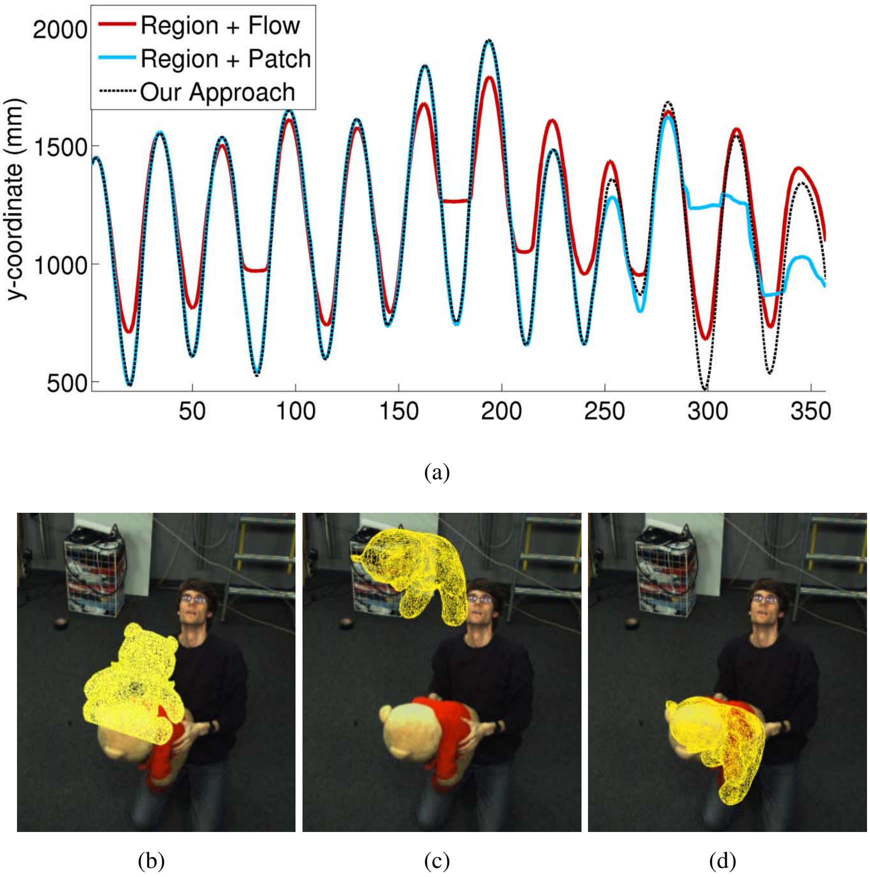
Initialization

Unsupervised initialization is important for applications since an initial pose is typically not given. Since neither a predicted pose nor a shape prior is available, we estimate the pose from the textured model assuming that the object is observable. To this end, we preliminarily render some initial views by rotating the textured object with a fixed joint configuration, e.g. the same as used for the texture acquisition, extract the features, and store them together with the mean values of the patches $\bar{P}^s$ and the corresponding pose parameters. For initialization, the extracted keypoints for the first frame are matched with the database and the best initial view is selected for estimating the pose. The obtained correspondences with mean values $\bar{P}_i^f$ and $\bar{P}_i^s$ are weighted by $|\bar{P}_i^f - \bar{P}_i^s|^{-2}$ for stabilizing the estimation.

5.5.3 Experiments

For evaluating the performance of our approach, we captured several scenes with different objects by 3–5 synchronized and calibrated cameras with 25 frames per second and resolution of 1004×1004 pixels. The 3D models were acquired by a 3D scan and the images for the texture acquisition were taken from a sequence where lighting conditions and camera positions differed from the test sequences.

Row 1 of Figure 5.34 shows some results for a sequence with a stuffed bear tracked using a rigid model. The bear with non-trivial shape is tossed by a human – the second moving object – and rotates in the air by more than 180 degrees. The scene contains background clutter and



(a)

(b)

(c)

(d)

Figure 5.33: Comparison. **Row 1:** Estimated y-coordinate of the bear for the sequence shown in row 1 of Figure 5.34. The approaches that rely only on multi-cue integration cannot handle the drift. **Row 2:** Frame 298. **From left to right:** **a)** Region-based matching with optical flow [BRCS06]. **b)** Region-based matching with PCA-SIFT. **c)** Our approach estimates the pose without drift.

is captured by 3 cameras. The occlusion detection is demonstrated in Figure 5.30. The stuffed kangaroo moves from right to left and occludes the stuffed bear moving from left to right such that the occluded target and the occluding object are moving at the same time. The occlusion between the frames 36 and 52 is correctly recognized and the pose is accurately estimated during the entire sequence whereas the tracking fails without an occlusion handling as shown in row 3 of Figure 5.30. Since the occlusion detection produces also false positives, e.g. for a rotation away from the camera, we measured the impact of the occlusion handling for another sequence without occlusions. In Figure 5.32 a), the absolute deviations of the estimates with occlusion handling from the estimates without detecting occlusions are plotted for the x-coordinate. The image next to the diagram shows that the induced error is small. Additional results for sequences with occlusions are shown in Figure 5.32 c).

To compare our method with multi-cue approaches that were proposed for rigid objects, we used the sequence corresponding to row 1 of Figure 5.34 and plotted the estimates for the y-coordinate in Figure 5.33. We applied the same level-set segmentation for all methods to make a fair com-

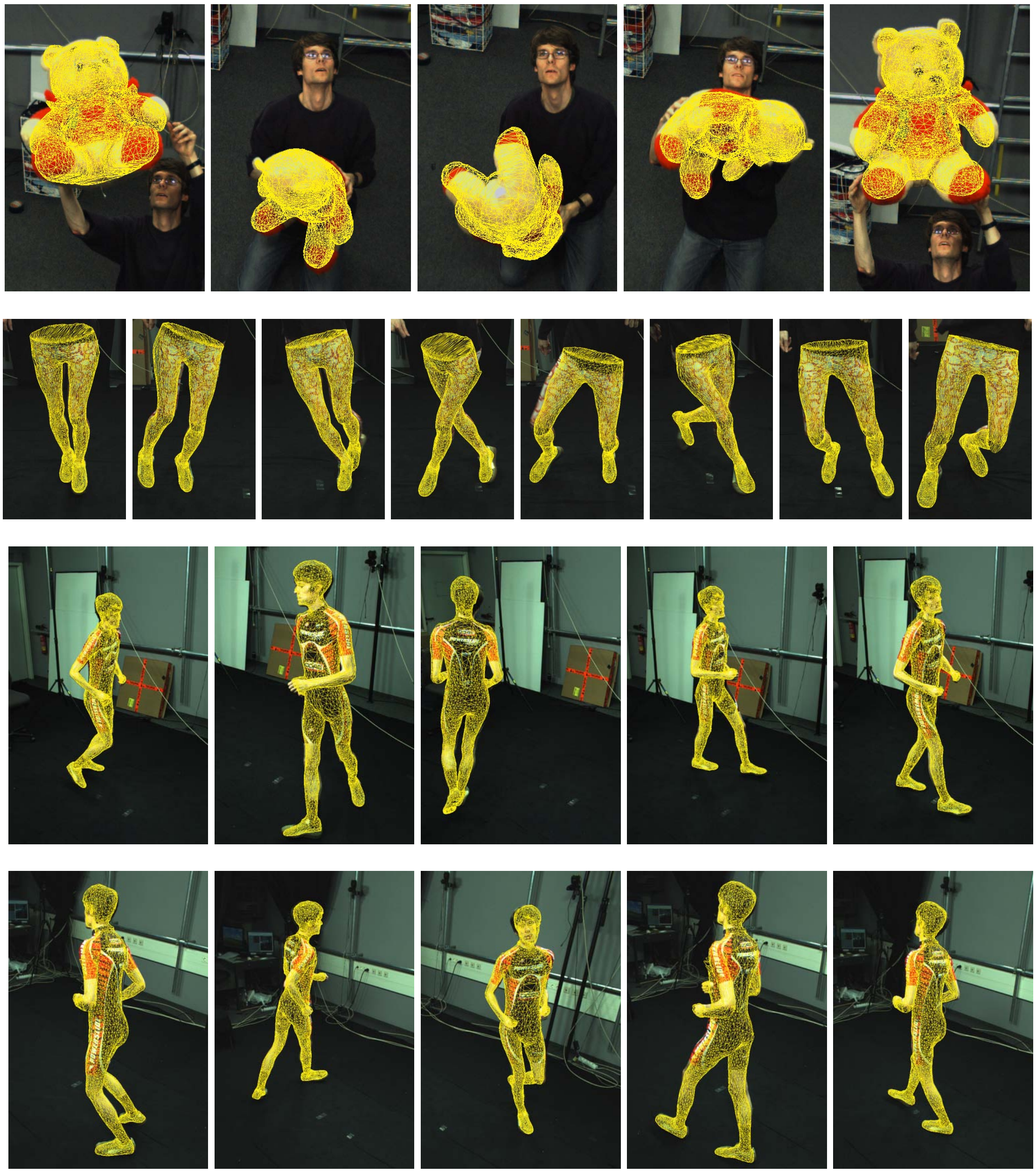


Figure 5.34: Estimates for three different sequences. **Row 1:** The bear is tossed and rotates (360 frames). One of three views for frames 60, 115, 180, 235, and 315. **Row 2:** Complex and fast movements of the legs including many self-occlusions (400 frames). One of four views for frames 45, 90, 135, 180, 225, 270, 315, and 360. **Row 3, 4:** Full human body walking in a circle with clutter (205 frames). Two of five views for frames 35, 70, 105, 140, and 175. 3D views of the objects are shown in Figure 5.35.

parison. While the approaches that combine region-based matching with optical flow [BRCS06] or patch-based matching cannot prevent an accumulation of estimation errors over time, our method tracks the stuffed bear accurately over the entire sequence. It demonstrates that our

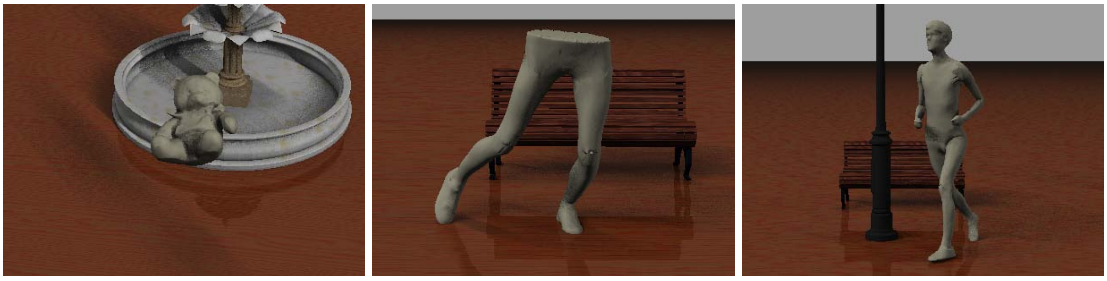


Figure 5.35: The objects from Figure 5.34 are inserted into virtual scenes.

framework solves the drift problem better than approaches that rely only on multi-cue integration.

The lower part of a human body was tracked using an articulated model with 18 DOF. As one can observe from the images with the projected meshes of the estimates in row 2 of Figure 5.34, the sequence recorded with 4 cameras is very challenging for a tracker. The movement is fast and the velocity and the direction change rapidly. In addition, self-occlusions occur since the legs are frequently crossed. We also tracked a full human body using an articulated model with 30 DOF. In rows 3 and 4 of Figure 5.34, estimates for 2 of 5 views are shown. The sequence with a human walking in a circle contains several difficulties. Self-occlusions occur since the arms are close to the body and the segmentation is hindered by clutter – particularly due to cables and metallic pipes –, shadows, and the similarity between the dark color of the sports suit and the background. Some body parts like the hands are furthermore small and homogeneous yielding only few correspondences from patch-based matching.

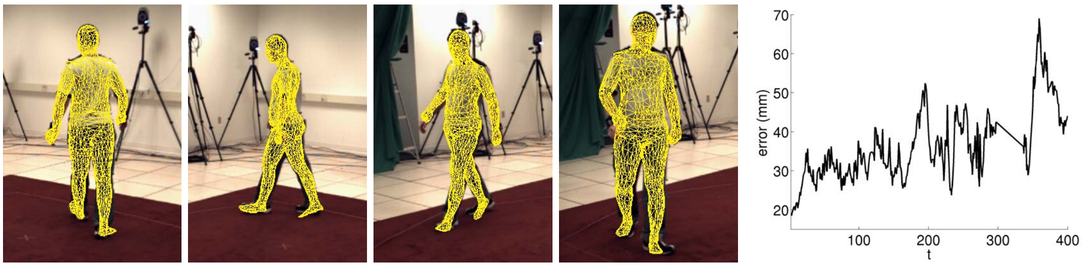


Figure 5.36: Quantitative error analysis for subject $S4$ of HumanEva-II. Estimates for frames 80, 160, 240, and 320. The frames $298-335$ are neglected for the error analysis since the ground truth is corrupted for these frames.

For a quantitative error analysis, we applied our approach to the HumanEva-II dataset [SB06] and measured the absolute 3D tracking error. The available model is not perfect since it does not contain the clothing of the subject $S4$ wearing a white T-shirt and blue jeans. The texture was acquired from the first frame and the available silhouettes were treated as an additional channel for the segmentation. Even though the surface of the object is rather homogeneous, we achieve accurate estimates as shown in Figure 5.36. Since the set-up and movement of the sequence, namely walking in a circle, is similar to the one used in [BB06], we compare the results in Table 5.3. Our implementation requires 7.6 seconds per image which is faster than the

90 seconds reported in [BB06].

	Our approach	$RoAM$ body model [BB06]
error (mm)	36.16 ± 9.12	> 60

Table 5.3: Our framework performs significantly better than a statistical appearance model for human motion capture.

5.5.4 Summary

In this section, we have presented a model-based tracking framework for solving the drift-problem for rigid and articulated objects. An occlusion detection, which evaluates the probability of an occlusion, observes significant changes of the visible area of the projected surface during the sequence and initiates recognition of occluded patches by comparing the original image with a synthesized image, if it is necessary. Since the synthesized image also provides accurate correspondences, an accumulation of estimation errors is prevented. By combining the complementary concepts of region and patch-based matching, both structured and homogeneous body parts can be tracked. A comparison with other model-based approaches for rigid objects has revealed that the proposed method handles the drift problem better. Our experiments have demonstrated that our framework is not restricted to a single rigid object but tackles the drift problem also for multiple moving objects and humans in challenging scenes containing fast movements, occlusions, and clutter. Although our framework benefits from objects with structured surfaces and accurate 3D models, a quantitative error analysis for the `HumanEva-II` dataset has shown that we still achieve accurate results when these assumptions are not completely satisfied. Indeed, the tracking error is significantly lower than the one that is obtained by a statistical appearance model for human motion capture. Furthermore, the proposed framework can be applied to challenging real-world problems as we demonstrate in the following section where crash test video analysis is used as an example.

5.6 Crash Test Video Analysis

The analysis of crash test videos is an important task for the automotive industry in order to improve the passive safety components of cars. In particular, the motion estimation of crash test dummies helps to improve the protection of occupants and pedestrians. The standard techniques for crash analysis use photogrammetric markers that provide only sparse 3D measurements, which do not allow the estimation of the head orientation.

Here we address motion capture of rigid body parts in crash test videos where we concentrate on the head – one of the most sensitive body parts in traffic accidents. As shown in Figure 5.37 a), this is very challenging since the head covers only a small area of the image and large parts are occluded by the airbag. In addition, shadows and background clutter make it difficult to distinguish the target object from the background. To this end, we propose our model-based approach that estimates the absolute 3D rotation and position of the object from multiple views independently of photogrammetric markers. In order to make the estimation robust to occlusions, reference images are synthesized using a 3D model that contains the geometry and the texture of the object. Since our approach further combines region-based and patch-based matching, reliable estimates are obtained even for small body parts as demonstrated in Figure 5.37.

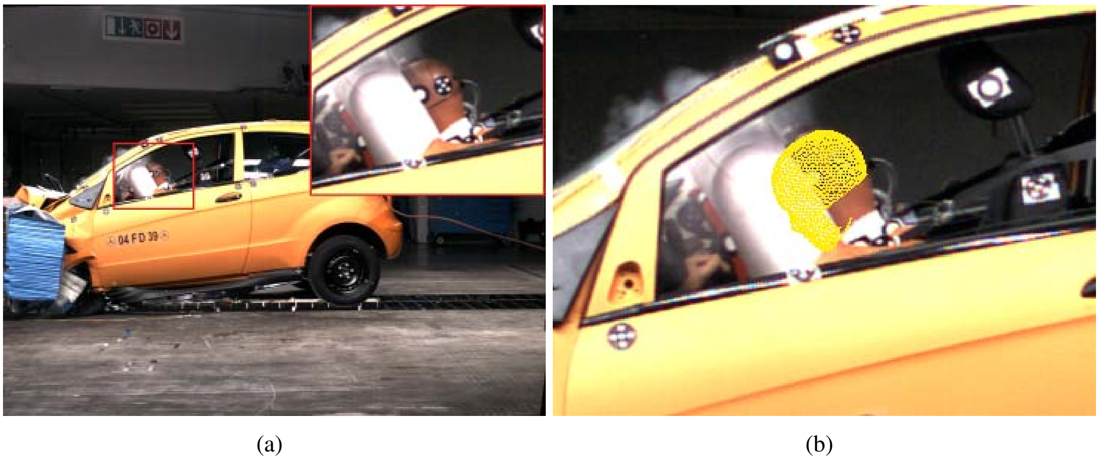

(a) (b)

Figure 5.37: **From left to right: a)** Estimating the pose of the dummy's head from crash test videos is very challenging. The target object is relatively small and partially occluded by the airbag. Furthermore, background clutter and the car's shadow make it difficult to distinguish the head from the background. **b)** Estimated pose of the dummy's head. The 3D surface model is projected onto the image.

Tracking of small objects for crash video analysis without a surface model has also been investigated in [GBP06], where the relative transformation is reconstructed from a 3D point cloud that is tracked using KLT [ST94] and stereo depth data. In contrast to model-based approaches, point clouds do not provide all relevant information like depth of penetration or absolute head orientation.

5.6.1 Implementation

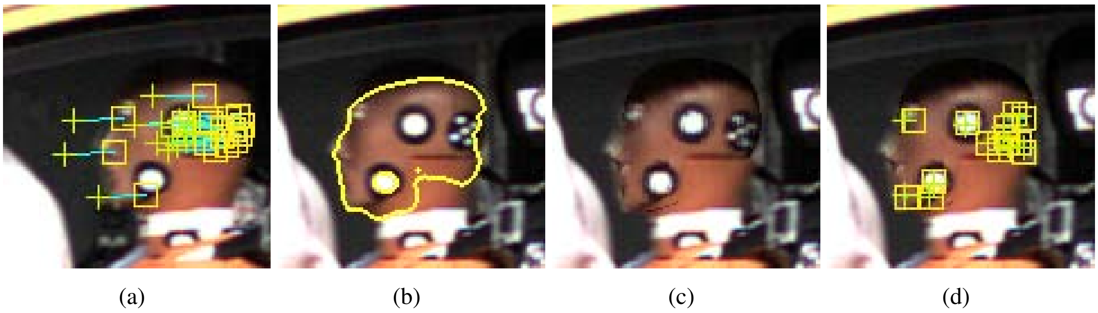
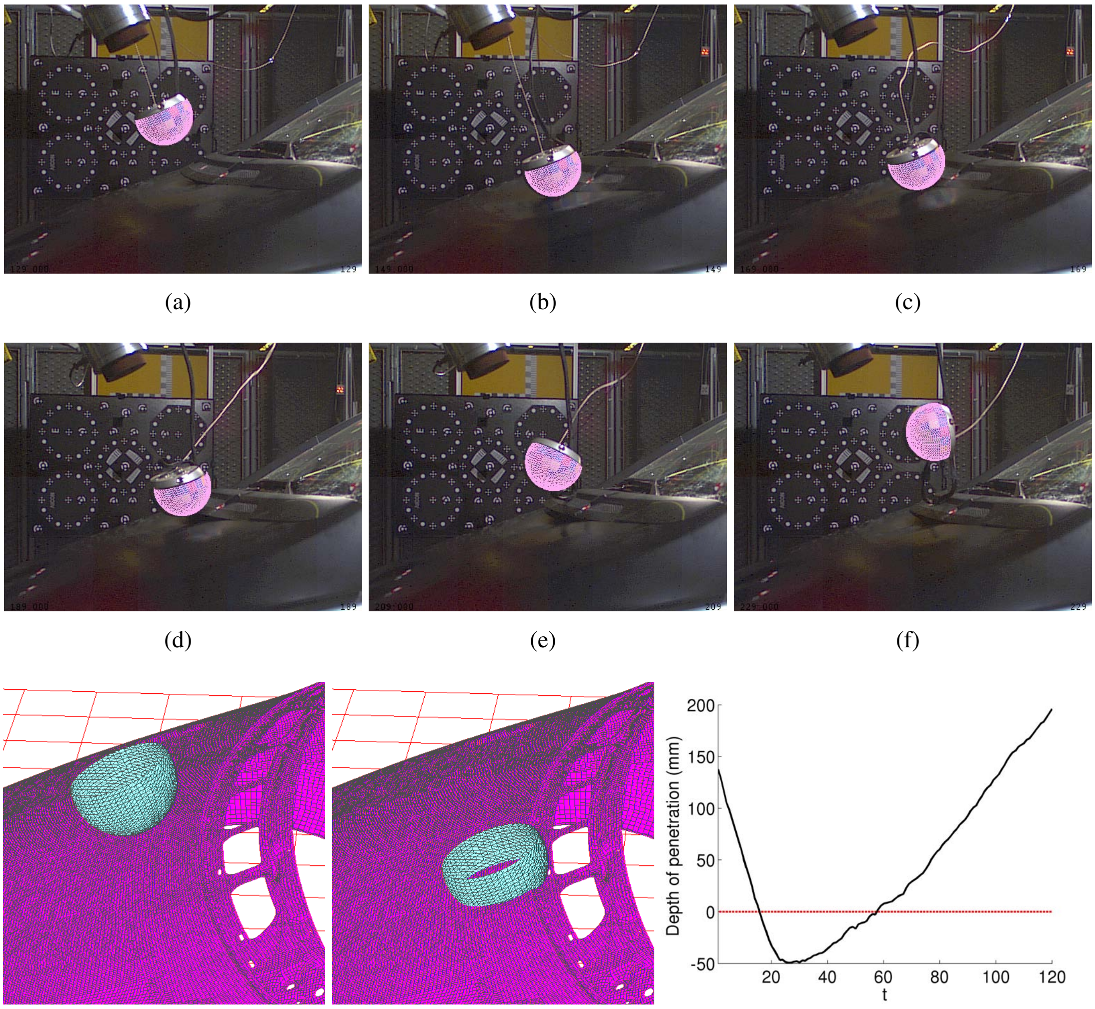

(a) (b) (c) (d)

Figure 5.38: . **From left to right: a)** Correspondences between current frame *(square)* and next frame *(cross)*. **b)** Estimated contour. **c)** Synthesized image. **d)** Correspondences between synthesized image (*square*) and original image (*cross*).

The 3D surface model of a crash test dummy is readily available as most dummies are manufactured according to ISO standards. The texture is often not provided but it can be acquired by projecting the images from the calibrated cameras on the object's surface using the technique described in [GRS07] and Section 6.2 to align the 3D model to the images. The tracking system is the same as described in Section 5.5. The used cues are illustrated in Figure 5.38.

5.6.2 Experiments


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.39: **Rows 1, 2:** The head crashes onto the engine hood. Estimates for frames 5, 25, 45, 65, 85, and 105 are shown (*from top left to bottom right*). The pose of the head is well estimated for the entire sequence. **Row 3:** Virtual reconstruction of the crash showing the 3D surface model of the head and of the engine hood. **From left to right: g)** Frame 5. **h)** Frame 25. The head penetrates the engine hood. **i)** Depth of penetration. The black curve shows the distance of the head to the engine hood (*dashed line*).

The first experiment investigates the dynamics of a pedestrian head crashing onto the engine hood, see Figure 5.39. The sequence has been captured at 1000 Hz by two calibrated cameras with 512×384 pixel resolution. For segmentation, the images have been converted to the CIELab color space that mimics the human perception of color differences. Since we have registered the engine hood as shown in row 3 of Figure 5.39, the depth of penetration can be measured from

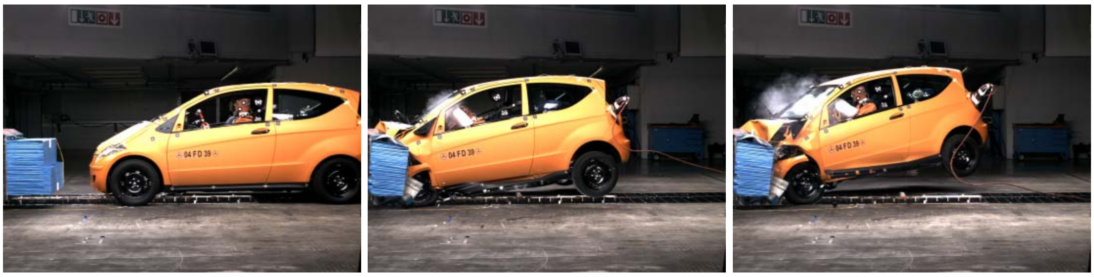
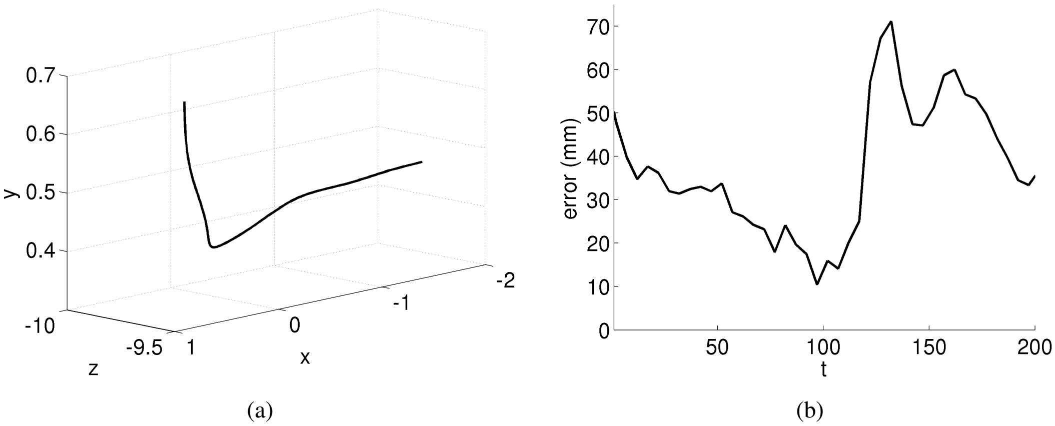


Figure 5.40: Three frames of the EuroNCAP offset crash sequence. The car jumps and moves laterally due to the offset barrier. The head is occluded by more than 50% at the moment of the deepest airbag penetration.



(a)

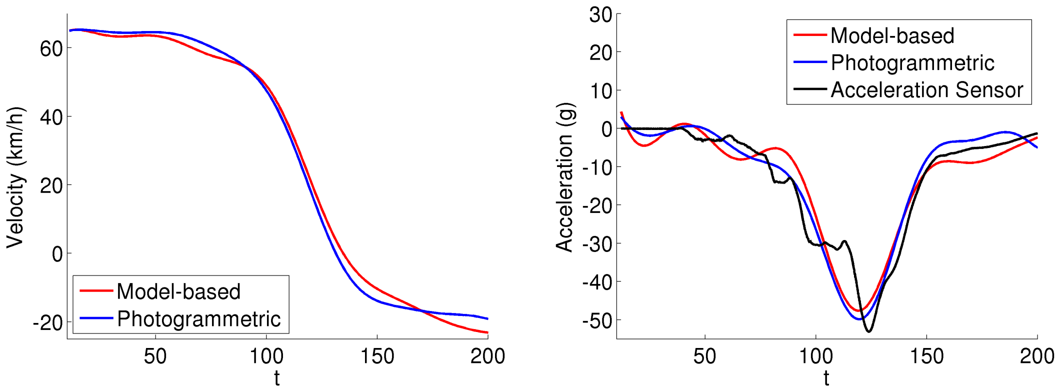
(b)

Figure 5.41: **From left to right: a)** 3D trajectory of the head. **b)** 3D tracking error of the head. The ground truth is obtained from a marker-based system. Note that the object is about $10m$ away from the camera.

the estimated head pose. In this case, the head penetrates $49.1mm$ into the engine compartment. This information is relevant for crash test analysis since severe head injuries might be caused by crashing into the solid engine block. Note that a standard silhouette-based approach would not be able to estimate the rotation due to the symmetric shape of the object whereas our approach provides good results for the entire sequence.

For the second experiment, the head of a dummy is tracked during a EuroNCAP offset crash where the car drives into an aluminum barrier with 40% overlap as shown in Figure 5.40. Due to the barrier, the car jumps and moves laterally. Although the sequence was captured at 1000 Hz by 3 cameras with 1504×1128 pixel resolution, the head covers only 70×70 pixels, i.e., less than 0.3% of the image pixels. In addition, the head is occluded by more than 50% at the moment of the deepest airbag penetration and the segmentation is hindered by shadows and background clutter. Nevertheless, Figure 5.43 demonstrates that the head pose is well estimated by our model-based approach during the crash. The trajectory in Figure 5.41 reflects the upward and lateral movement of the car away from the camera due to the offset barrier.

For a quantitative error analysis, we have compared the results with a marker-based system using



(a)

(b)

Figure 5.42: Comparison with a marker-based system and an acceleration sensor. The model-based approach provides accurate estimates for velocity and acceleration. **From left to right: a)** Velocity (x-axis). **b)** Acceleration (x-axis).

photogrammetric markers. The 3D tracking error is obtained by the Euclidean distance between the estimated position and the true position of the 5-dot marker on the left hand side of the dummy's head. The results are plotted in Figure 5.41 where the average error is $37mm$ with standard deviation of $15mm$. For computing the velocity and the acceleration of the head, the trajectories from the marker-based and the model-based method are slightly smoothed, as it is common for crash test analysis. Figure 5.42 shows that the velocity and the acceleration are well approximated by our approach. A comparison with an acceleration sensor attached to the head further reveals that the deceleration is similar to the estimates of our approach. For the offset crash sequence, our current implementation requires 6 seconds per frame on a consumer PC. Finally, we remark that our approach estimates all six degrees of freedom of dummy body parts like the head in contrast to conventional marker-based systems. This opens up new opportunities for analyzing pedestrian crashes where many biomechanical effects are not fully understood.

5.7 Summary

Local optimization can solve human motion capture accurately but it cannot recover from errors. Hence, cues are required that guide the local optimization to the true pose. Furthermore, they need to be robust to occlusions, illumination changes, and clutter and they need to be reliable for homogeneous and structured surfaces. Since none of the typical cues for motion capture like silhouettes, edges, color, motion, and texture meets the demands, a multi-cue integration is necessary for tracking complex objects like humans. For instance, the region-based approach, Section 5.2, works well for homogeneous objects but it requires many iterations until convergence, which makes the approach very expensive. Particularly for large transformations from frame to frame, the segmentation and consequently the pose estimation usually get stuck in a local optimum. Another problem is ambiguous solutions for symmetric objects. Hence, we have extended the approach with motion cues in Section 5.3. Motion-cues are complementary to silhouettes since they perform better on sufficiently structured objects. Furthermore, they can handle large transformations between successive frames, so that the number of required itera-

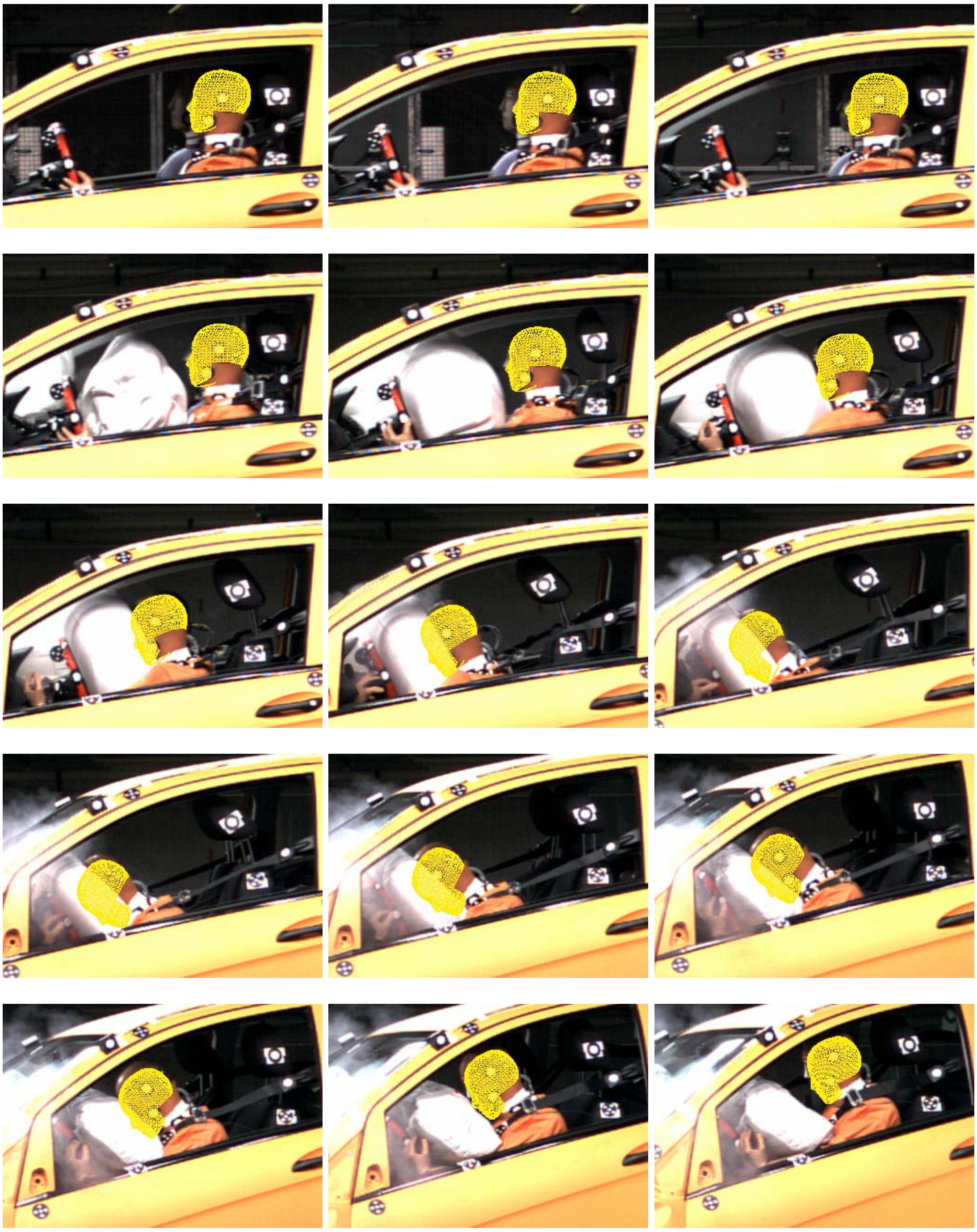


Figure 5.43: Estimated pose of the dummy's head for frames 7, 22, 37, 52, 67, 82, 97, 112, 127, 142, 157, 172, 187, 202, and 217 (*from top left to bottom right*).

tions for optimization is reduced. Since the impact of a cue should be large in situations when its extraction is reliable, and small, if the information is likely to be erroneous, we have proposed an adaptive weighting scheme that combines the complementary cues. We have also compared dense and sparse features, namely dense optical flow and local descriptors. As local descriptors, we have used the popular SIFT features but they could also be replaced by other features like SURF [BETG08] or DAISY [TLF08]. The highest accuracy and robustness has been achieved by using sparse and dense features at the same time. Although both are motion cues, they have different strength. While the estimated optical flow might not be exact in difficult situations, it provides at least enough correspondences for a unique approximate solution. SIFT correspondences are usually more reliable, but their number is sometimes not sufficient to estimate the pose. Instead of computing the flow independently from the local descriptors, one could also use the SIFT correspondences as prior for the energy function (5.17). This, however, would increase the influence of outliers from patch-based matching on the estimated pose since the outliers misguide the flow computation.

Even though the combination of surface-region matching, optical flow, and SIFT tracking provides precise estimates for rigid and articulated objects with homogeneous and structured surfaces, it does not solve the drift problem. Since motion cues rely on image features that are tracked over time, the accumulation of small errors results in a drift away from the target object that cannot be compensated by the region-based features. Hence, we have addressed the drift problem for human motion capture and tracking in the presence of multiple moving objects where the error accumulation becomes even more problematic due to occlusions. To this end, we have proposed in Section 5.5 an analysis-by-synthesis framework for articulated models relying on a combination of region-based and motion-based cues. A comparison with other model-based approaches for rigid objects has revealed that the proposed method handles the drift problem better. Our experiments have demonstrated that our framework is not restricted to a single rigid object but tackles the drift problem also for multiple moving objects and humans in challenging scenes containing fast movements, occlusions, and clutter. Furthermore, the proposed framework can be applied to challenging real-world problems as we have demonstrated in Section 5.6 where crash test video analysis is used as an example. In general, one could also use several motion cues for the analysis-by-synthesis framework to get a better performance, but computation of local descriptors and optical flow for each frame is still very expensive in spite of efficient implementations of optical flow [BW05] or SIFT [SFPG07]. Hence, we have used only local descriptors since the constancy assumptions of the optical flow are clearly violated between the original and the synthesized image.

There are still several limitations of the analysis-by-synthesis framework. Drift is only prevented as long as there are enough correspondences between the synthesized and the original image. This means that the approach assumes that there is enough texture information to establish these correspondences. However, even in the worst case where no correspondences are available, the method still behaves as the approach presented in Section 5.3. In general, the method benefits from high resolution images whereas the framerate is less important since large transformations are captured by patch-based matching. Since high-definition cameras are already widely used in contrast to high-speed cameras, assuming a high image resolution is not very restrictive. Another limitation is given by the clothing. Although the approach does not require tight-fitting apparel, it cannot handle arbitrary deformable surfaces or wide apparel like skirts. Since the skeleton-based surface deformation is only an approximation of the real surface deformation, the accuracy of the correspondences between the textured surface and the original image depends on the quality of the approximation. Furthermore, changes of the illumination are implicitly handled by the robustness of the used features. For handling illumination changes that exceed the abilities of the features, the textured model needs to be extended such that the lighting environment is also taken into account [LSLF08, YWP06].

In order to estimate the human pose precisely and accurately, local optimization requires strong cues. Since local optimization itself is very fast, the extraction of the cues dominates the computing time. However, when the sources for feature extraction are limited, only weak cues are available. This might also occur due to low contrast or low size of the object despite high-resolution images. Without strong cues, however, local optima become more critical and need to be resolved by global optimization methods. Another global optimization problem is the initialization of model-based tracking approaches where the pose cannot be predicted from the previous frame. The initialization problem occurs also for texture acquisition which is needed for the analysis-by-synthesis framework. Although static 3D scan devices equipped with cameras or photogrammetric reconstruction techniques can acquire shape and texture, it is more convenient to acquire the texture directly from the video stream since the surface color is likely to change from sequence to sequence in contrast to the human body shape. Therefore, there is a need for global optimization techniques that meet the demands of human pose estimation and tracking.

6

Global Optimization

While local optimization and filtering approaches have been widely employed for human motion capture as discussed in Section 2.3, global optimization techniques are hardly used for pose estimation due to the high computational cost of standard approaches. For instance, the method proposed in [CMC$^+$06] relies on fast simulated annealing and requires hours to estimate a single pose. On the contrary, local optimization requires an initialization near the global optimum for each frame. As we have seen in Chapter 5, this can be achieved by an analysis-by-synthesis framework under the assumption that enough texture information is available. In the context of initialization, texture acquisition, or low contrast videos, however, this assumption is usually not satisfied. Hence, there is a need for a global optimization approach that solves the human motion capture problem more efficiently than standard techniques.

To this end, we introduce in Section 6.1 a global optimization method based on an interacting particle system [Mor04] that overcomes the dilemma of local optima and that is suitable for the optimization problems as they arise in human motion capturing. In contrast to many other optimization algorithms, a distribution instead of a single value is approximated by a particle representation similar to particle filters [DFG01]. This property is beneficial, particularly for tracking where the right parameters are not always exact at the global optimum depending on the image features that are used. Besides a discussion of the asymptotic behavior, we also provide an exhaustive parameter evaluation and two examples, namely a standard global optimization problem and a naïve human motion capture approach. More advanced extensions for full-body human pose estimation and tracking are discussed in Sections 6.2 and 6.3.

The method is capable of estimating the human pose without initial information, which is a challenging optimization problem in a high dimensional space and is essential for initialization and texture acquisition. Furthermore, we propose a tracking framework that is based on this optimization technique to achieve both the *robustness* of filtering strategies and a remarkable *accuracy*. The latter is demonstrated by a quantitative error analysis that includes the `HumanEva-II` benchmark [SB06] and a comparison with several optimization and particle filtering approaches.

6.1 Interacting Simulated Annealing

6.1.1 Interacting Particle Systems

A popular global optimization method inspired by statistical mechanics is known as simulated annealing [GG84, KJV83]. Similar to our approach, a function $V \geq 0$ interpreted as energy is minimized by means of an unnormalized Boltzmann-Gibbs measure that is defined in terms of

V and an inverse temperature $\beta > 0$ by

$$g(dx) = \exp\left(-\beta\, V(x)\right)\, \lambda(dx), \tag{6.1}$$

where λ is the Lebesgue measure. This measure has the property that the probability mass concentrates at the global minimum of V as $\beta \to \infty$.

The key idea behind simulated annealing is taking a random walk through the search space while β is successively increased. The probability of accepting a new value in the space is given by the Boltzmann-Gibbs distribution. While values with less energy than the current value are accepted with probability one, the probability that values with higher energy are accepted decreases as β increases. Other related approaches are fast simulated annealing [SH87] using a Cauchy-Lorentz distribution and generalized simulated annealing [TS96] based on Tsallis statistics.

Interacting particle systems [Mor04] approximate a distribution of interest by a finite number of weighted random variables $X^{(i)}$ called particles. Provided that the weights $\Pi^{(i)}$ are normalized such that $\sum \Pi^{(i)} = 1$, the set of weighted particles determines a random probability measure by

$$\sum_{i=1}^{n} \Pi^{(i)} \delta_{X^{(i)}}. \tag{6.2}$$

Depending on the weighting function and the distribution of the particles, the measure converges to a distribution η as n tends to infinity. When the particles are identically independently distributed according to η and uniformly weighted, i.e. $\Pi^{(i)} = 1/n$, the convergence follows directly from the law of large numbers [Bau96].

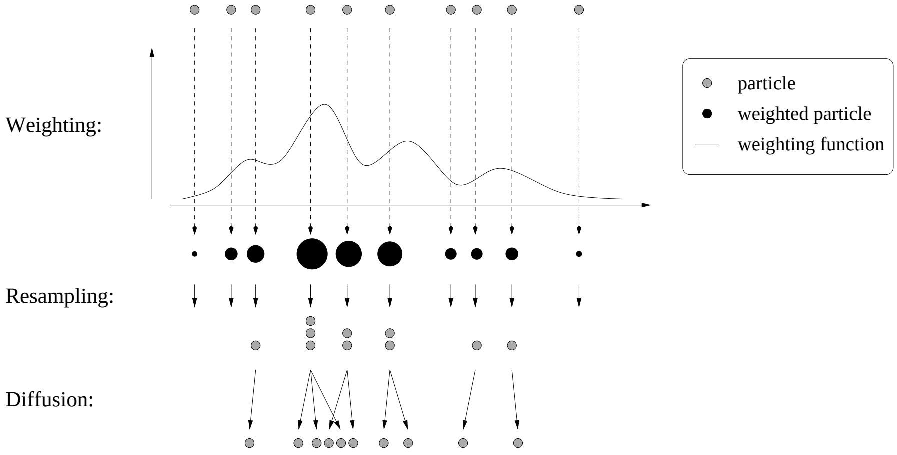


Figure 6.1: Operation of an interacting particle system. After weighting the particles (*black circles*), the particles are resampled and diffused (*gray circles*).

Interacting particle systems are mostly known in computer vision as particle filter [DFG01] where they are applied for solving non-linear, non-Gaussian filtering problems as described in Section 3.4. However, these systems also apply for trapping analysis, evolutionary algorithms, statistics [Mor04], and optimization as we demonstrate in this chapter. They usually consist of two steps as illustrated in Figure 6.1. During a selection step, the particles are weighted according to a weighting function and then resampled with respect to their weights, where particles with a great weight generate more offspring than particles with lower weight. In a second step, the particles mutate or are diffused. Since a particle filter is only a special case of an interacting particle system, we use the more general terms “Selection” and “Mutation” for the two steps instead of “Updating” and “Prediction”, which have been used in Figure 3.2.

6.1.2 Interaction and Annealing

Simulated annealing approaches are designed for global optimization, i.e. for searching the global optimum in the entire search space. Since they are not capable of focusing the search on some regions of interest in dependency on the previous visited values, they are not suitable for tasks in human motion capturing. Our approach, in contrast, is based on an interacting particle system that uses Boltzmann-Gibbs measures (6.1) similar to simulated annealing. This combination ensures not only the annealing property as we will show, but also exploits the distribution of the particles in the search space as measure for the uncertainty in an estimate. The latter allows an automatic adaption of the search on regions of interest during the optimization process. The principle of the annealing effect is illustrated in Figure 6.2.

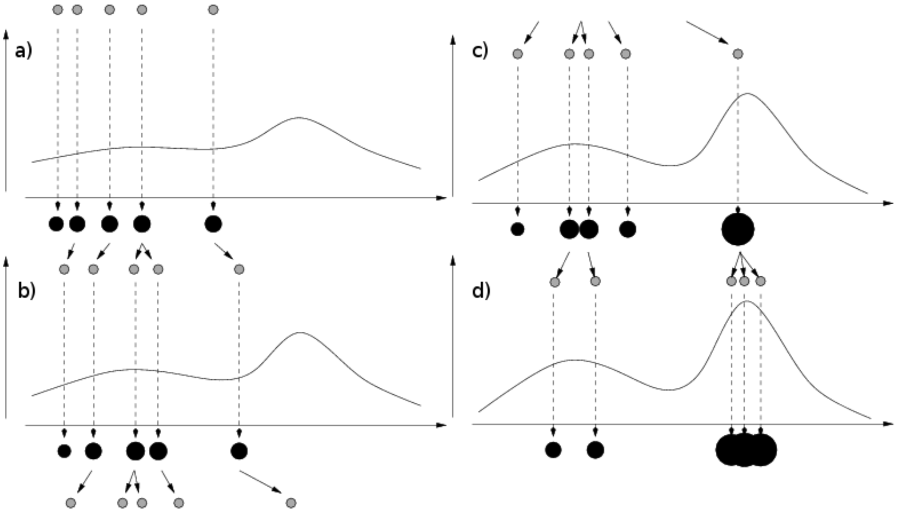


Figure 6.2: Illustration of the annealing effect with three runs. Due to annealing, the particles migrate towards the global maximum without getting stuck in the local maximum.

A first attempt to fuse interaction and annealing strategies for human motion capturing has become known as annealed particle filter [DR05]. Even though the heuristic is not based on a mathematical background, it already indicates the potential of such combination. Indeed, the annealed particle filter can be regarded as a special case of interacting simulated annealing where the particles are predicted for each frame by a stochastic process, see Section 6.1.5.

6.1.3 Notations

The notations introduced in Section 3.4.1 are repeated for the reader’s convenience. We always regard E as a subspace of R^d, and let $\mathcal{B}(E)$ denote its Borel σ-algebra. $B(E)$ denotes the set of bounded measurable functions, δ_x is the Dirac measure concentrated in $x \in E$, $\| \cdot \|_2$ is the Euclidean norm, and $\| \cdot \|_\infty$ denotes the supremum norm. Let $f \in B(E)$, μ be a measure on E, and let K be a Markov kernel on E. We write

$$\langle \mu, f\rangle = \int_E f(x)\,\mu(dx), \quad \langle \mu, K\rangle(B) = \int_E K(x,B)\,\mu(dx) \quad \text{for } B \in \mathcal{B}(E).$$

Furthermore, $U[0,1]$ denotes the uniform distribution on the interval $[0,1]$ and

$$\operatorname{osc}(\varphi) := \sup_{x,y\in E} \{|\varphi(x) - \varphi(y)|\}\,. \tag{6.3}$$

is an upper bound for the oscillations of f.

6.1.4 Feynman-Kac Model

Let $(X_t)_{t\in\mathbb{N}_0}$ be an E-valued Markov process with family of transition kernels $(K_t)_{t\in\mathbb{N}_0}$ and initial distribution η_0. We denote by P_{η_0} the distribution of the Markov process, i.e. for $t \in \mathbb{N}_0$,

$$P_{\eta_0}\left(d(x_0, x_1, \ldots, x_t)\right) = K_{t-1}(x_{t-1}, dx_t)\ldots K_0(x_0, dx_1)\,\eta_0(dx_0),$$

and by $E_{\eta_0}[\cdot]$ the expectation with respect to P_{η_0}. The sequence of distributions $(\eta_t)_{t\in\mathbb{N}_0}$ on E defined for any $\varphi \in B(E)$ and $t \in \mathbb{N}_0$ as

$$\langle \eta_t, \varphi\rangle := \frac{\langle \gamma_t, \varphi\rangle}{\langle \gamma_t, 1\rangle}, \qquad \langle \gamma_t, \varphi\rangle := E_{\eta_0}\left[\varphi\left(X_t\right)\exp\left(-\sum_{s=0}^{t-1}\beta_s\,V(X_s)\right)\right],$$

is called the *Feynman-Kac model* associated with the pair $(\exp(-\beta_t\,V), K_t)$.
The Feynman-Kac model as defined above satisfies the recursion relation

$$\eta_{t+1} = \langle \Psi_t(\eta_t), K_t\rangle, \tag{6.4}$$

where the *Boltzmann-Gibbs transformation* Ψ_t is defined by

$$\Psi_t\left(\eta_t\right)\left(dy_t\right) = \frac{E_{\eta_0}\left[\exp\left(-\sum_{s=0}^{t-1}\beta_s\,V(X_s)\right)\right]}{E_{\eta_0}\left[\exp\left(-\sum_{s=0}^{t}\beta_s\,V(X_s)\right)\right]}\exp\left(-\beta_t\,V_t(y_t)\right)\,\eta_t(dy_t).$$

The particle approximation of the flow (6.4) depends on a chosen family of Markov transition kernels $(K_{t,\eta_t})_{t\in\mathbb{N}_0}$ satisfying the compatibility condition

$$\langle \Psi_t\left(\eta_t\right), K_t\rangle := \langle \eta_t, K_{t,\eta_t}\rangle.$$

A family $(K_{t,\eta_t})_{t\in\mathbb{N}_0}$ of kernels is not uniquely determined by these conditions.
As in [Mor04, Chapter 2.5.3], we choose

$$K_{t,\eta_t} = S_{t,\eta_t}K_t, \tag{6.5}$$

where

$$\begin{aligned} S_{t,\eta_t}(x_t, dy_t) \;=\;& \epsilon_t\,\exp\left(-\beta_t\,V_t(x_t)\right)\,\delta_{x_t}(dy_t) \\ &+ \left(1 - \epsilon_t\,\exp\left(-\beta_t\,V_t(x_t)\right)\right)\,\Psi_t\left(\eta_t\right)\left(dy_t\right), \end{aligned} \tag{6.6}$$

with $\epsilon_t \geq 0$ and $\epsilon_t \left\|\exp(-\beta_t\,V)\right\|_\infty \leq 1$. The parameters ϵ_t may depend on the current distribution η_t.

6.1.5 Interacting Simulated Annealing

Similar to simulated annealing, one can define an annealing scheme $0 \leq \beta_0 \leq \beta_1 \leq \ldots \leq \beta_t$ in order to search for the global minimum of an energy function V. Under some conditions that will be stated later, the flow of the Feynman-Kac distribution becomes concentrated in the region of global minima of V as t goes to infinity. Since it is not possible to sample from the distribution directly, the flow is approximated by a particle set as it is done by a particle filter. We call the algorithm for the flow approximation *interacting simulated annealing* (ISA).

Algorithm

The particle approximation for the Feynman-Kac model is completely described by the Equation (6.5). The particle system is initialized by n identically, independently distributed random variables $X_0^{(i)}$ with common law η_0 determining the random probability measure $\eta_0^n := \sum_{i=1}^n \delta_{X_0^{(i)}}/n$. Since K_{t,η_t} can be regarded as the composition of a pair of selection and mutation Markov kernels, we split the transitions into the following two steps

$$\eta_t^n \xrightarrow{Selection} \check{\eta}_t^n \xrightarrow{Mutation} \eta_{t+1}^n,$$

where

$$\eta_t^n := \frac{1}{n}\sum_{i=1}^n \delta_{X_t^{(i)}}, \qquad \check{\eta}_t^n := \frac{1}{n}\sum_{i=1}^n \delta_{\check{X}_t^{(i)}}.$$

During the selection step each particle $X_t^{(i)}$ evolves according to the Markov transition kernel $S_{t,\eta_t^n}(X_t^{(i)},\cdot)$. That means $X_t^{(i)}$ is accepted with probability $\epsilon_t \exp(-\beta_t\, V(X_t^{(i)}))$, and we set $\check{X}_t^{(i)} = X_t^{(i)}$. Otherwise, $\check{X}_t^{(i)}$ is randomly selected with distribution

$$\sum_{i=1}^n \frac{\exp(-\beta_t\, V(X_t^{(i)}))}{\sum_{j=1}^n \exp(-\beta_t\, V(X_t^{(j)}))}\,\delta_{X_t^{(i)}}.$$

The mutation step consists in letting each selected particle $\check{X}_t^{(i)}$ evolve according to the Markov transition kernel $K_t(\check{X}_t^{(i)},\cdot)$.

There are several ways to choose the parameter ϵ_t of the selection kernel (6.6) that defines the resampling procedure of the algorithm, cf. [Mor04]. If

$$\epsilon_t := 0 \qquad \forall t, \tag{6.7}$$

the selection can be done by multinomial resampling. Provided that[1]

$$n \geq \sup_t \left(\exp(\beta_t \ \mathrm{osc}(V)\right),$$

another selection kernel is given by

$$\epsilon_t(\eta_t) := \frac{1}{n\,\langle \eta_t, \exp(-\beta_t\, V)\rangle}. \tag{6.8}$$

[1] The inequality satisfies the condition $\epsilon_t \left\|\exp(-\beta_t\, V)\right\|_\infty \leq 1$ for Equation (6.6).

Algorithm 2 Interacting Simulated Annealing Algorithm

Requires: parameters $(\epsilon_t)_{t\in\mathbb{N}_0}$, number of particles n, initial distribution η_0, energy function V, annealing scheme $(\beta_t)_{t\in\mathbb{N}_0}$ and transitions $(K_t)_{t\in N_0}$

```
1. Initialization
   • Sample $x_0^{(i)}$ from $\eta_0$ for all $i$
2. Selection
   • Set $\pi^{(i)} \leftarrow \exp(-\beta_t\, V(x_t^{(i)}))$ for all $i$
   • For $i$ from 1 to $n$:
         Sample $\kappa$ from $U[0,1]$
         If $\kappa \leq \epsilon_t\pi^{(i)}$ then
           ⋆ Set $\check{x}_t^{(i)} \leftarrow x_t^{(i)}$
         Else
           ⋆ Set $\check{x}_t^{(i)} \leftarrow x_t^{(j)}$ with probability $\frac{\pi^{(j)}}{\sum_{k=1}^n \pi^{(k)}}$
3. Mutation
   • Sample $x_{t+1}^{(i)}$ from $K_t(\check{x}_t^{(i)}, \cdot)$ for all $i$ and go to step 2
```

In this case, the expression $\epsilon_t\pi^{(i)}$ in Algorithm 2 is replaced by $\pi^{(i)}/\sum_{k=1}^n \pi^{(k)}$. A third kernel is determined by

$$\epsilon_t(\eta_t) := \frac{1}{\inf\{y \in \mathbb{R} \,:\, \eta_t\left(\{x \in E \,:\, \exp(-\beta_t\, V(x)) > y\}\right) = 0\}}, \tag{6.9}$$

yielding the expression $\pi^{(i)}/\max_{1\leq k\leq n}\pi^{(k)}$ instead of $\epsilon_t\pi^{(i)}$.
Pierre del Moral showed in [Mor04, Chapter 9.4] that for any $t \in \mathbb{N}_0$ and $\varphi \in B(E)$ the sequence of random variables

$$\sqrt{n}(\langle\eta_t^n, \varphi\rangle - \langle\eta_t, \varphi\rangle)$$

converges in law to a Gaussian random variable W when the selection kernel (6.6) is used to approximate the flow (6.4). Moreover, it turns out that when (6.8) is chosen, the variance of W is strictly smaller than in the case with $\epsilon_t = 0$.
We remark that the annealed particle filter [DR05] relies on interacting simulated annealing with $\epsilon_t = 0$. The operation of the method is illustrated by

$$\eta_t^n \xrightarrow{Prediction} \hat{\eta}_{t+1}^n \xrightarrow{\quad ISA \quad} \eta_{t+1}^n. \tag{6.10}$$

The ISA is initialized by the predicted particles $\hat{X}_{t+1}^{(i)}$ and performs M times the selection and mutation steps. Afterwards the particles $X_{t+1}^{(i)}$ are obtained by an additional selection. This shows that the annealed particle filter uses a simulated annealing principle to locate the global minimum of a function V at each time step. However, the original algorithm in [DR05] contains two aspects that are not supported by interacting simulated annealing. First, the method uses a crossover operator for each mutation step, which makes the analysis difficult since the mutation kernels K_t depend then on the set of particles. Second, it does not perform annealing in the classical sense where the temperature is monotonically decreased, but relies on the fluctuating survival rate of the particles. This entails that divergence can be observed when the temperature increases in order to ensure a fixed survival rate. It typically occurs when the mutation kernels spread the particles broadly, i.e. when the global optimum is expected to be far away from the initialization. Hence, the original annealed particle filter cannot be used for global optimization and pose initialization, but the idea is preserved by the modification (6.10) where convergence can be proved.

Convergence

This section discusses the asymptotic behavior of the interacting simulated annealing algorithm. For this purpose, we introduce some definitions in accordance with [Mor04] and [Gid95].

Definition 6.1.1. The *Dobrushin contraction coefficient* of a kernel K on E is defined by

$$\beta(K) := \sup_{x_1,x_2 \in E} \sup_{B \in \mathcal{B}(E)} \left| K(x_1,B) - K(x_2,B) \right|. \tag{6.11}$$

Furthermore, $\beta(K) \in [0,1]$ and $\beta(K_1 K_2) \leq \beta(K_1)\, \beta(K_2)$.

When the kernel M is a composition of several mixing Markov kernels (3.29), i.e. $M := K_s K_{s+1} \ldots K_t$, and each kernel K_k satisfies the mixing condition for some ε_k, the Dobrushin contraction coefficient can be estimated by $\beta(M) \leq \prod_{k=s}^{t}(1-\varepsilon_k)$.
The asymptotic behavior of the interacting simulated annealing algorithm is affected by the convergence of the flow of the Feynman-Kac distribution (6.4) to the region of global minima of V as t tends to infinity and by the convergence of the particle approximation to the Feynman-Kac distribution at each time step t as the number of particles n tends to infinity.

Convergence of the flow We suppose that $K_t = K$ is a Markov kernel satisfying the mixing condition (3.29) for an $\varepsilon \in (0,1)$ and $\operatorname{osc}(V) < \infty$. A time mesh is defined by

$$t(n) := n(1 + \lfloor c(\varepsilon) \rfloor) \quad c(\varepsilon) := (1 - \ln(\varepsilon/2))/\varepsilon^2 \quad \text{for } n \in \mathbb{N}_0. \tag{6.12}$$

Let $0 \leq \beta_0 \leq \beta_1 \ldots$ be an annealing scheme such that $\beta_t = \beta_{t(n+1)}$ is constant in the interval $(t(n), t(n+1)]$. Furthermore, we denote by $\check{\eta}_t$ the Feynman-Kac distribution after the selection step, i.e. $\check{\eta}_t = \Psi_t(\eta_t)$. According to [Mor04, Proposition 6.3.2], we have

Theorem 6.1.2. *Let $b \in (0,1)$ and $\beta_{t(n+1)} = (n+1)^b$. Then for each $\delta > 0$*

$$\lim_{n \to \infty} \check{\eta}_{t(n)} \left(V \geq V_\star + \delta \right) = 0,$$

where $V_\star = \sup\{v \geq 0;\ V \geq v \text{ a.e.}\}$.

The rate of convergence is $d/n^{(1-b)}$ where d is increasing with respect to b and $c(\varepsilon)$ but does not depend on n as given in [Mor04, Theorem 6.3.1]. This theorem establishes that the flow of the Feynman-Kac distribution $\check{\eta}_t$ becomes concentrated in the region of global minima as $t \to +\infty$.

Convergence of the particle approximation Del Moral established the following convergence theorem [Mor04, Theorem 7.4.4].

Theorem 6.1.3. *For any* $\varphi \in B(E)$*,*

$$E_{\eta_0}\left[\left|\langle \eta_{t+1}^n, \varphi\rangle - \langle \eta_{t+1}, \varphi\rangle\right|\right] \leq \frac{2\,\mathrm{osc}(\varphi)}{\sqrt{n}}\left(1 + \sum_{s=0}^{t} r_s \beta(M_s)\right),$$

where

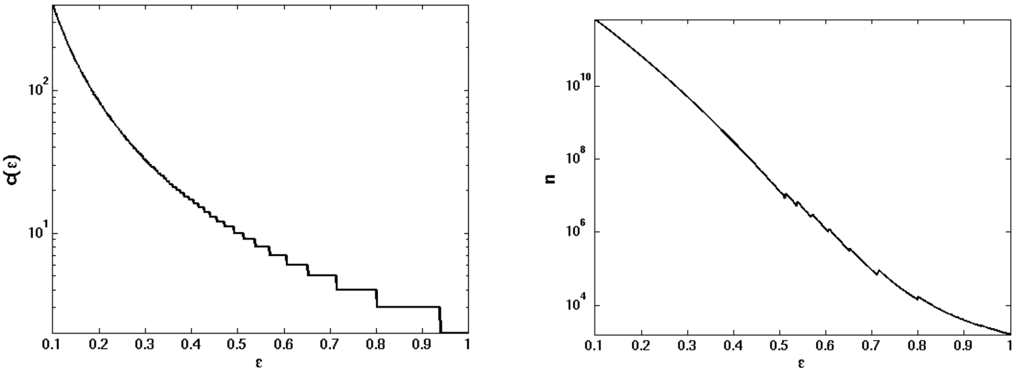
$$\begin{aligned} r_s &:= \exp\left(\mathrm{osc}(V)\sum_{r=s}^{t}\beta_r\right), \\ M_s &:= K_s K_{s+1}\dots K_t, \end{aligned}$$

for $0 \leq s \leq t$*.*

Assuming that the kernels K_s satisfy the mixing condition with ε_s, we get a rough estimate for the number of particles

$$n \geq \frac{4\,\mathrm{osc}(\varphi)^2}{\delta^2}\left(1 + \sum_{s=0}^{t}\left\{\exp\left(\mathrm{osc}(V)\sum_{r=s}^{t}\beta_r\right)\prod_{k=s}^{t}(1-\varepsilon_k)\right\}\right)^2 \tag{6.13}$$

needed to achieve a mean error less than a given $\delta > 0$.



(a) (b)

Figure 6.3: Impact of the mixing condition satisfied for $\varepsilon_s = \varepsilon$. **From left to right: a)** Parameter $c(\varepsilon)$ of the time mesh (6.12). **b)** Rough estimate for the number of particles needed to achieve a mean error less than $\delta = 0.1$.

Optimal transition kernel The mixing condition is not only essential for the convergence result of the flow as stated in Theorem 6.1.2, but also influences the time mesh by the parameter ε. In view of Equation (6.12), kernels with ε close to 1 are preferable, e.g. Gaussian kernels on a bounded set with a very high variance. The right hand side of (6.13) can also be minimized if Markov kernels K_s are chosen such that the mixing condition is satisfied for a ε_s close to 1, as shown in Figure 6.3. However, we have to consider two facts. First, the inequality in Theorem

6.1.3 provides an upper bound of the accumulated error of the particle approximation up to time $t+1$. It is clear that the accumulation of the error is reduced when the particles are highly diffused, but it also means that the information carried by the particles from the previous time steps is mostly lost by the mutation. Second, we cannot sample from the measure $\check{\eta}_t$ directly, instead we approximate it by n particles. Now the following problem arises. The mass of the measure concentrates on a small region of E on one hand and, on the other hand, the particles are spread over E if ε is large. As a result, we get a degenerated system where the weights of most of the particles are zero. Consequently, the global minima are estimated inaccurately, particularly for small n. If we choose a kernel with small ε in contrast, the convergence rate of the flow is very slow. Since neither of them is suitable in practice, we suggest a *dynamic variance scheme* instead of a fixed kernel K.

It can be implemented by Gaussian kernels K_t with covariance matrices Σ_t proportional to the sample covariance after resampling. That is, for a constant $c > 0$,

$$\Sigma_t = \frac{c}{n-1}\left(\rho I + \sum_{i=1}^{n}(x_t^{(i)} - \mu_t)\,(x_t^{(i)} - \mu_t)^T\right), \qquad \mu_t := \frac{1}{n}\sum_{i=1}^{n} x_t^{(i)}, \tag{6.14}$$

where I denotes the identity matrix and ρ is a small positive constant that ensures that the covariance does not become singular. The elements off the diagonal are usually set to zero, in order to reduce computation time.

Optimal parameters The computation cost of the interacting simulated annealing algorithm with n particles and T annealing runs is $O(n_T)$, where

$$n_T := n \cdot T. \tag{6.15}$$

While more particles give a better particle approximation of the Feynman-Kac distribution, the flow becomes more concentrated in the region of global minima as the number of annealing runs increases. Therefore, finding the optimal values is a trade-off between the convergence of the flow and the convergence of the particle approximation provided that n_T is fixed.

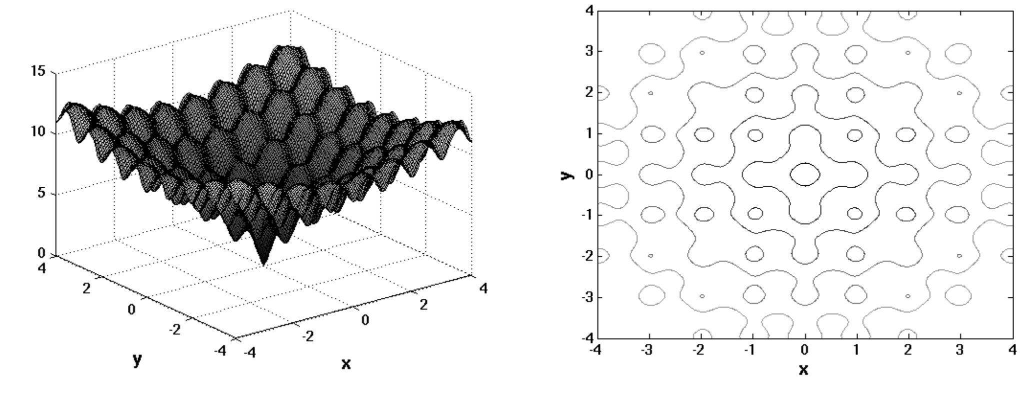
Another important parameter of the algorithm is the annealing scheme. The scheme given in Theorem 6.1.2 ensures convergence for any energy function V — even for the worst one in the sense of optimization — as long as $\operatorname{osc}(V) < \infty$, but it is too slow for most applications, as it is the case for simulated annealing. In our experiments the schemes

$$\beta_t = \ln(t+b) \quad \text{for some } b > 1 \qquad (\textit{logarithmic}), \tag{6.16}$$

$$\beta_t = (t+1)^b \quad \text{for some } b \in (0,1) \qquad (\textit{polynomial}) \tag{6.17}$$

performed well. Note that in contrast to the time mesh (6.12) the schemes are not anymore constant on a time interval.

In the following section, two examples are discussed that demonstrate settings that perform well, in particular for human motion capturing. A thorough evaluation of the various parameters on synthetic data is provided in Appendix A.



(a) (b)

Figure 6.4: Ackley function. Unique global minimum at $(0, 0)$ with several local minima around it.

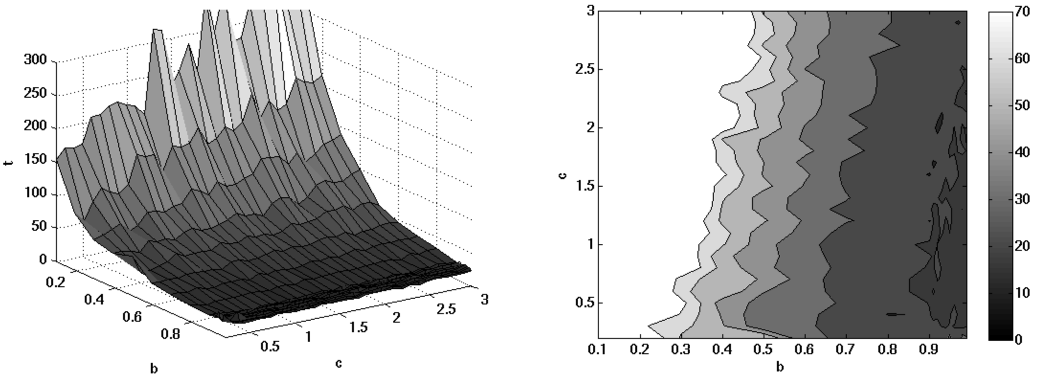
6.1.6 Examples

Global Optimization

The Ackley function [BS93, Ack87]

$$f(x) = -20\ \exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi\, x_i)\right) + 20 + e$$

on $\mathbb{R}^d$ is a widely used multimodal test function for global optimization algorithms. As one can see from Figure 6.4, the function has a global minimum at $(0, 0)$ that is surrounded by several local minima. The problem consists of finding the global minimum in a bounded subspace $E \subset \mathbb{R}^d$ with an error less than a given $\delta > 0$ where the initial distribution is the uniform distribution on E.



(a) (b)

Figure 6.5: Average time steps needed to find the global minimum with an error less than 10^{-3} with respect to the parameters b and c.

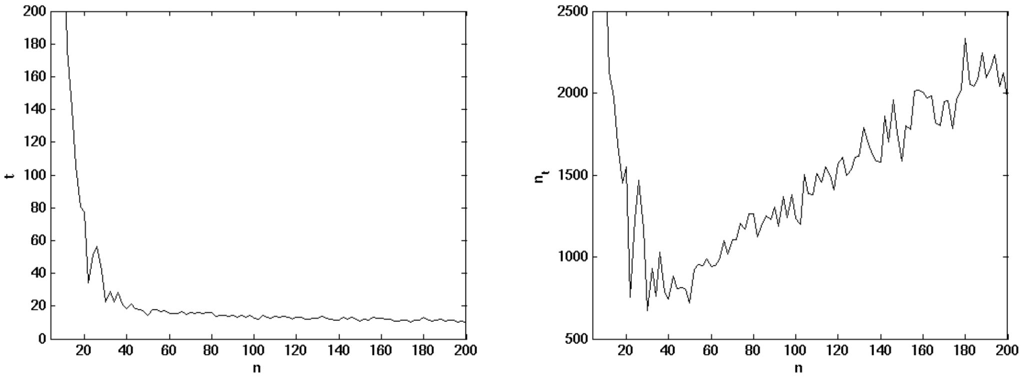
In our experiments, the maximal number of time steps are limited by 999 and we set $E = [-4, 4] \times [-4, 4]$ and $\delta = 10^{-3}$. The interacting simulated annealing algorithm is stopped when the Euclidean distance between the global minimum and its estimate is less than δ or when the limit of time steps is exceeded. All simulations have been repeated 50 times and the average number of time steps needed by ISA has been used for evaluating the performance of the algorithm. Depending on the chosen selection kernel (6.7), (6.8), and (6.9), we write ISA_{S1}, ISA_{S2}, and ISA_{S3}, respectively.

Using a polynomial annealing scheme (6.17), we have evaluated the average time steps needed by ISA_{S1} with 50 particles to find the global minimum of the Ackley function. The results with respect to the parameter of the annealing scheme, $b \in [0.1, 0.999]$, and the parameter of the dynamic variance scheme, $c \in [0.1, 3]$, are given in Figure 6.5. The algorithm performs best with a fast increasing annealing scheme, i.e. $b > 0.9$, and with c in the range $0.5 - 1.0$. The plots in Figure 6.5 also reveal that the annealing scheme has greater impact on the performance than the factor c. When the annealing scheme increases slowly, i.e. $b < 0.2$, the global minimum has not actually been located within the given limit for all 50 simulations.

	Ackley			Ackley with noise		
	ISA_{S1}	ISA_{S2}	ISA_{S3}	ISA_{S1}	ISA_{S2}	ISA_{S3}
b	0.993	0.987	0.984	0.25	0.35	0.27
c	0.8	0.7	0.7	0.7	0.7	0.9
t	14.34	15.14	14.58	7.36	7.54	7.5

Table 6.1: Parameters b and c with lowest average time t for different selection kernels.

The best results with parameters b and c for ISA_{S1}, ISA_{S2}, and ISA_{S3} are listed in Table 6.1. The optimal parameters for the three selection kernels are quite similar and the differences of the average time steps are marginal.



(a)

(b)

Figure 6.6: **From left to right: a)** Average time steps needed to find the global minimum with respect to number of particles. **b)** Computation cost.

In a second experiment, we have fixed the parameters b and c, where we have used the values from Table 6.1, and have varied the number of particles in the range from 4 to 200 with step size 2. The results for ISA_{S1} are shown in Figure 6.6. While the average of time steps declines rapidly for $n \leq 20$, it is hardly reduced for $n \geq 40$. Hence, n_t and consequently the computation cost are lowest in the range $20 - 40$. This shows that a minimum number of particles are required to achieve a success rate of 100%, i.e., the limit has not been exceeded for all simulations. In this example, the success rate has been 100% for $n \geq 10$. Furthermore, it indicates that the average of time steps is significantly higher for n less than the optimal number of particles. The results for ISA_{S1}, ISA_{S2}, and ISA_{S3} are quite similar. The best results are listed in Table 6.2.

	Ackley			Ackley with noise		
	ISA_{S1}	ISA_{S2}	ISA_{S3}	ISA_{S1}	ISA_{S2}	ISA_{S3}
n	30	30	28	50	50	26
t	22.4	20.3	21.54	7.36	7.54	12.54
n_t	672	609	603.12	368	377	326.04

Table 6.2: Number of particles with lowest average computation cost for different selection kernels.

The ability of dealing with noisy energy functions is one of the strength of ISA as we will demonstrate. This property is very useful for applications where the measurement of the energy of a particle is distorted by noise. On the left hand side of Figure 6.7, the Ackley function is distorted by Gaussian noise with standard deviation 0.5, i.e.,

$$f_W(x) := \max\{0, f(x) + W\}, \qquad W \sim N(0, 0.5^2).$$

As one can see, the noise deforms the shape of the function and changes the region of global minima. In our experiments, ISA has been stopped when the true global minimum at $(0, 0)$ has been found with an accuracy of $\delta = 0.01$. Note that δ is larger than in the experiment without noise.

For evaluating the parameters b and c, we set $n = 50$. As shown on the right hand side of Figure 6.7, the best results are obtained by annealing schemes with $b \in [0.22, 0.26]$ and $c \in [0.6, 0.9]$. In contrast to the undistorted Ackley function, annealing schemes that increase slowly have performed better than the fast one. Indeed, the success rate has dropped below 100% for $b \geq 0.5$. The reason is obvious from the left hand side of Figure 6.7. Due to the noise, the particles are more easily distracted and a fast annealing scheme diminishes the possibility of escaping from the local minima. The optimal parameters for the dynamic variance scheme are hardly affected by the noise.

The best parameters for ISA_{S1}, ISA_{S2}, and ISA_{S3} are listed in the Tables 6.1 and 6.2. Except for ISA_{S3}, the optimal number of particles is higher than it is the case for the simulations without noise. The minimal number of particles to achieve a success rate of 100% has also increased, e.g. 28 for ISA_{S1}. We remark that ISA_{S3} has required the least number of particles for a complete success rate, namely 4 for the undistorted energy function and 22 in the noisy case.

We finish this section by illustrating two examples of energy functions where the dynamic variance schemes might not be suitable. On the left hand side of Figure 6.8, an energy function with shape similar to the Ackley function is drawn. The dynamic variance schemes perform well for this type of function with a unique global minimum and several local minima around it. Due to the scheme, the search focuses on the region near the global minimum after some time steps. The second function, see Figure 6.8 b), has several, widely separated global minima yielding a

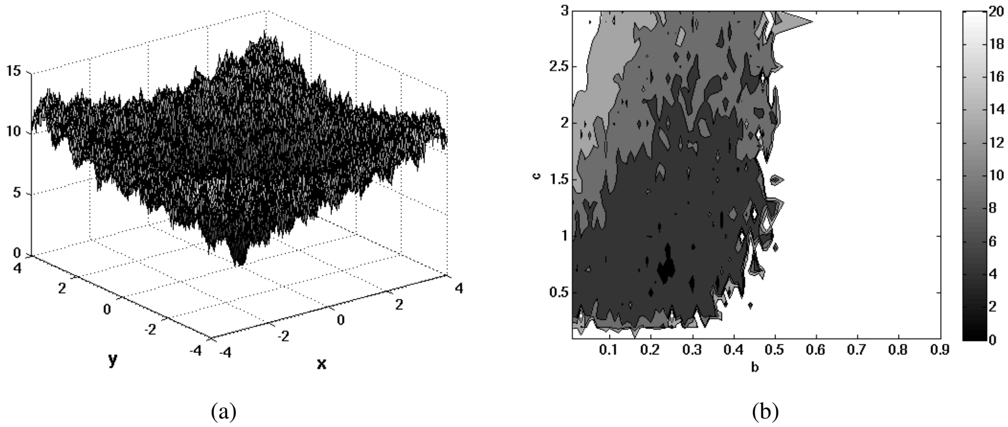


Figure 6.7: **From left to right: a)** Ackley function distorted by Gaussian noise with standard deviation 0.5. **b)** Average time steps needed to find the global minimum with an error less than 10^{-2} with respect to the parameters b and c.

high variance of the particles even in the case that the particles are near to the global minima. Moreover, when the region of global minima is regarded as a sum of Dirac measures, the mean is not essentially a global minimum. In the last example shown on the right hand side of Figure 6.8, the global minimum is a small peak far away from a broad basin with a local minimum. When all particles fall into the basin, the dynamic variance schemes focus the search on the region near the local minimum and it takes a long time to discover the global minimum.

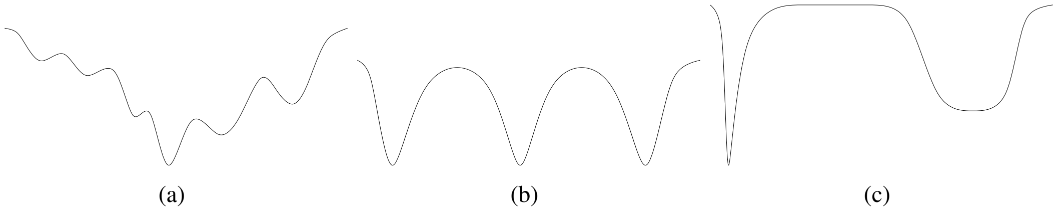


Figure 6.8: Different cases of energy functions. **From left to right: a)** Optimal for dynamic variance schemes. A unique global minimum with several local minima around it. **b)** Several global minima that are widely separated. This yields a high variance even in the case that the particles are near to the global minima. **c)** The global minimum is a small peak far away from a broad basin. When all particles fall into the basin, the dynamic variance schemes focus the search on the basin.

The first case, however, where the dynamic variance schemes perform well, is most relevant for optimization problems arising in the field of computer vision. One application is human motion capturing which we will discuss in the next section.

Human Motion Capture

In our second experiment, we apply the interacting simulated annealing algorithm to model-based 3D tracking of the lower part of a human body, see Figure 6.9 a). This means that each particle represents the 3D rigid body motion and the joint angles, also called the pose, using the

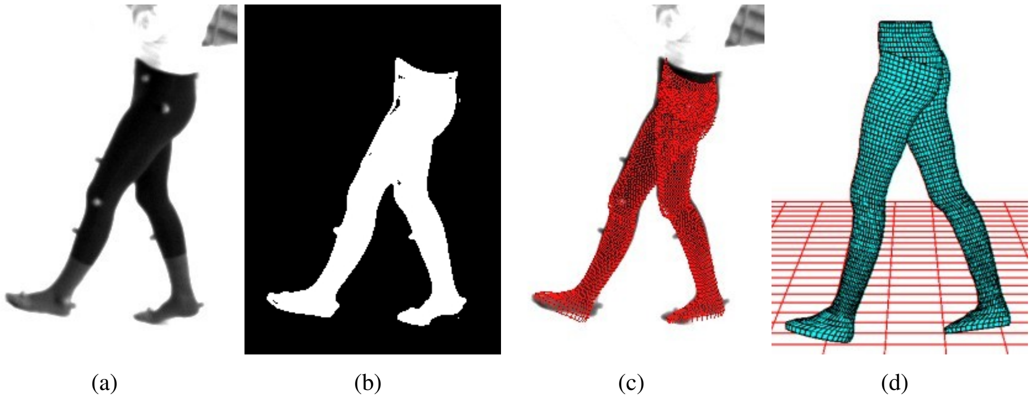


Figure 6.9: **From left to right: a)** Original image. **b)** Silhouette. **c)** Estimated pose. **d)** 3D model.

representation introduced in Section 3.1. The mesh model illustrated in Figure 6.9 d) has 18 degrees of freedom, namely 6 for the rigid body motion and 12 for the joint angles of the hip, knees, and feet. Although a markerless motion capture system is discussed, markers are also attached to the target object in order to provide a quantitative comparison with a commercial marker-based system.

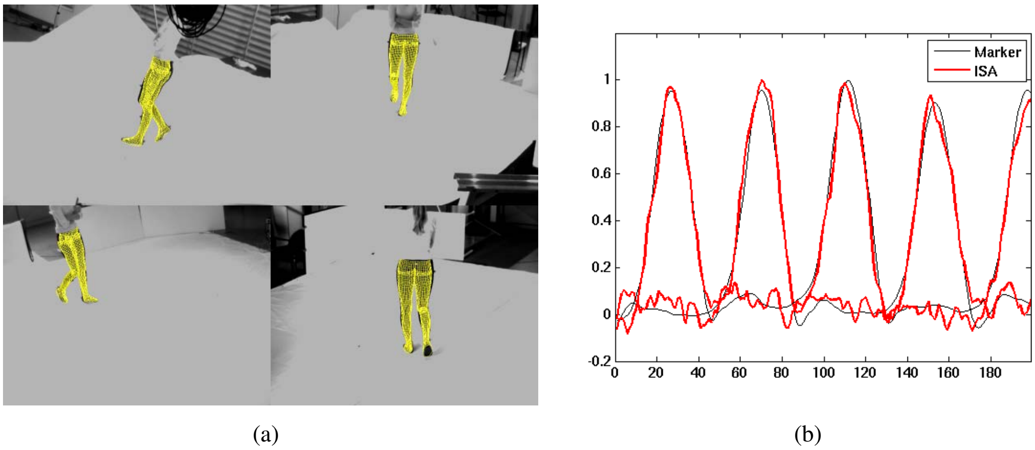


Figure 6.10: **From left to right: a)** Results for a walking sequence captured by four cameras (200 frames). **b)** The joint angles of the right and left knee in comparison with a marker-based system.

Using the extracted silhouette as shown in Figure 6.9 b), one can define an energy function V which measures the difference between the silhouette and an estimated pose. The pose that fits the silhouette best takes the global minimum of the energy function, which is searched by ISA. The estimated pose projected onto the image plane is displayed in Figure 6.9 c). This is a very naïve approach since the local optimization (Section 5.1) is actually replaced by global optimization to minimize the non-linear least squares problem (5.3). Correspondences are established by region-based matching (Section 5.2) and the pose is predicted for the next frame by an autoregression (Section 5.4.2). In addition, the learned probability of a static pose p_{pose} (Section 4.2.1) is incorporated to stabilize the pose estimation. Altogether, the energy function V of a particle

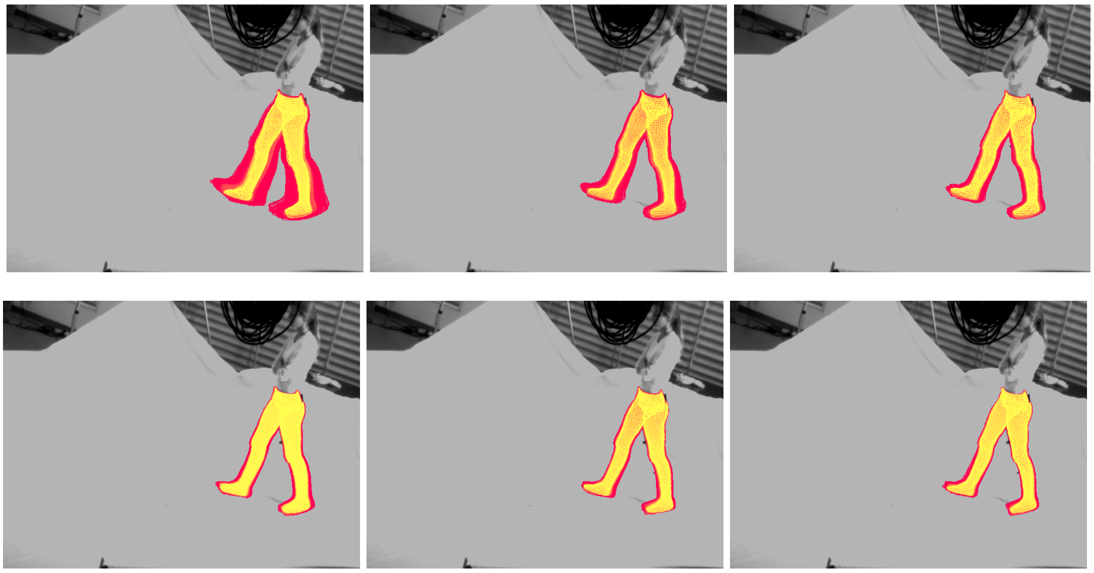
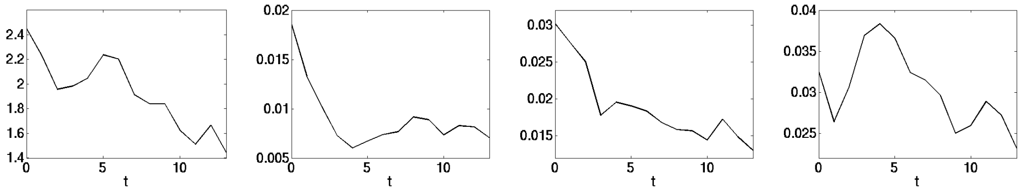


Figure 6.11: **From top left to bottom right:** Weighted particles at $t = 0$, 1, 2, 4, 8, and 14 of ISA. Particles with a higher weight are brighter and particles with a lower weight are darker. The particles converge to the pose with the lowest energy as t increases.



(a) Z-coordinate.

(b) Hip.

(c) Knee.

(d) Foot.

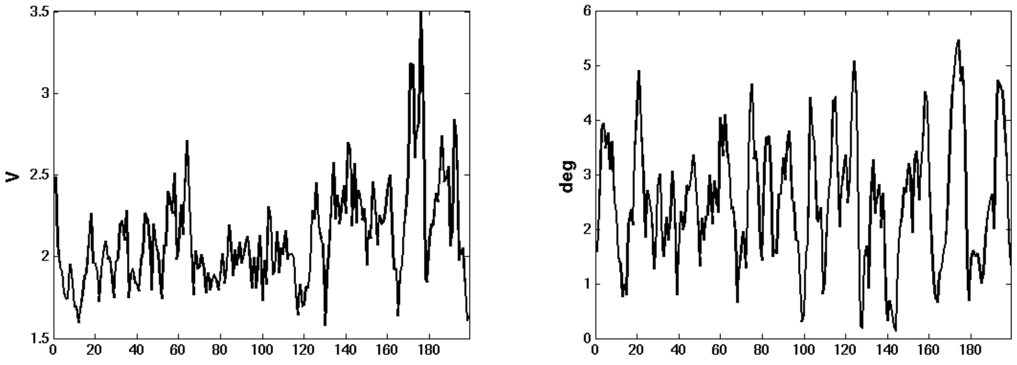
Figure 6.12: Variance of the particles during ISA. The scaled standard deviations for the z-coordinate of the position and for three joint angles are given. The variances decrease with an increasing number of annealing steps.

x is defined by

$$V(x) := \frac{1}{l}\sum_{i=1}^{l} err_S(x,i)^2 - \eta \ln(p_{pose}(x)), \tag{6.18}$$

where l is the number of correspondences and err_S is the error of a correspondence (5.2). The pose prior term is weighted by $\eta = 8$.

Results For evaluation, we have tracked the lower part of a human body using four calibrated and synchronized cameras. The walking sequence has been simultaneously captured by a commercial marker-based system [Mot08] allowing a quantitative error analysis. The training data used for learning p_{pose} consists of 480 samples that have been obtained from walking sequences. The data has been captured by the commercial marker-based system before recording the test sequence, which is not part of the training data.
The global optimization algorithm ISA performs well for the sequence consisting of 200 frames using a polynomial annealing scheme with $b = 0.7$, a dynamic variance scheme with $c = 0.3$,



(a)

(b)

Figure 6.13: **a)** Energy of estimate for walking sequence (200 frames). **b)** Error of estimate (left and right knee).

and the selection kernel (6.8). Results are given in Figure 6.10 where the estimated knee-joint angles are compared with the ground-truth from the marker-based system.

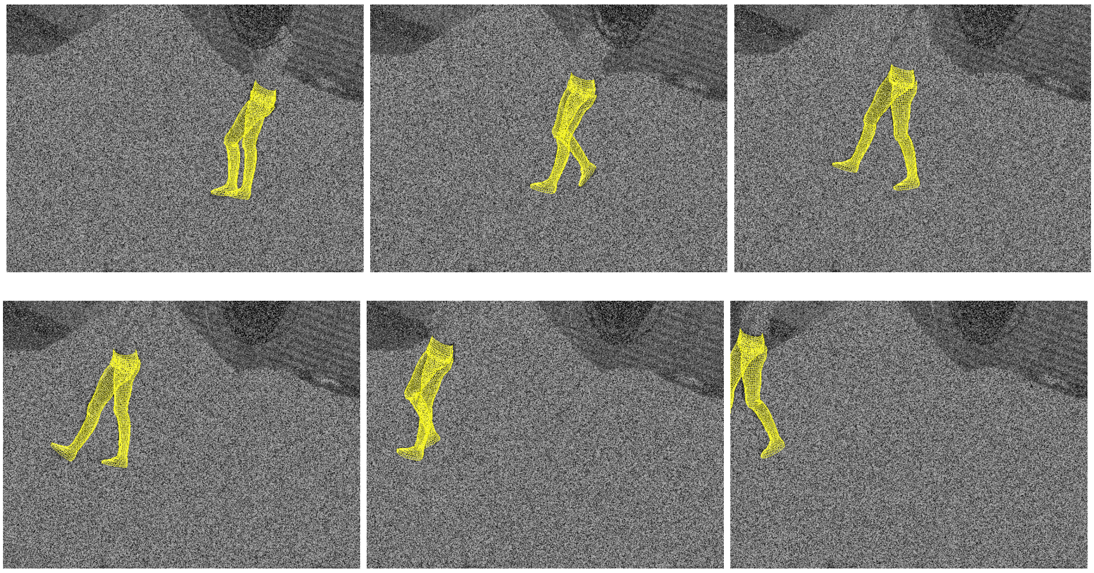


Figure 6.14: Estimates for a sequence distorted by 70% random pixel noise. One view of frames 35, 65, 95, 125, 155, and 185 is shown *(from top left to bottom right)*.

The convergence of the particles towards the pose with the lowest energy is illustrated for one frame in Figure 6.11. Moreover, it shows that the variance of the particles decreases with an increasing number of annealing steps. This can also be seen from Figure 6.12 where the standard deviations for four parameters, which are scaled by c, are plotted. While the variances of the hip-joint and the knee-joint decline rapidly, the variance of the ankle increases for the first iterations before it decreases. This behavior results from the kinematic chain of the legs. Since the ankle is the last joint in the chain, the energy for a correct ankle is only low when also the previous joints of the chain are well estimated.

On the right hand side of Figure 6.13, the energy of the estimate during tracking is plotted. We have also plotted the root-mean-square error of the estimated knee-angles for comparison where the results from the marker-based system are used as ground truth with an accuracy of 3 degrees. For $n = 250$ and $T = 15$, an overall root-mean-square error of 2.74 degrees is achieved. The error is still below 3 degrees with 375 particles and $T = 7$, i.e. $n_T = 2625$. With this setting, ISA requires $7 - 8$ seconds for approximately 3900 correspondences that are established in the 4 images of one frame. The whole system including segmentation, takes 61 seconds for one frame. For comparison, the iterative method discussed in Section 5.2 takes 59 seconds with an error of 2.4 degrees.

However, we have to remark that for this sequence the local optimization performs very well since the motion is simple and slow. As we have already mentioned, replacing local optimization by global optimization is very naïve since the correspondences are still extracted locally. More challenging sequences and advanced pose estimation frameworks are discussed in Sections 6.2 and 6.3. Nevertheless, it demonstrates that ISA can keep up even in situations that are perfect for local optimization methods.

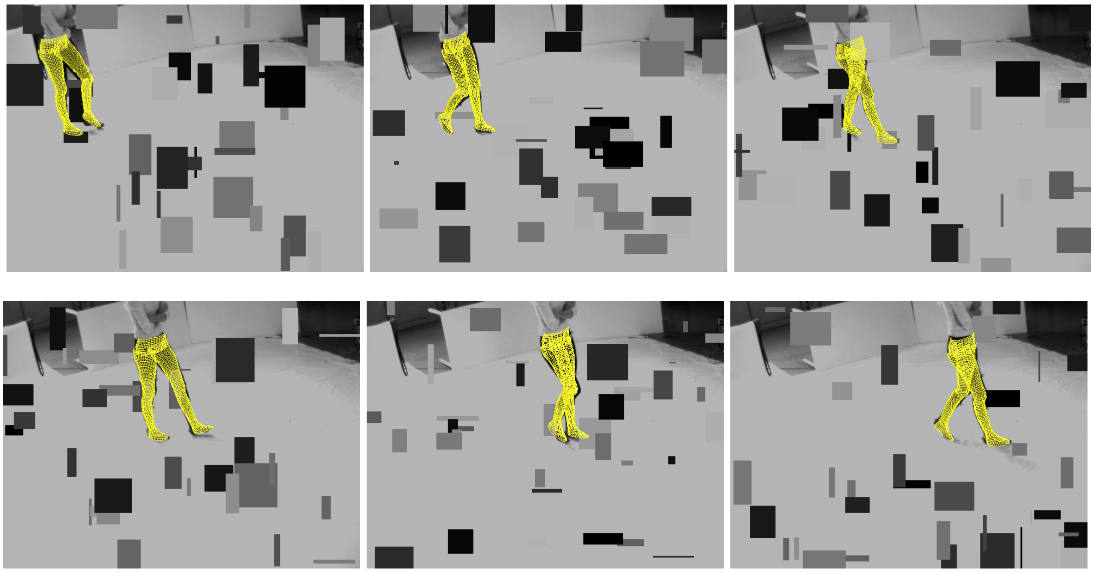
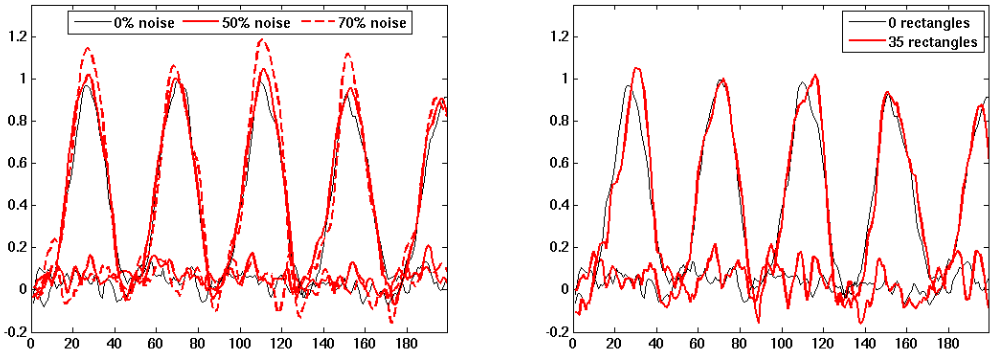


Figure 6.15: Estimates for a sequence with occlusions by 35 rectangles with random size, color, and position. One view of frames 35, 65, 95, 125, 155, and 185 is shown *(from top left to bottom right)*.

Figures 6.14 and 6.15 show the robustness in the presence of noise and occlusions. For the first sequence, each frame has been independently distorted by 70% pixel noise, i.e., each pixel value has been replaced with probability 0.7 by a value uniformly sampled from the interval $[0, 255]$. The second sequence has been distorted by occluding rectangles of random size, position, and gray value, where the edge lengths are in the range from 1 to 40. The knee angles are plotted in Figure 6.16. The root mean-square errors are 2.97 degrees, 4.51 degrees, and 5.21 degrees for 50% noise, 70% noise, and 35 occluding rectangles, respectively.



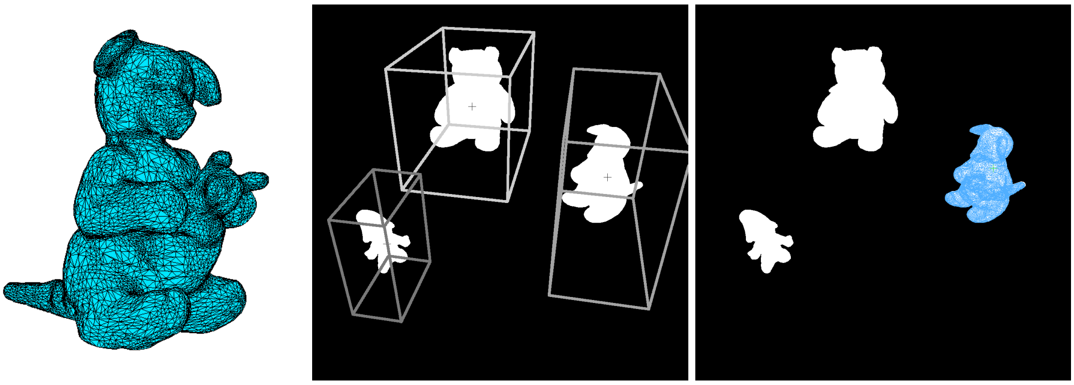
(a)

(b)

Figure 6.16: **a)** Random pixel noise. **b)** Occlusions by random rectangles.

6.2 Pose Initialization

Finding the 3D position and rotation of a rigid object in a set of images from calibrated cameras without any initial information is a difficult optimization problem in a 6-dimensional space. The task becomes even more challenging for articulated objects where the dimensionality of the search space is much higher, e.g., a coarse model of a human skeleton has already 24 degrees of freedom yielding a 30-dimensional space. Although the initial pose is essential for many state-of-the-art model-based tracking algorithm as discussed in Section 2.3, relatively little attention has been paid to the initialization of rigid and articulated models. A manual initialization is usually required, which is time demanding and assumes some expertise on the model and on the world coordinate system. Texture-based [LPF04, LLF05] and analysis-by-synthesis approaches (Sections 5.4 and 5.5) are an exception that use the surface texture for initialization. Apart from the fact that they require structured surfaces for self-initialization, the texture needs to be registered to the model beforehand, i.e., a manual initialization is done for the texture acquisition during preprocessing.



(a)

(b)

(c)

Figure 6.17: **From left to right:** **a)** 3D model of object. **b)** Potential bounded subsets of the search space. **c)** Projection of the mesh. The pose is correctly estimated.

Our approach for solving the initialization problem estimates the pose of rigid and articulated objects by minimizing an energy function based only on the silhouette information. Although we are not restricted to silhouettes, the object region has the advantage that it is an appearance independent feature that can be easily extracted from a single frame, e.g. by background subtraction. Since an initial guess is not available, local optimization algorithm like iterative closest point (ICP) [BM92, Zha94], see also Section 5.1, are not suitable for this task. For finding the exact pose, we use the particle-based global optimization *interacting simulated annealing* (ISA) that has been introduced in Section 6.1.5. In order to deal with multiple objects, we extend the algorithm by clustering the particles with respect to previously detected bounded subsets of the search space.

6.2.1 Global Optimization

In contrast to particle filter that estimate the posterior distribution for a sequence of images, we apply ISA for estimating the global optimum in still images where no initial information is available. For this purpose, the steps *Selection* and *Mutation* of Algorithm 2 are iterated until the global minimum of V is well approximated. During the selection, the particles are weighted according to a given energy function V where greater weight is given to particles with a lower energy. The weights associated to the particles refer to the probability that a particle is selected for the next step. We used the parameter $\epsilon_t = 1/\sum_{k=1}^{n} \pi^{(k)}$ for selection since it has slightly better convergence properties than $\epsilon_t = 0$, see Section 6.1.5. If a particle is not accepted with probability $\epsilon_t \pi^{(i)}$, a new particle is selected from all particles, e.g. by multinomial sampling. The selection process removes particles with a high energy while particles with a low energy are reproduced each time they are selected. An overview of various resampling schemes can be found in [DCM05]. In the second step, the selected particles are distributed according to Markov kernels K_t specified by a modified dynamic variance scheme, which we propose in Section 6.2.2.

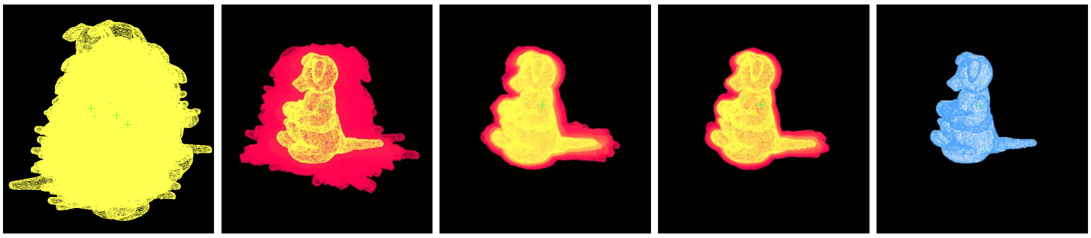


Figure 6.18: Particles at $t = 0, 5, 10, 15$, and 19 for ISA. Particles with a higher weight are brighter and particles with a lower weight are darker. The particles converge to the pose with the lowest energy as t increases. **Most left:** Equally weighted particles after initialization. **Most right:** Estimate after 20 iterations.

While the annealing scheme prevents the particles from getting stuck in local minima, the dynamic variance scheme focuses the search around selected particles. When t increases, only particles with low energy are selected and the search is concentrated on a small region, see also Figure 6.18. Indeed, Theorem 6.1.2 states that ISA approximates a distribution η_t that becomes concentrated in the region of global minima of V as t tends to infinity provided that the annealing scheme β_t increases slow enough and the search space is bounded. In our experiments, a polynomial scheme, i.e.

$$\beta_t = (t+1)^b \quad \text{for some } b \in (0,1), \tag{6.19}$$

performed well with $b = 0.7$.

6.2.2 Clustered Optimization

Initial Subsets

Having a binary image for each camera view, where pixels that belong to the foreground are set to 1 else to 0, the pixels are first clustered with respect to the 8-neighbor connectivity. In order to make the system more robust to noise, clusters covering only a very small area are discarded. In the next step, the 4 corners of the bounding box of each cluster are determined and the projection ray for each corner is calculated. The projection rays are represented as Plücker lines [Sto91], i.e., the 3D line is determined by a normalized vector n and a moment m such that $x \times n = m$ for all points x on the line. Provided that two projection rays from different views are not parallel, the midpoint p of the shortest line segment between the two rays l_1 and l_2 is unique and can be easily calculated. If the minimum distance between l_1 and l_2 is below a threshold, p is regarded as a corner of a convex polyhedron. After 8 corners of the polyhedron are detected for two clusters from two different views, the bounding cube is calculated as shown in Figure 6.17 b). In the case of more than two available camera views, each pair of images – starting with the views containing the most clusters – is checked until a polyhedron is found. The corners are similarly refined by calculating the midpoint of the shortest line segment between a ray from another view and a corner of a polyhedron. The resulting bounding cubes provide the initial bounded subsets of the search space. We remark that the algorithm is not very sensitive to the thresholds as long as the searched object is inside a bounding cube. This can be achieved by using very conservative thresholds.

Particles

Since we know the 3D model, the pose is determined by a vector in $\mathbb{R}^{6+m}$, i.e., each particle is a $6 + m$-dimensional random vector where m is the number of joints. The rigid body motion M is represented by the axis-angle representation given by the 6D vector $(\theta\omega, t)$ with $\omega = (\omega_1, \omega_2, \omega_3)$ and $\|\omega\|_2 = 1$. The mappings from $\theta\omega$ to a rotation matrix R and vice versa can be efficiently computed by the Rodriguez formula [MLS94] and are denoted by $\exp(\theta\omega)$ and $\log(R)$, see Section 3.1.

Since ISA approximates a distribution by a finite set of particles, we take the first moment of the distribution η_t as estimate of the pose, i.e., the mean of a set of rotations $r^{(i)}$ weighted by $\pi^{(i)}$ is required. To this end, the geodesic on the Riemannian manifold determined by the set of 3D rotations is computed according to Equation (3.5). The variance and the normal density on a Riemannian manifold can also be approximated, cf. [Pen06]. Since, however, the variance is only used for diffusing the particles, a very accurate approximation is not needed. Hence, the variance of a set of rotations r_i is calculated in the Euclidean space $\mathbb{R}^3$. Instead of taking the mean as estimate, the density could also be estimated from the set of particles by kernel smoothing in order to take the peak of the density function as estimate. However, kernel smoothing is more expensive than calculating the first moment of a density and it also needs to be performed in the space of 3D rotations.

Initialization

Due to multiple objects as shown in Figure 6.17, each particle belongs to a certain cluster C given by the bounding cubes and denoted by $x^{(i,C)}$. At the beginning, a small number of particles are generated with different orientations located in the center of the cube for each cluster. The complete set of particles is initialized by randomly assigning each particle the values of one of the generated particles. Afterwards, each particle is independently diffused by a normal distribution with mean $x^{(i,C)}$ and a diagonal covariance matrix with fixed entries except for the translation vector t where the standard deviations are given by the edge lengths of the cube divided by 6 such that over 99.5% of the particles are inside the cube.

Mutation

The dynamic variance scheme for the mutation step is implemented by cluster dependent Gaussian kernels $K_t^{(C)}$ with covariance matrices $\Sigma_t^{(C)}$ proportional to the sampling covariance matrix of each cluster:

$$\Sigma_t^{(C)} := \frac{d}{|C|-1}\left(\rho I + \sum_{\substack{i=1\\ i\in C}}^{n} (x_t^{(i,C)} - \mu_t)\,(x_t^{(i,C)} - \mu_t)^T\right), \quad \mu_t := \frac{1}{|C|}\sum_{\substack{i=1\\ i\in C}}^{n} x_t^{(i,C)}, \qquad (6.20)$$

where $|C|$ is the number of particles in cluster C and ρ is a small positive constant that ensures that the covariance does not become singular. In practice, we set $d = 0.4$ and compute only a sparse covariance matrix, see also Section 6.2.3. Samples from a multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ are drawn via a Cholesky decomposition $\Sigma = AA^T$, i.e., $x = \mu + Az$ where z is drawn from $\mathcal{N}(0, I)$.

Selection

Since each particle defines the pose of the model, the fitness of a particle $x \in \mathbb{R}^{6+m}$ can be evaluated by the difference between the original image and the template image that is the projected surface of the model. For this purpose, we apply a signed Euclidean distance transformation [FH04] on the silhouette image I_v and on the template $T_v(x)$ for each view v. The energy function is defined by $V(x) := \frac{\alpha}{r}\sum_{v=1}^{r} V_v(x)$ with

$$V_v(x) = \frac{1}{2|T_v^0(x)|}\sum_{p\in T_v^0(x)} |T_v(x,p) - I_v(p)| + \frac{1}{2|I_v^0|}\sum_{p\in I_v^0} |I_v(p) - T_v(x,p)|\,, \qquad (6.21)$$

where $I_v(p)$ and $T_v(x,p)$ are the pixel values for a pixel p and the sets of pixels inside the silhouettes are denoted by I_v^0 and $T_v^0(x)$. The normalization constant $\alpha = 0.1$ ensures that V is approximately in the range between 0 and 10, which is suitable for the selected annealing scheme.

The resampling step is cluster independent, i.e., the particles migrate to the most attractive cluster where the particles have more weight and give more offspring. At the end, there are no particles left where the silhouettes do not fit the model, see Figure 6.17 c).

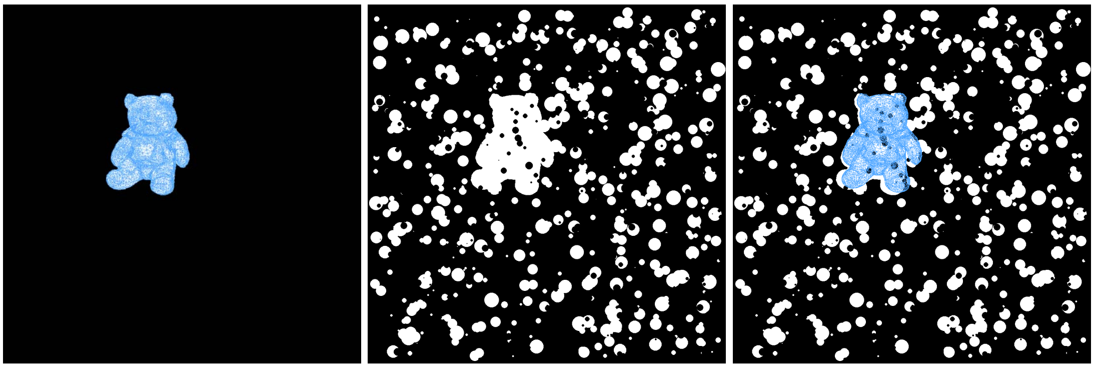


Figure 6.19: **From left to right: a)** Estimated pose without noise. The error is less than $1mm$ (median). **b)** Silhouettes are randomly distorted by 500 white and 500 black circles. **c)** Median estimate with error less than $4cm$.

6.2.3 Human Bodies

While for rigid bodies correlations between the parameters are neglected due to computational efficiency, correlations between connected joints in the human skeleton are incorporated. That is, correlations of the joints that belong to the same skeleton branch, e.g. the left leg, are calculated in the dynamic variance scheme (6.20) while correlations with joints to other branches are set to zero.

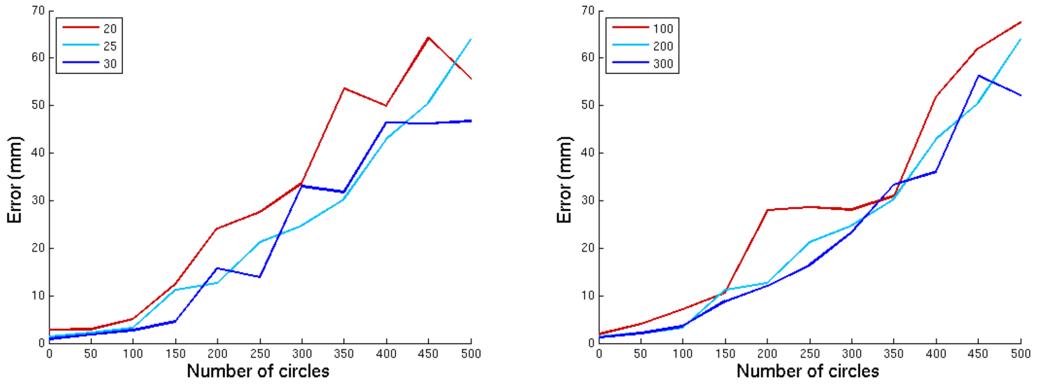
In order to focus the search on poses with higher probabilities, prior knowledge is incorporated into the energy function as soft constraint. The probability of a pose p_{pose} is estimated by a Parzen-Rosenblatt estimator with Gaussian kernels (4.2) over a set of subsamples from different motions from the CMU motion database [CMU08]. Since the dependency between the joints of the upper body and the joints of the lower body is low, the sample size can be reduced by splitting p_{pose} up into two independent probabilities p_{pose}^{upper} and p_{pose}^{lower}, respectively. Hence, the energy function is extended by

$$V(x) := \frac{\alpha}{r} \sum_{v=1}^{r} V_v(x) - \frac{\eta}{2} \ln \left(p_{pose}^{upper}(x) p_{pose}^{lower}(x) \right), \tag{6.22}$$

where $\eta = 2.0$ regulates the influence of the prior. Moreover, the mean and the variance of the joints in the training data are used to initialize the particles. To get rid of a biased error from the prior, the final pose is refined locally by ICP (Section 5.1) that is initialized by the estimate of ISA.

6.2.4 Results

For the error analysis, synthetic images with silhouettes of the bear have been generated by projecting the model onto 3 different views. The error has been measured by the Euclidean distance between the estimated 3D position and the exact position of the joints. Each simulation was repeated 25 times and the average errors for different numbers of particles and iterations are plotted in Figure 6.20. The estimates for 200 particles and 30 iterations are very accurate with a median error less than $1mm$, see Figure 6.19 a). The influence of distorted silhouettes is simulated by randomly drawing, first, a fixed number of white circles and then black circles.



(a)

(b)

Figure 6.20: Average error of the estimates for different numbers of iterations and 200 particles **(a)** and for different numbers of particles and 25 iterations **(b)**. 200 particles and 25 iterations are sufficient for rigid bodies.

Holes, dilatation, and erosion are typically for background subtraction and change the outcome of the Euclidean distance transform. The diagrams in Figure 6.20 show that our method performs also well for distorted silhouettes. In the case of 500 white and 500 black circles, the error of the median estimate shown in Figure 6.19 is still less than $4cm$.

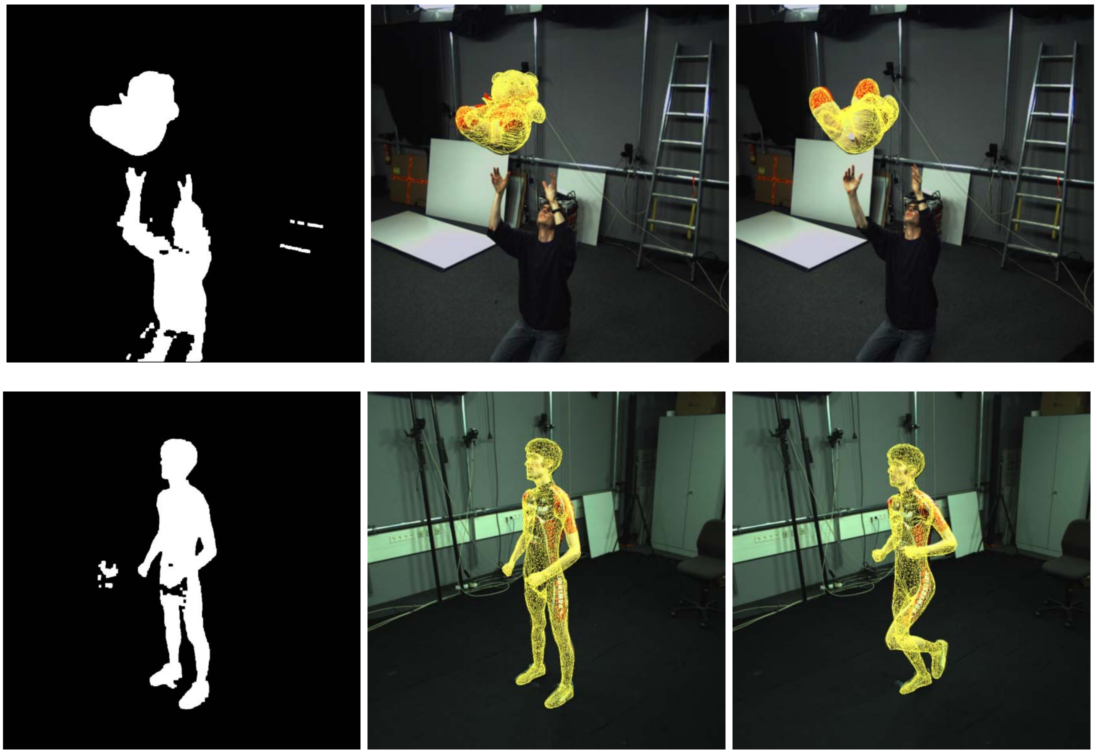


Figure 6.21: Estimates for a real scene. 3 views were segmented for the bear and 4 views for the human (only one is shown). **Most left:** Silhouettes from background subtraction.

The performance for a human body with 30 DOF has been tested by generating synthetic images with silhouettes for 12 single poses from a sequence of the CMU database that has not been used for the prior. The estimates are given in Figure 6.22. The average error of the joints for 400 particles and 40 iterations is 1.05°. Results for a real scene with background subtraction are shown in Figure 6.21. The method has also been used to initialize most of the sequences presented in this thesis and to acquire the texture for the objects shown in Section 5.5. For images of size 1004×1004 pixels, the computation cost is given by number of views × number of iterations × number of particles × 0.0346 seconds. When the signed Euclidean distance transform is replaced by a Chamfer distance transform [Bor86], the computation time can be reduced by 66% to 0.0117 seconds. An additional speed-up by a factor of 20 has been observed when the evaluation of the energy (6.21) is performed on a GPU.

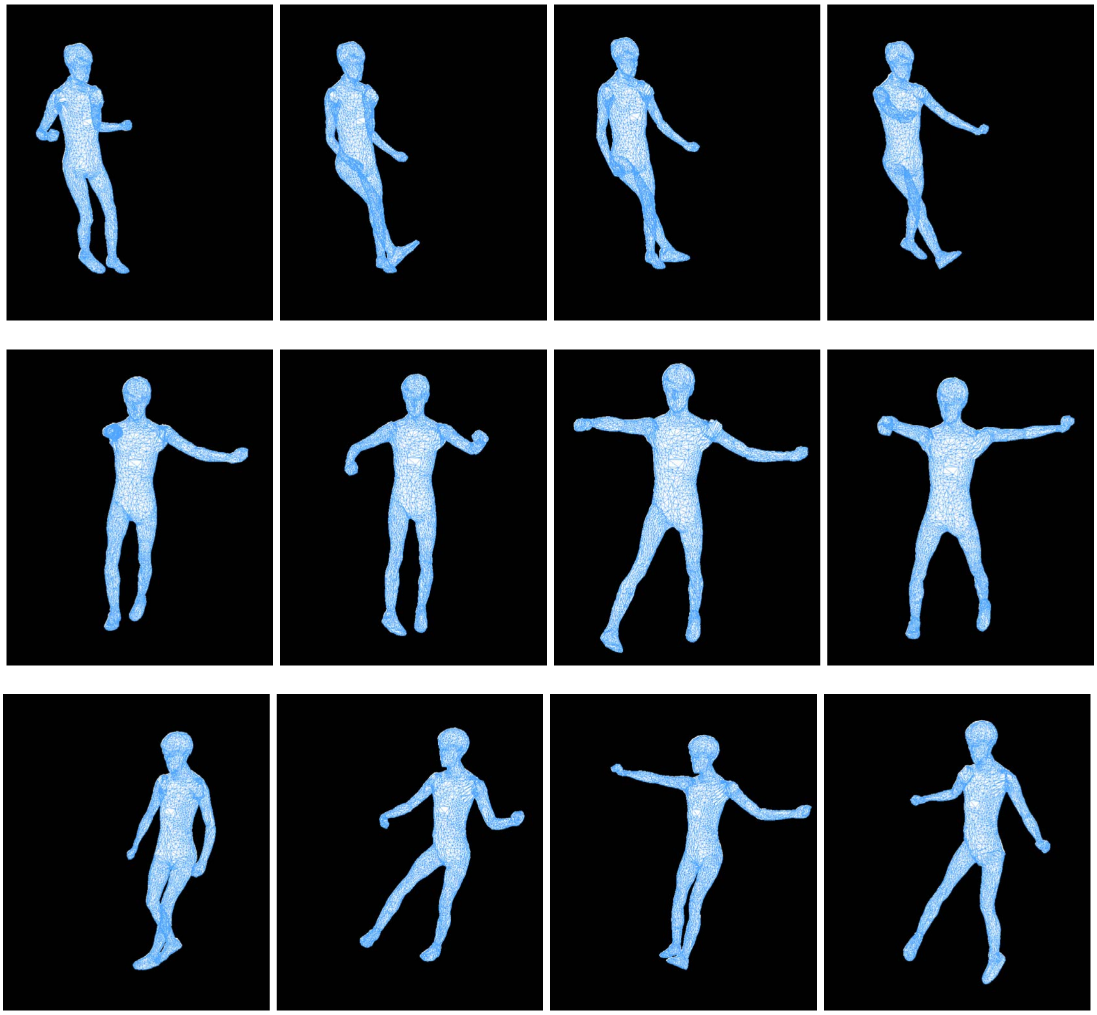


Figure 6.22: Estimates for 12 poses from a motion sequence from the CMU database. The estimated poses of the human model are projected onto the silhouette images. Each row shows one of the three views.

6.2.5 Summary

In this section, we have proposed an accurate and robust approach, which relies on a global optimization method with clustered particles, for estimating the 3D pose of rigid and articulated objects with up to 30 DOF. It does not require any initial information about position or orientation of the object and solves the initial problem as it occurs for tracking and texture acquisition. The experiments have demonstrated that the correct pose is estimated even when multiple objects appear. It could also be extended to the case when the object is not visible by rejecting estimates with a high energy. In general, our method can be easily modified for certain applications, e.g., by including prior as we have done for humans. Other possibilities are multi-cue integration and exploitation of a hierarchical structure. These features are, however, object specific and not suitable for a general solution.

6.3 Pose Tracking

In the previous section, the experiments have shown that global stochastic optimization performs well for pose estimation on still images. Although it is possible to apply the clustered optimization to each frame independently in order to estimate the poses for a whole sequence, this approach is very inefficient since the knowledge from the previous frame is not used. Another naïve approach for human motion capture that uses the framework from Chapter 5 and replaces the local optimization by global optimization does not outperform local optimization as shown in Section 6.1.6. Hence, we propose a novel tracking framework that embeds interacting simulated annealing and we compare it with several other optimization and particle filtering approaches. Generally speaking, we now address the question *"Is human motion capture a filtering or an optimization problem?"* from Section 1.5.2. To this end, we briefly recapitulate the advantages and disadvantages of filtering and optimization approaches.

Filtering methods are known to be robust and can recover from errors since they can model noise and resolve ambiguities over time. Particularly, particle filters are popular due to the multimodality of the solution since they approximate a distribution instead of a single value. Furthermore, they do not require linearity of the involved model like the Kalman filter. However, the available convergence results assume that the underlying stochastic processes are known – which in practice is rarely the case. Finding the right models for human motion tracking – both for the dynamics and for the likelihood – is very difficult and so far unsolved. Instead, the weakness of the models is often handled by overestimating the noise yielding a poor performance in high dimensional spaces as discussed in Chapter 4. Energy minimization approaches are usually more flexible with regard to the underlying model. However, local optimization suffers from local optima and usually cannot recover from errors. This has the effect that tracking fails when there are not strong enough cues to compensate for the drift as discussed in Chapter 5.

Since ISA approximates a distribution rather than a single value, similar to a particle filter, it inherits the advantages of filtering like multimodality and robustness to some extent. However, instead of approximating the posterior distribution, the distribution of interest concentrates its mass around the global optima as illustrated in Figure 6.23. Hence, we avoid the modeling problem of filtering approaches where the types of the involved distributions affect the posterior, and thus the outcome. Whereas for global optimization, the shape of the energy function is unimportant as long as the true state is close to the global optimum, which simplifies the modeling task as illustrated in Figure 1.3. Indeed, our experiments reveal a better accuracy of our framework than filtering approaches and more robustness than local optimization.

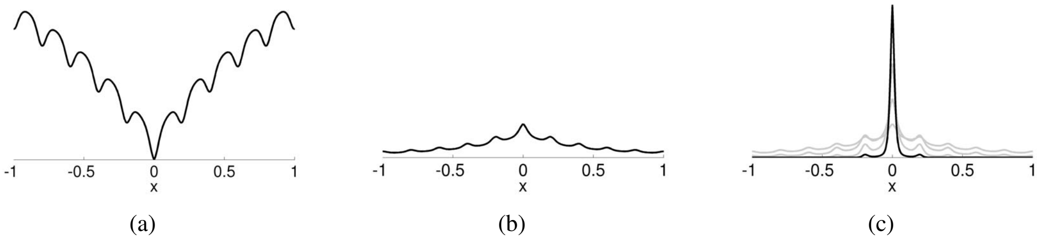


Figure 6.23: **From left to right: a)** Energy function V with global minimum at zero. **b)** η_1. **c)** The mass of η_t concentrates around the global minimum as t increases. For a limited number of iterations, η_t is multimodal.

We make use of the larger flexibility in the modeling by introducing an energy function that relies on silhouettes and color, as well as some prior information on physical constraints. In contrast to other works, which regard the appearance of the human for each view as independent, we estimate a statistical appearance model of the 3D surfaces of individual body parts using histogram representations. This makes the model more robust to 3D rotations than comparable 2D models as they are used, e.g., in [BB06].

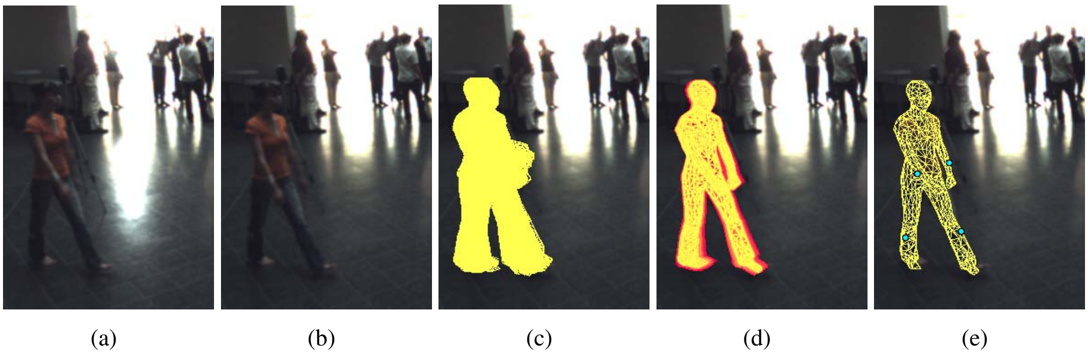
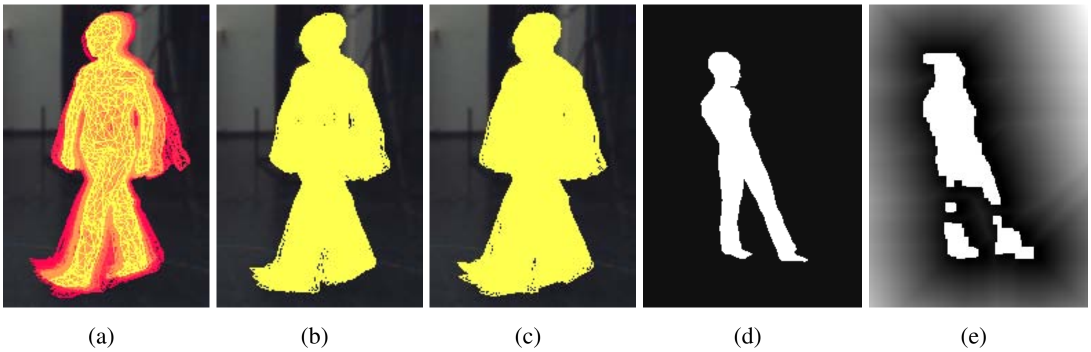


Figure 6.24: **From left to right: a, b)** Two successive frames of a multi-view video sequence with low contrast, rapidly changing illumination, and people in the background. **c)** A human specific mutation operator spreads particles in the search space. **d, e)** The pose is accurately estimated by global stochastic optimization. The cyan dots are estimates for the markers that were used for a quantitative error analysis.

A quantitative error analysis is performed to compare our approach with several optimization and particle filtering approaches. Our framework features automatic initialization and provides accurate estimates even in the case of video sequences with low contrast and challenging illumination, see Figure 6.24. Since neither excessive preprocessing nor strong assumptions are required, except a 3D model, it is a very general solution to human motion capture.

6.3.1 Pose Estimation by Global Optimization

Similar to Section 6.2, the weighting, selection, and mutation operation of Algorithm 2 are iterated T times, see Figure 6.25 a)-c). Finally, an estimate for the pose is obtained by the mean



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

Figure 6.25: The set of particles. **From left to right: a)** Weighting. Particles with higher weights are brighter. **b)** Selection. **c)** Mutation. **d)** Template image $T_v(x)$. **e)** Silhouette image I_v.

$\hat{x} = \int \eta_T^n(x)\, dx$.

Weighting. Assuming that a set of particles $(x_t^{(i)})_{i=1\dots n}$ exists, each particle is weighted by the Boltzmann-Gibbs measure

$$\pi^{(i)} = \exp\left(-\beta_t\, V\left(x_t^{(i)}\right)\right), \tag{6.23}$$

where $\beta_t = (t+1)^b$ with $b = 0.7$ is an annealing scheme that increases monotonically. After normalizing the weights such that $\sum_i \pi^{(i)} = 1$, the weight indicates the probability that a particle is selected for the next step.

Selection. The particles are accepted with probability $\pi^{(i)} / \max_k \pi^{(k)}$, i.e. the particle with the highest weight is always accepted. In order to leave the number of particles constant, the set of particles is completed by stratified resampling [DCM05] with acceptance probabilities $(\pi^{(i)})_i$. Due to the selection operation, similar particles with high weights are contained several times in the new set whereas particles with low weights might disappear completely. Since after this first stage only $m \leq n$ particles are selected, additional $n - m$ particles are drawn in a second stage, replacing those from the old set. This selection step corresponds to ISA_{S3} with selection kernel (6.9) that has performed slightly better on the Ackley test function than the other kernels ISA_{S1} and ISA_{S2}, see Table 6.2.

Mutation. In order to explore the search space, the particles are spread out according to a Gaussian K_t whose covariance matrix is the sampling covariance matrix

$$\Sigma_t = \frac{\alpha_\Sigma}{n-1}\left(\rho\, I + \sum_{i=1}^{n}(x_t^{(i)} - \mu_t)\,(x_t^{(i)} - \mu_t)^T\right), \tag{6.24}$$

scaled by $\alpha_\Sigma = 0.4$, where μ_t is the average, I the identity matrix, and ρ a small positive constant that ensures that the covariance does not become singular. The computational cost is reduced by using a sparse matrix that takes only correlations of joints into account that belong to the same skeleton branch. In general, the Gaussian distribution can be replaced by any distribution that satisfies the mixing condition to ensure the convergence on a bounded search space.

Energy Function

The energy of a particle x is calculated by

$$V(x) = \nu\, V_{silh}(x) + \tau\, V_{app}(x) + \upsilon\, V_{phys}(x), \tag{6.25}$$

where the parameters ν, τ, and υ control the influence of the three terms, namely silhouettes, appearance, and physical constraints.

Silhouettes. As in Section 6.2.2, a template image $T_v(x)$ is generated by projecting the surface of the human model that is translated, rotated, and deformed according to the particle x, see Figure 6.25 d). The inconsistent areas between a silhouette image I_v and the template $T_v(x)$ are measured for each view v by

$$V_v(x) = \frac{1}{2|T_v^0(x)|} \sum_{p \in T_v^0(x)} |T_v(x,p) - I_v(p)| + \frac{1}{2|I_v^0|} \sum_{p \in I_v^0} |I_v(p) - T_v(x,p)| , \tag{6.26}$$

where $I_v(p)$ and $T_v(x,p)$ are the pixel values for a pixel p and the sets of pixels inside the silhouettes are denoted by I_v^0 and $T_v^0(x)$. Since pixels that are far away from the silhouette should be penalized more severely, a Chamfer distance transform [Bor86] is previously applied to I_v as shown in Figure 6.25 e). In the optimal case, the Chamfer distance transform is also applied to the template $T_v(x)$, but this would be very expensive since the transform needs to be computed for each particle. Hence, we use only a constant value where pixels inside the silhouette are set to 0, as it is the case for the distance transform, and pixels outside the silhouette have a constant 'distance' to compensate for the differences between the error of the first and the second term of Equation (6.26). In our experiments, we have found that a value of 8 is a proper compensation factor. The energy term V_{silh} is finally defined as the average error of all views.

Appearance. Our approach for integrating color information is motivated by 2D segmentation where the separation of foreground pixels from the background relies on region statistics as discussed in Section 3.3. Since we know the 3D model, we combine the pixel information from all views to model the statistics of different body parts rather than their separate projections to the images. For efficiency reasons, we assume the image channels u_c to be uncorrelated. Hence, the joint probability density function for a body part s can be written as

$$p_s(u) = \prod_c p_{s,c}(u_c). \tag{6.27}$$

Instead of assuming a certain family of distribution functions, we approximate the probabilities $p_{s,c}$ in a more general manner by normalized histograms $H^{(s,c)}$ where we fixed the number of bins to $K = 64$. The updating of the appearance model during tracking is explained in Section 6.3.2.

In order to measure deviations of the appearance of a particle x from the appearance model given by $H^{(s,c)}$, the particle's appearance $\tilde{H}^{(s,c)}(x)$ is estimated by sampling from all views. For this purpose, the triangles of the human model are used to encode the body parts of the projected surface as shown in Figure 6.26 a). Hence, a pixel p that belongs to a body part s contributes for each channel u_c to the histogram $\tilde{H}^{(s,c)}(x)$. For histogram comparison, we choose the Bhattacharya distance since it is also stable for empty bins in contrast to χ^2 or KLD [PBRT99]. The total deviation is then measured according to (6.27) by

$$V_{app}(x) = \sum_s \frac{w_s}{C} \sum_{c=1}^{C} \left(1 - \sum_{k=1}^{K} \sqrt{h_k^{(s,c)} \tilde{h}_k^{(s,c)}(x)} \right) , \tag{6.28}$$

where the weights w_s reflect the size of the body parts and are determined during initialization, see Section 6.3.2.

Since the distinctiveness of the appearance model depends on the used image channels, the images are preprocessed to get a better image representation than the raw image data. We achieved good results with the CIELab color space that mimics the human perception of color differences. Since the L-channel is very sensitive to illumination changes, we used only the a- and b-channel. For small body parts like the hands where the sample sizes are rather small, image noise becomes an important issue. In order to reduce noise without smoothing over the edges that separate body parts, we apply the edge-enhancing diffusivity function [BRDW03]

$$g(|\nabla u|^2) = \frac{1}{|\nabla u|^p + \epsilon} \tag{6.29}$$

with $\epsilon = 0.001$ and $p = 1.5$, where the smoothing is efficiently implemented by the AOS scheme [WRV98].

Physical Constraints. Since human motion is subject to physical restrictions like anatomical constraints and self-intersections, the search can be focused on poses with higher probabilities by adding a soft constraint to the energy function. To this end, the probability of a skeleton deformation p_{pose} is estimated from a set of training samples y_l taken from the CMU motion database [CMU08] as in Section 6.2.3. Since self-intersections between the head, the upper body, and the lower body rarely occur, the sample size L can be reduced by regarding the probabilities for the three body parts, denoted by p_{pose}^{head}, p_{pose}^{upper}, and p_{pose}^{lower}, as uncorrelated. The probability for a body part is approximated by a Parzen-Rosenblatt estimator with a Gaussian kernel K (4.2):

$$p_{pose}(x) = \frac{1}{L\,h^d} \sum_l K\left(\frac{x - y_l}{h}\right), \tag{6.30}$$

where the d-dimensional vectors x and y_l contain only the joints for the body part. The bandwidth h is given by the maximum second nearest neighbor distance between all training samples. Finally, we used less than 200 samples from different motions for modeling the physical constraints by

$$V_{phys}(x) = -\frac{1}{3} \ln\left(p_{pose}^{head}(x) p_{pose}^{upper}(x) p_{pose}^{lower}(x)\right). \tag{6.31}$$

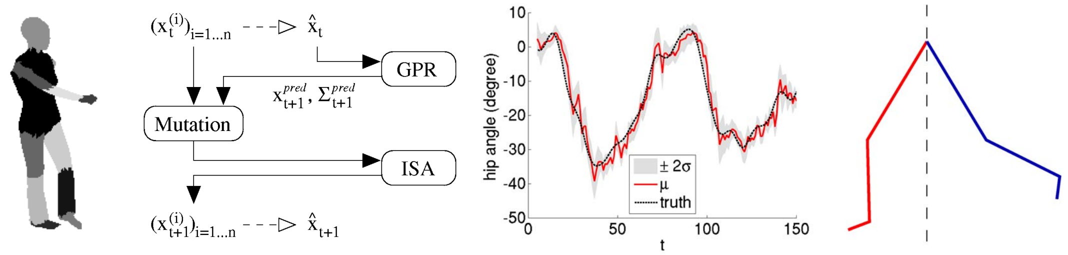
6.3.2 Tracking

For tracking, the pose estimation is embedded in a framework that takes advantage of temporal coherence of sequential data. An outline of the tracking system is given in Figure 6.26 b). For the first frame, the pose is detected automatically and the appearance model is initialized. Before estimating the pose via ISA, the particles are spread in the search space. After the optimization, the appearance model is updated.

Initialization

For finding the initial pose, ISA searches for the global minimum of the energy function defined in Equation (6.25) where only the terms V_{silh} and V_{phys} are used since the appearance of the model is unknown a priori. To this end, the search space is bounded by a cube that is determined by the silhouettes as described in Section 6.2.

After the pose $\hat{x}_0$ is estimated for the first frame, the histograms $H^{(s,c)}$ are generated by sampling from the images as described in Section 6.3.1. During sampling, the range of each feature channel is also determined and divided into uniform bins. Furthermore, the weights w_s in Equation (6.28) are given by the sample size for each body part s after normalizing such that $\sum_s w_s = 1$.



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

Figure 6.26: **From left to right:** **a)** Human model with 2000 triangles. The triangles encode the body parts. **b)** Outline of the tracking system. While the particle set $(x_t^{(i)})_i$ represents the distribution of the solution, the mean $\hat{x}_t$ provides a single estimate for the pose. The pose for the next frame x_{t+1}^{pred} is predicted by Gaussian process regression (GPR) and an additional mutation operator spreads the particles in the search space. The pose is then estimated by stochastic optimization (ISA). The system is closed in the sense that any uncertainty that arises from the prediction and estimation is preserved in terms of Σ_{t+1}^{pred} and $(x_{t+1}^{(i)})_i$. **c)** Prediction of a joint angle by GPR. Predicted Gaussian distribution with μ and σ. *d)* Mutation operator. The left branch *(red)* is reconstructed from the right branch *(blue)* by mirroring the first joint.

Mutation

After estimating the pose $\hat{x}_t$, the particles $x_t^{(i)}$ congregate around the global optima for frame t. Since this set is not well distributed for estimating the pose in the next frame, a mutation step spreads the particles in the search space. For this purpose, a 3rd order autoregression is used to predict the pose from the previous estimates, i.e. $x_{t+1}^{pred} = f(\hat{x}_{t:3})$ where we denote the last three estimates by $\hat{x}_{t:3} = (\hat{x}_t, \hat{x}_{t-1}, \hat{x}_{t-2})$. The function f can be learned online from the history of estimates given by the equations

$$\hat{x}_{t-r+1} = f(\hat{x}_{t-r:3}) \quad \text{for} \quad r = 1 \ldots R. \tag{6.32}$$

The regression is implemented by Gaussian processes [WR96] that fit very well in our framework since the prediction is given by a Gaussian distribution with mean x_{t+1}^{pred} and covariance matrix Σ_{t+1}^{pred}, see Figure 6.26 c). To simplify matters, we briefly summarize only the one-dimensional prediction by Gaussian processes where the set of training data is given by

$$\hat{x}_R = (\hat{x}_{t-1:3}, \ldots, \hat{x}_{t-R:3})^T \quad \text{and} \quad f(\hat{x}_R) = (f(\hat{x}_{t-1:3}), \ldots, f(\hat{x}_{t-R:3}))^T. \tag{6.33}$$

The predictive distribution for the last three estimates $\hat{x}_{t:3}$ is obtained by the conditional Gaussian distribution $p(\hat{x}_{t+1}|\hat{x}_{t:3}, \hat{x}_R, f(\hat{x}_R))$ with mean and variance

$$x_{t+1}^{pred} = k(\hat{x}_{t:3}, \hat{x}_R)^T \mathbf{K}^{-1} f(\hat{x}_R), \tag{6.34}$$

$$(\sigma_{t+1}^{pred})^2 = k(\hat{x}_{t:3}, \hat{x}_{t:3}) - k(\hat{x}_{t:3}, \hat{x}_R)^T \mathbf{K}^{-1} k(\hat{x}_{t:3}, \hat{x}_R). \tag{6.35}$$

The covariance matrix for the training data $\mathbf{K}$ is modeled by the general covariance function

$$\begin{aligned} k(\hat{x}_{r:3}, \hat{x}_{s:3}) = a_0 \exp & \left(-\frac{1}{2} \sum_{j=0}^{2} a_{j+1} \left(\hat{x}_{r-j} - \hat{x}_{s-j} \right)^2 \right) \\ & + \sum_{j=0}^{2} a_{j+4} \, \hat{x}_{r-j} \hat{x}_{s-j} + \sigma_{noise}^2 \delta_{rs}, \end{aligned} \tag{6.36}$$

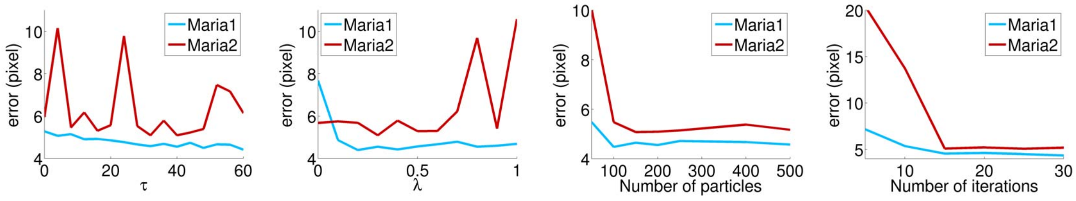
where the hyperparameters a_j and σ_{noise}^2 are learned offline by minimizing the log likelihood as proposed in [WR96]. For learning, we have used the sequences shown in rows 2-4 of Figure 5.34. Due to computational efficiency, all parameters of the search space are assumed to be independent which yields a one-dimensional prediction for each degree of freedom.
Since the dynamics are learned online, the prediction adapts to the current motion but it also might be corrupted by tracking errors in the past. Hence, we shift only 40% of the particles according to x_{t+1}^{pred}, another 40% is kept as it is, and 20% are mutated. The mutation is motivated by evolutionary algorithms where a larger variety among a population helps to recover from errors. We propose a human specific mutation operator that is useful when only one of two legs or arms is well estimated due to occlusions. In order to reconstruct its counterpart, we imitate the behavior of humans to use their arms or legs to balance. For this purpose, the first joint of the kinematic branch is mirrored while the other joint angles remain unchanged as illustrated in Figure 6.26 d). Even though the mutated particles will be mostly rejected after the first iterations of the optimization, they support the tracker in recovering from errors. Finally, all particles are propagated by a zero-mean Gaussian distribution with covariance matrix proportional to Σ_{t+1}^{pred}.
The prediction by Gaussian process regression has two advantages. When the movement is fast or the frame rate is low, x_{t+1}^{pred} guides some particles towards the next potential pose such that fewer iterations are required for optimization. More important, however, is Σ_{t+1}^{pred} which spreads the particles in the search space before optimization. Without the prediction, it would be necessary to set Σ_{t+1}^{pred} manually but the optimal values depend on the motion and the frame rate. GPR provides this information where the variance becomes larger for fast motions or a reduced frame rate. Note that we do not require a first order Markov process for the transitions as it is usually assumed for filtering approaches. In our experiments, we have observed that a 3rd order autoregression performs well for human motion whereas models with higher order improve only marginally the prediction.

Estimation

Having a well distributed set of particles, the pose is estimated by interacting simulated annealing as described in Section 6.3.1. Since many applications expect a single estimate for each frame, the weighted mean $\hat{x}_{t+1}$ is returned as estimate.

Update

After estimation, the covariance matrices for the regression are updated in Equations (6.34) and (6.35) by adding $\hat{x}_{t+1}$ to the history of estimates. Furthermore, the histograms $H^{(s,c)}$ are adapted to the changing appearance. First, a normalized histogram $\hat{H}^{(s,c)}$ is generated for $\hat{x}_{t+1}$



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

Figure 6.27: Tracking errors with respect to various parameters for sequences `Maria1` and `Maria2`. **From left to right: a)** Weight for appearance term V_{app}. **b)** Speed of adaption. **c)** Number of particles. **d)** Number of iterations.

by sampling from all views. The update for bin k is then given by

$$\frac{(1-\lambda)M^{(s)}\,h_k^{(s,c)} + \lambda\hat{M}^{(s)}\,\hat{h}_k^{(s,c)}}{(1-\lambda)M^{(s)} + \lambda\hat{M}^{(s)}}, \tag{6.37}$$

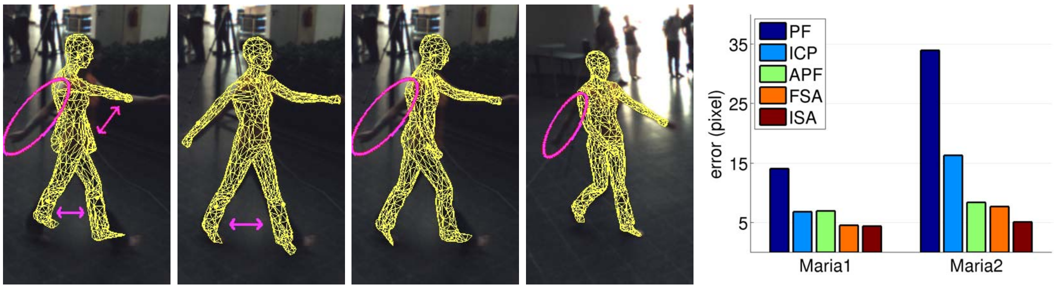
where $M^{(s)}$ and $\hat{M}^{(s)}$ are the sample sizes for the body part s to generate H and $\hat{H}$, respectively. The parameter λ controls the speed of adaptation and the consideration of the sample sizes avoids that the statistics are distorted by a small number of samples, e.g. due to self-occlusions.

6.3.3 Experiments

The first two rows of Figure 6.29 show estimates for the sequences `Maria1` and `Maria2`. Both sequences were captured by 5 synchronized and calibrated cameras with resolution of 640×480 pixels and 50 fps. They contain a walking person in a natural environment with people in the background, low contrast, motion blur, and challenging illumination changes as shown in Figure 6.24. In `Maria2`, the walking person additionally swings her arms. The human model is a low-resolution model of a 3D scan that consists of 2000 triangles. For a quantitative error analysis, circular markers with a diameter of approximately 5 pixels were attached to the forearms and lower legs and were tracked manually.

In order to register the 2D markers to the 3D model as accurate as possible, the first frame of each sequence was manually segmented. After estimating the pose for the first frame as described in Section 6.3.2, the nearest vertices of the projected mesh were used as 3D coordinates for the markers as illustrated by the cyan dots in Figure 6.29. The tracking error is then measured by the average 2D distance between the ground truth and the projected markers.

In our experiments, we fixed the parameters $\nu = 2.0$ and $\upsilon = 2.0$ in Equation (6.25) and we evaluated how sensitive our approach is with respect to the appearance parameters τ and λ as plotted in Figure 6.27. Unless otherwise stated, we used $\tau = 40$, $\lambda = 0.3$, 200 particles, and 15 iterations. The diagrams show clearly that the sequence `Maria2` is more challenging for tracking due to the dynamic movement of the arms. The resulting motion blur in the images affects the appearance of the arm and explains the increase of the error for large values of τ in contrast to the `Maria1` sequence. Good values for both sequences are in a broad range from 30 to 50. The optimal value for the speed of adaption λ depends on the environment. Figure 6.27 b), however, shows that the error is not very sensitive to the chosen value as long as the adaption is not too fast. The optimal numbers of particles and iterations are trade-offs between accuracy and computation cost. Figures 6.27 c) and d) show a significant decrease of the error until 100



(a) PF (b) ICP (c) APF (d) FSA (e)

Figure 6.28: A quantitative comparison with a particle filter (PF), local optimization (ICP), annealed particle filter (APF), and fast simulated annealing (FSA). Estimates for frame 94 or 18 of `Maria2` are shown for PF, ICP, APF, and FSA (*from left to right*). The estimates for ISA are given in row 2 of Figure 6.29. While our approach tracks the sequence without significant errors, the other approaches fail to estimate the barely visible right arm or swap the legs. **Rightmost:** The annealing approaches FSA and ISA perform best. For sequence `Maria2`, the error obtained by ISA increases only slightly and is significant lower than for APF and FSA, cf. Table 6.3.

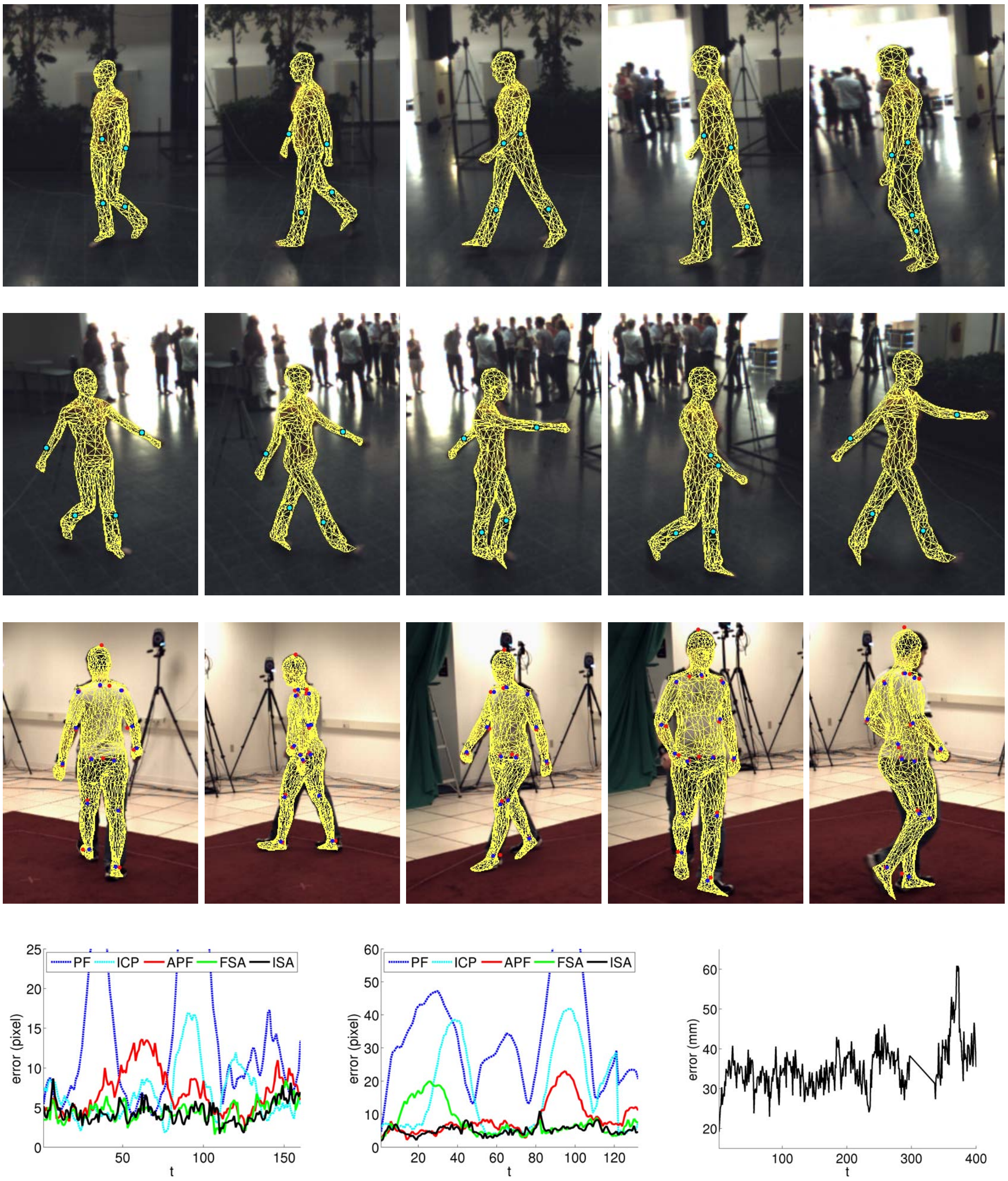
particles and 15 iterations yielding a computation time of 4 seconds per image. More particles or iterations improve the results only marginally.

For comparison with filtering and optimization, we applied a standard particle filter (PF), an iterative closest point approach (ICP) [BM92], annealed particle filtering (APF) [DR05], and fast simulated annealing (FSA) [CMC$^+$06] to the sequences. The same energy model was used. For the PF, we employed the weighting function (6.23) with $\beta_t = 1$. This is similar to the assumption that the likelihood is proportional to a product of normal densities. The particles are predicted as described in Section 6.3.2 without using the mutation operator since it is not supported by a filtering framework, i.e. 50% of the particles are shifted according to the predicted mean and the remaining 50% are directly selected. The number of particles and iterations was set to 3000 for PF and FSA, respectively, which yields the same computational effort as APF and ISA with 200 particles and 15 iterations. The results are plotted in Figure 6.28. The annealing approaches clearly outperform the local optimization and filtering methods. While the huge error of the PF indicates the weakness of the likelihood and dynamics, ICP gets stuck in local optima. FSA provides similar results as ISA for `Maria1`, however, the error significantly increases for `Maria2` whereas our approach performs well for both sequences, see Table 6.3. Since FSA cannot handle ambiguities, it lacks the robustness of ISA. The error for each frame is given in row 4 of Figure 6.29.

error(pix)	PF	ICP	APF	FSA	ISA
`Maria1` $\lambda = 0.2$	14.09 ± 9.95	6.81 ± 3.45	6.96 ± 2.74	4.56 ± 1.28	4.40 ± 1.26
`Maria2` $\lambda = 0.3$	33.96 ± 16.55	16.33 ± 12.88	8.40 ± 4.98	7.71 ± 4.69	5.09 ± 1.43

Table 6.3: Average error and standard deviation. For `Maria2`, the error is reduced by 39% and 34% compared to APF and FSA, respectively.

We also applied our approach to the dataset `HumanEva-II` [SB06] to measure the absolute



(p) `Maria1` (q) `Maria2` (r) $S4$

Figure 6.29: Error analysis. **Row 1:** Estimates for frames 45, 68, 91, 114, and 137 of `Maria1`. **Row 2:** Estimates for frames 18, 37, 56, 75, and 94 of `Maria2`. **Row 3:** Estimates for frames 80, 160, 240, 320, and 400 for subject $S4$ of `HumanEva-II`. **Row 4: p)** `Maria1`. The simulated annealing approaches FSA and ISA perform best. **q)** `Maria2`. Only ISA is able to track the sequence without significant errors. **r)** 3D tracking error for subject $S4$ obtained by our approach. The frames $298 - 335$ are neglected since the ground truth is corrupted for these frames.

3D tracking error. The available model is not perfect since it does not contain the clothing of the subject $S4$. Since the lighting conditions are controlled, we set $\lambda = 0$. Nevertheless, we achieve accurate estimates with 250 particles as shown in row 3 of Figure 6.29. Since the set-up and movement of the sequence, namely walking in a circle, is similar to the ones used in [BB06, XL07], we compare the results in Table 6.4. Even though these two approaches use body models with truncated cones, the comparison clearly indicates the superior accuracy of our tracking framework. We also compared ISA to the analysis-by-synthesis framework from Section 5.5. Since this approach relies on texture information, it benefits from the attached markers. Nevertheless, the standard deviations shown in Table 6.4 indicate a better performance of the global optimization. On the sequences `Maria1` and `Maria2`, the analysis-by-synthesis framework behaves like a standard ICP approach since there are not enough correspondences between the synthesized and the original image due to the low contrast. This shows that local optimization approaches like ICP only work well when strong cues are available. In practice, the analysis-by-synthesis framework is the method of choice for high resolution video footage whereas global optimization should be used for medium resolution videos (VGA or less).

	ISA	APF [BB06]	RBPF [XL07]	Drift-free (ICP)
error (mm)	35.02 ± 5.73	> 60	$51.66 - 148.67$	36.16 ± 9.12

Table 6.4: The comparison with other tracking approaches based on annealed or Rao-Blackwellised particle filter reveals that our framework performs significantly better. The standard deviation of the estimates is also significantly lower than the one obtained by the analysis-by-synthesis framework (Section 5.5).

6.4 Summary

We have shown that global stochastic optimization is a promising alternative to existing filtering and local optimization approaches for markerless human motion capture. It not only solves the difficult initialization problem (Section 6.2) that is relevant for texture acquisition as well, but also applies to pose tracking (Section 6.3) where stable results are achieved with a remarkable accuracy. Indeed, a quantitative comparison with several optimization and particle filtering approaches has revealed that our tracking framework gives significantly better results even for challenging scenes where the silhouette information is unreliable. Since the framework is easy to implement and requires neither excessive preprocessing nor strong assumptions, it is a very general solution to human motion tracking that can be specialized further.

The experiments demonstrate that regarding human motion capture as optimization problem avoids the modeling problem of filtering approaches as discussed in Section 4.1. As long as the problem cannot be well approximated by the Equations (2.1) and (2.2), filtering approaches perform poorly for this task. We have also illustrated in Figure 4.1 that image noise is not particularly critical for model-based human motion capture. This has been supported by our experiments where accurate results have been achieved by global optimization despite significant errors in the silhouettes. Hence, human motion capture with skeleton-based shape model can be very well modeled as optimization problem.

Local optimization may perform better than global optimization for sequences where local optima are not critical - but this requires high resolution image data where strong cues can be extracted. Apart from that, global optimization is the method of choice where *interacting simulated annealing* takes a leading role. Since ISA approximates a distribution instead of a single value, it has several advantages for pose tracking in comparison to standard optimization techniques. The optimization can be initialized with a large set of hypotheses, up to one hypothesis for each particle. This is exploited by the mutation step between two frames (Section 6.3.2) and helps to recover from large errors or ambiguities as they occur for a small number of cameras. Furthermore, the system outlined in Figure 6.26 preserves the uncertainty that arises from the prediction and optimization and takes it into account for the next frame. This increases the robustness and efficiency of the approach compared to other techniques like fast simulated annealing as we have demonstrated in our experiments. The approach might also be applied to other problems where filtering or local optimization perform poorly.

Although global stochastic optimization performs very well compared to local optimization and filtering approaches, there are still some limitations that need to be mentioned. When the estimates are observed over time, some jitter is noticeable which is typical for stochastic approaches like ISA that sample from a distribution of interest. Variations between estimates of two frames might also occur, when the tracker recovers from an ambiguity in the previous frame. Moreover, Figures 6.27 c) and d) indicate that ISA provides estimates close to the global optimum in reasonable time, but more precise estimates are only achieved at a very high cost. Hence, one might consider using local optimization to increase the accuracy and filtering to reduce the noise as it will be discussed in the following section. Since local approaches are preferable on high quality footage and global approaches on medium or low quality data, it would be more convenient to have a hybrid approach that handles both scenarios instead of manually switching between local and global optimization depending on the data. Furthermore, we have to consider that the optimization fits the surface model to the image data. This means that not only the image data has an impact on the accuracy but also the human body model. Aside from the accuracy of the surface model, the used deformation with a standard skeleton (Figure 1.2 d)) imposes limitations on the accuracy since it only approximates the surface deformation. While the artifacts are relatively small for tight cloth, wide apparel like skirts might introduce significant errors that need to be handled.

7

High-Performance Tracking Systems

While the previous chapters treat filtering approaches, local optimization, and global optimization separately to evaluate the performance of the various strategies, this chapter focuses on high-performance tracking systems for human motion capture. Since none of the approaches proposed in Chapters 4, 5, and 6 provides a perfect solution to human motion capture, we propose to combine the basic ideas of the three concepts to overcome the limitations that are inherent within each approach. Since global stochastic optimization has been shown to be superior to other approaches, interacting simulated annealing is an essential part of the systems. Section 7.1 introduces a multi-layer framework that extends the approach from Section 6.3 to benefit from optimization and filtering. While the first layer relies on ISA and some weak prior information on physical constraints, the second layer refines the estimates by filtering and local optimization such that the accuracy is increased and ambiguities are resolved over time without imposing restrictions on the dynamics. The system is developed for motion analysis scenarios with noisy silhouette data and is rigorously evaluated on the `HumanEva-II` benchmark [SB06]. Section 7.2 addresses the problem of non-rigid surface deformations, e.g. caused by tissue or garment. The system recovers not only the movement of the skeleton, but also the possibly non-rigid temporal deformation of the 3D surface. While large scale deformations or fast movements are captured by the skeleton pose and approximate surface skinning, true small scale deformations or non-rigid garment motion are captured by fitting the surface to the silhouette. In order to perform global optimization efficiently on high quality footage, the tree structure of the skeleton is exploited to split the optimization problem into a local one and a lower dimensional global one.

7.1 Multi-layer Framework

The strategies for model-based pose estimation can be classified into global optimization, filtering, and local optimization as in Section 2.3. Since all these strategies have some drawbacks, we propose a multi-layer framework that employs the basic ideas of all three concepts.

Global Optimization Since stochastic global optimization (Chapter 6) searches for the globally best solution, it is also suitable for initialization of model-based approaches. Its ability to recover from errors and its precise estimates satisfy the requirements for the first layer where robustness and accuracy are essential. However, when the estimates are observed over time, some jitter is noticeable which is typical for stochastic approaches like ISA that sample from a distribution of interest. Variations between estimates of two frames might also occur when the tracker recovers from an ambiguity in the previous frame. Moreover, while stochastic global optimization provides estimates close to the global optimum in reasonable time, the ratio between accuracy and computation cost is unsatisfactory when more precise estimates are required, as we will show.

Filtering/Smoothing Filtering approaches (Chapter 4) estimate the unknown true state x_t from some noisy observations y_t. The filtering problem is typically solved by Kalman filtering [Kal60] or particle filtering [DFG01] where it is assumed that the underlying stochastic processes

$$x_{t+1} = f_t(x_t) + v_t, \tag{7.1}$$

$$y_t = h_t(x_t) + w_t \tag{7.2}$$

with noise v_t and w_t are known. Even though filtering approaches exploit temporal coherence, handle noise, and are able to recover from errors, they are usually too imprecise for motion analysis in high dimensional spaces. Since accurate models for f_t and h_t are rarely available, the model's weakness is compensated by overestimating the noise vectors v_t and w_t at the expense of poor performance.

Local Optimization Local optimization (Chapter 5) provides very accurate results provided that the state vector is initialized near the global optimum. Since it searches only for the locally best solution, it usually cannot recover from errors and requires an initialization. Without additional prior information, the tracking often fails in case of fast motions and ambiguities.

Multi-layer The idea of several layers has been used for tracking-by-detection approaches [FDLF07, RFZ07] that rely on a learned template model. Since the detection is usually limited to canonical poses like lateral walking, the human poses are only detected on a subset of frames. A second step is therefore required to interpolate or track between the detected frames. While the tracking is usually done offline since the detected poses are used to learn a subject specific appearance model, our framework processes the image data online or with a very short delay.

7.1.1 Overview

In this section, we propose a model-based approach for 3D human motion capture that meets important needs of motion analysis since it does not rely on prior knowledge of the dynamics. In order to increase the accuracy and resolve ambiguities over time without imposing restrictions on the dynamics, we introduce a multi-layer framework that combines global optimization, filtering, and local optimization. While the first layer relies on global stochastic optimization, the second layer refines the estimates by filtering and local optimization as outlined in Figure 7.1.

For the first layer, the images are processed and silhouettes are extracted (Section 7.1.2). Interacting simulated annealing initializes the tracker and estimates the pose for each frame by minimizing an image-based energy function that relies on silhouettes and color, as well as some weak prior on physical constraints (Section 7.1.3). Although the first layer provides a robust and relatively accurate estimate of the human pose in the current frame, the estimate is still corrupted by noise due to sampling and the unsteady quality of the image features. Besides the missing temporal consistency, some bias might have been introduced by the weak prior.

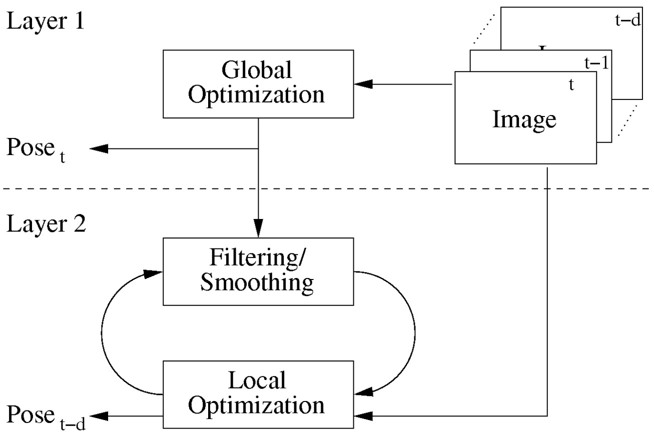


Figure 7.1: A multi-layer framework for tracking. While the first layer based on global stochastic optimization provides robust and relatively accurate estimates, the second layer increases the accuracy and reduces jitter and potential bias from the first layer with a short delay d.

The second layer refines the estimate with a short delay of $d \geq 0$ frames, where the estimate is filtered or smoothed (Section 7.1.4). Although the smoothing reduces the jitter from the stochastic global optimization by introducing temporal consistency, it improves only slightly the accuracy of the estimate. The latter is achieved by local optimization and segmentation where the smoothed estimate for frame $t-d$ serves as initial pose for optimization and as shape prior for the level-set segmentation (Section 7.1.5). The additional local segmentation improves the quality of the silhouettes of the first layer, which are obtained by global segmentation like background subtraction and often contain severe artifacts like shadows and holes. Since both segmentation and local optimization are initialized by good estimates from the first layer for each frame, an error accumulation due to the shape prior is prevented. We show that the second layer consisting of smoothing, local optimization, and local segmentation not only increases the accuracy, but also reduces jitter and potential bias from the first layer.
Indeed, our experimental evaluation in Section 7.1.6 demonstrates the improvements of the multi-layer framework in comparison to an increased number of iterations and samples for global optimization. It further comprises a quantitative error analysis using the `HumanEva-II` dataset [SB06], where we also compare interacting simulated annealing with particle filtering, annealed particle filtering, and local optimization.

7.1.2 Image processing

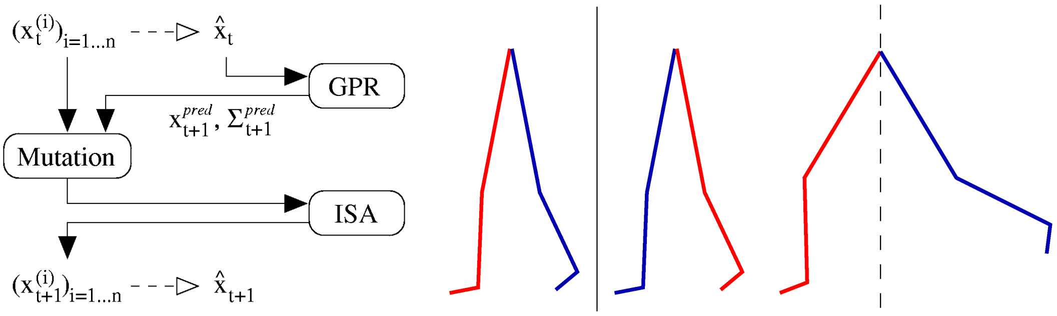
In our multi-layer framework, global and local optimization are applied to the same images, see Figure 7.1. Hence, the images need to be processed once such that they are suitable for the appearance model used for global optimization (Section 7.1.3) and the level-set segmentation in the second layer (Section 7.1.5). Both for segmentation and the appearance model, good results are obtained with the CIELab color space that mimics the human perception of color differences. In order to reduce noise without smoothing over the edges that separate body parts and background, we apply the edge-enhancing diffusivity function [BRDW03]

$$g(|\nabla u|^2) = \frac{1}{|\nabla u|^p + \epsilon} \tag{7.3}$$

with $\epsilon = 0.001$ and $p = 1.5$, where the smoothing is efficiently implemented by the AOS scheme [WRV98].

7.1.3 Global Optimization

The first layer of our tracking framework relies on the approach introduced in Section 6.3, where the pose $\hat{x}$ is obtained by searching for the global minimum of an energy function $V \geq 0$. During tracking the solution is represented by the set of particles $(x_t^{(i)})_i$ as outlined in Figure 7.2 a). Since the particles approximate a distribution, uncertainties from the pose estimation are propagated to the next frame making the estimation robust to ambiguities. An additional mutation operator between two frames spreads the particles in the search space where the predicted pose x_{t+1}^{pred} and its confidence Σ_{t+1}^{pred} are taken into account. The initial pose is also determined by ISA. Since the first layer is described in-depth in Section 6.3, we briefly mention relevant aspects and modifications for the multi-layer framework.

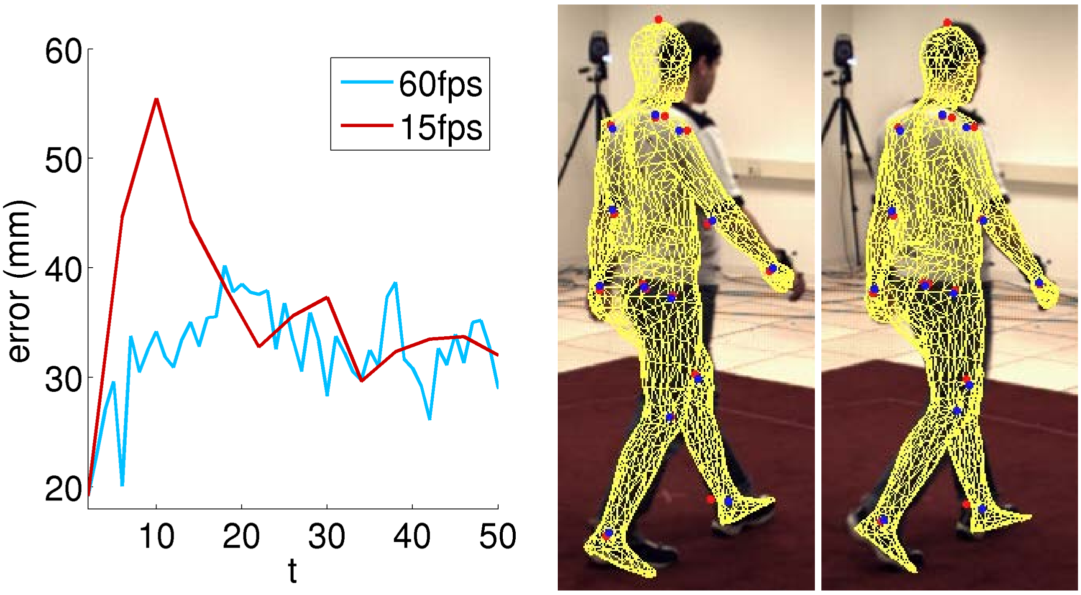


$(x_t^{(i)})_{i=1...n}$
$\hat{x}_t$
GPR
$x_{t+1}^{pred}, \Sigma_{t+1}^{pred}$
Mutation
ISA
$(x_{t+1}^{(i)})_{i=1...n}$
$\hat{x}_{t+1}$

(a) (b) (c)

Figure 7.2: **Left:** **a)** Outline of the first layer. While the particle set $(x_t^{(i)})_i$ represents the distribution of the solution, the mean $\hat{x}_t$ provides a single estimate for the pose. The pose for the next frame x_{t+1}^{pred} is predicted by Gaussian process regression (GPR) and an additional mutation operator spreads the particles in the search space. The pose is then estimated by stochastic optimization (ISA). **Right:** Two mutation operators. **b)** The left branch *(red)* and the right branch *(blue)* are swapped. **c)** The left branch *(red)* is reconstructed from the right branch *(blue)* by mirroring the first joint.

Mutation As in Section 6.3, a mutation module spreads the particles in the search space after the prediction by Gaussian process regression (GPR). However, we introduce an additional mutation operator that increases the variety among the particles in order to recover faster from errors. Hence, we shift 40% of the particles according to x_{t+1}^{pred}, another 30% is kept as it is, and 30% are mutated. We propose two human specific mutation operators as illustrated in Figures 7.2 b) and c). The first swaps two kinematic branches like the left and the right leg and helps to recover from ambiguous silhouettes which often occur when the legs are next to each other. The second is useful when only one of two legs or arms is well estimated due to occlusions. In order to reconstruct its counterpart, we imitate the behavior of humans to use their arms or legs to balance. For this purpose, the first joint of the kinematic branch is mirrored while the other joint angles remain unchanged. Even though the mutated particles will be mostly rejected after the first iterations of the optimization, they support the tracker in recovering from errors. Finally, all particles are propagated by a zero-mean Gaussian distribution with covariance matrix proportional to Σ_{t+1}^{pred}.



(a) (b) (c)

Figure 7.3: Impact of learning the motion model online. **From left to right: a)** To simulate the effect of a fast movement, only every 4th frame is used, i.e., the framerate of the camera is reduced from 60 fps to 15 fps. Since the dynamics are learned online, it takes some frames until good estimates for x_{t+1}^{pred} and Σ_{t+1}^{pred} are obtained. When the number of iterations for ISA remains unchanged, the error increases for the first frames. After the motion model is learned, the error is comparable to the 60 Hz sequence. **b)** Estimated pose for frame 3 of the 15 Hz sequence *(frame 10)*. **c)** After 5 frames at 15 Hz *(frame 18)*, the motion model is learned and the pose is well estimated.

Prediction Since the dynamics are learned online, the prediction adapts to the current motion but it also might be corrupted by tracking errors in the past. Particularly at the beginning, when not enough training samples are available, the prediction is usually poor. This is illustrated in Figure 7.3 where the error is higher for the first frames when the framerate is reduced. After a few frames, however, the motion model is learned and the error is comparable to the sequence with standard framerate. This shows that learning the motion online has the advantage that changes in motion or framerate can be handled without additional offline learning, parameter tuning, or large training data sets.

Energy As energy function for global optimization, we use

$$V(x) = \nu\, V_{silh}(x) + \tau\, V_{app}(x) + \upsilon\, V_{phys}(x), \tag{7.4}$$

where the parameters ν, τ, and υ control the influence of the three terms, namely silhouettes, appearance, and physical constraints that are explained in Section 6.3.1. Some of the used features are shown in Figure 7.4. Throughout this paper, we use the parameters $\nu = 2$, $\tau = 40$, and

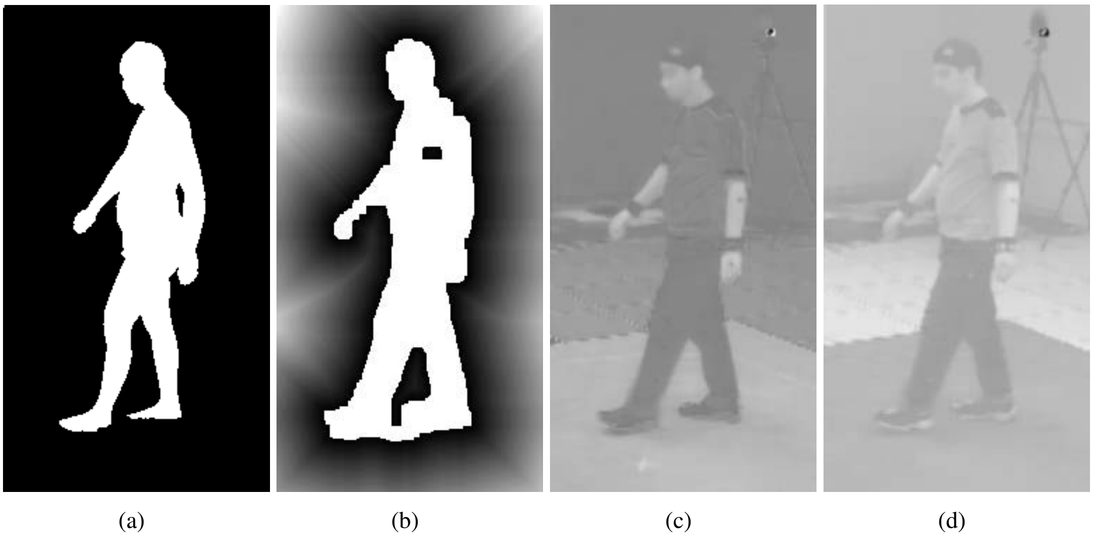


Figure 7.4: **From left to right: a)** Template image $T_v(x)$. **b)** Silhouette image I_v. **c)** Smoothed a-channel. **d)** Smoothed b-channel.

$v = 2$ that work well according to the evaluation in Section 6.3.3. In general, the appearance model needs to be updated during tracking as discussed in Section 6.3.2. However, when the lighting conditions are controlled as it is the case for the HumanEva-II dataset, an update is not necessary. We also want to emphasize that the term V_{phys} might still introduce some bias even though it is only a weak prior. Indeed, we will show that the bias can be reduced by the second layer.

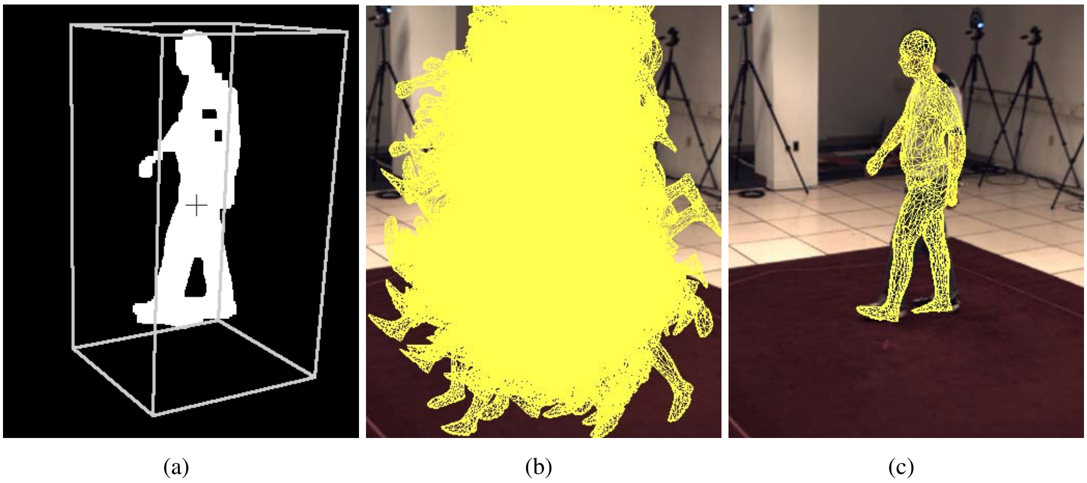


Figure 7.5: Initialization. **From left to right: a)** The search space is bounded by a cube. **b)** The initial set of particles is randomly distributed around the center of the cube. **c)** The pose is correctly initialized after 35 iterations.

Initialization The initial pose is obtained as in Section 6.3, where ISA searches for the global minimum of the energy function defined in Equation (6.25) using only the terms V_{silh} and V_{phys} since the appearance of the model is unknown a priori. First, the search space is bounded by a cube that is determined by the silhouettes as described in Section 6.2. The particles are then randomly distributed around the center of the cube and optimized by ISA, see Figure 7.5. After the pose $\hat{x}_0$ is estimated for the first frame, the appearance model is initialized as described in Section 6.3.2.

7.1.4 Smoothing

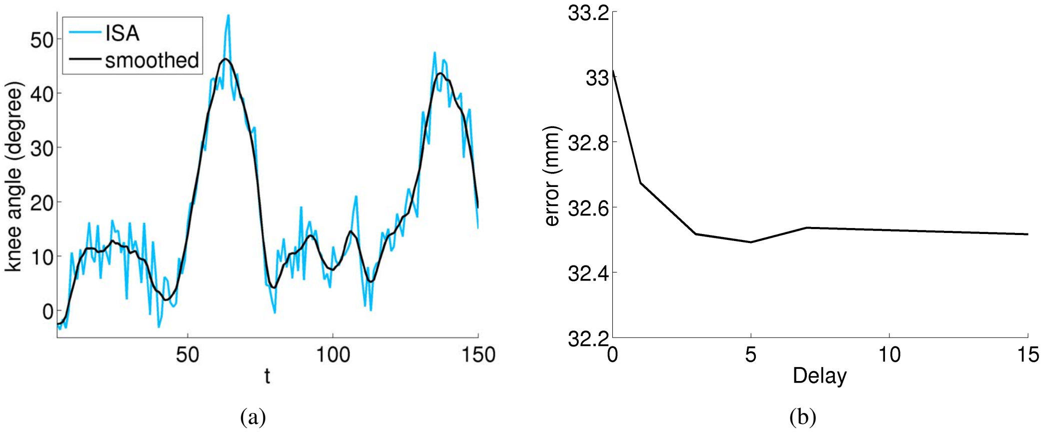


Figure 7.6: Impact of smoothing. **From left to right: a)** The smoothing reduces the jitter from global stochastic optimization. **b)** The absolute tracking error of the second layer with respect to the introduced delay d (Frames $2-821$ of sequence S4). The best result is achieved with a delay of only 5 frames. This corresponds to a delay of $83ms$ for a sequence with 60 fps. For $d = 0$, the estimates are filtered without delay.

Using the noisy mean estimates $\hat{x}_t$ from global optimization as observations instead of images, the filtering problem specified by Equations (7.1) and (7.2) is simplified such that h_t becomes the identity map. In addition, for considering the solutions of many frames for smoothing and not only a single one, we formulate the filtering as a regression problem.

As outlined in Figure 7.1, the second layer refines the estimates $\hat{x}_t$ from global optimization with a short delay of $d \geq 0$ frames by means of local optimization, as described later in Section 7.1.5. This yields more precise estimates x_t. Furthermore, we propose to couple regression and local optimization. Having R estimates

$$x_{t-R}, \ldots, x_{t-d-1}, \hat{x}_{t-d}, \ldots, \hat{x}_t, \tag{7.5}$$

we seek the function f that provides a smoothed version for frame $t-d$, i.e. $x_{t-d}^{smooth} = f(t-d)$. Since the refined values x_t should have more impact in the regression than the values $\hat{x}_t$, we add a binary indicator variable i_t as additional dimension to the input space. $i_t = 1$ indicates that the estimate has been already refined. The regressor $f(t, i_t)$ is then learned from the data

$$x_{t-r} = f(t-r, 1) \quad \text{for} \quad r = R \ldots d-1 \tag{7.6}$$

$$\hat{x}_{t-r} = f(t-r, 0) \quad \text{for} \quad r = d \ldots 0. \tag{7.7}$$

Similar to the prediction in Section 7.1.3, we apply Gaussian process regression. Let $\mathbf{t} := (t, i_t)$ and $\mathbf{t}_R := (\mathbf{t} - \mathbf{R}, \ldots, \mathbf{t})^T$. The smoothed estimate is then given by the mean

$$x_{t-d}^{smooth} = k\left((t-d,1), \mathbf{t}_R\right)^T \mathbf{K}^{-1} f\left(\mathbf{t}_R\right), \tag{7.8}$$

where the covariance matrix $\mathbf{K}$ is modeled by

$$\begin{aligned} k(\mathbf{t}-\mathbf{r}, \mathbf{t}-\mathbf{s}) =& a_0 \exp\left(-\frac{1}{2}\left(a_1 \left(r-s\right)^2 + a_2 \left(i_{t-r} - i_{t-s}\right)^2\right)\right) \\ &+ \sigma_{noise}^2 \delta_{rs}. \end{aligned} \tag{7.9}$$

The hyperparameters are learned offline as explained in Section 6.3.2. Since the correlation depends only on the temporal distance but not on the current value of t, $\mathbf{K}^{-1}$ needs to be calculated only once for a fixed number of training data R. Basically, the regression comes down to linear filtering with an asymmetric filter mask and the weights being learned from training data. Figure 7.6 shows the impact of d where we use $R = 10 + d$.
In general, a Kalman or particle filter could also be used for smoothing. However, the parameters need to be learned as well and we have not observed a significant improvement when the smoothing is performed with only a short delay.

7.1.5 Local Optimization

After smoothing, the accuracy of the estimated pose is increased by local optimization. Since the silhouettes from background subtraction often contain severe artifacts like shadows and holes, we improve the quality of the silhouettes by local segmentation before optimizing the pose, see Figure 7.7. The smoothed pose x_{t-d}^{smooth} serves both as shape prior for the segmentation and as initial estimate for local optimization. For reader's convenience, we briefly recapitulate the used local optimization and segmentation that are described in detail in Sections 5.1 and 5.2.

Local Segmentation

The silhouette of the human is extracted by a level-set segmentation that divides the image into fore- and background where the contour is given by the zero-line of a level-set function Φ. As proposed in [RBW07], the level-set function Φ is the minimum of the energy functional

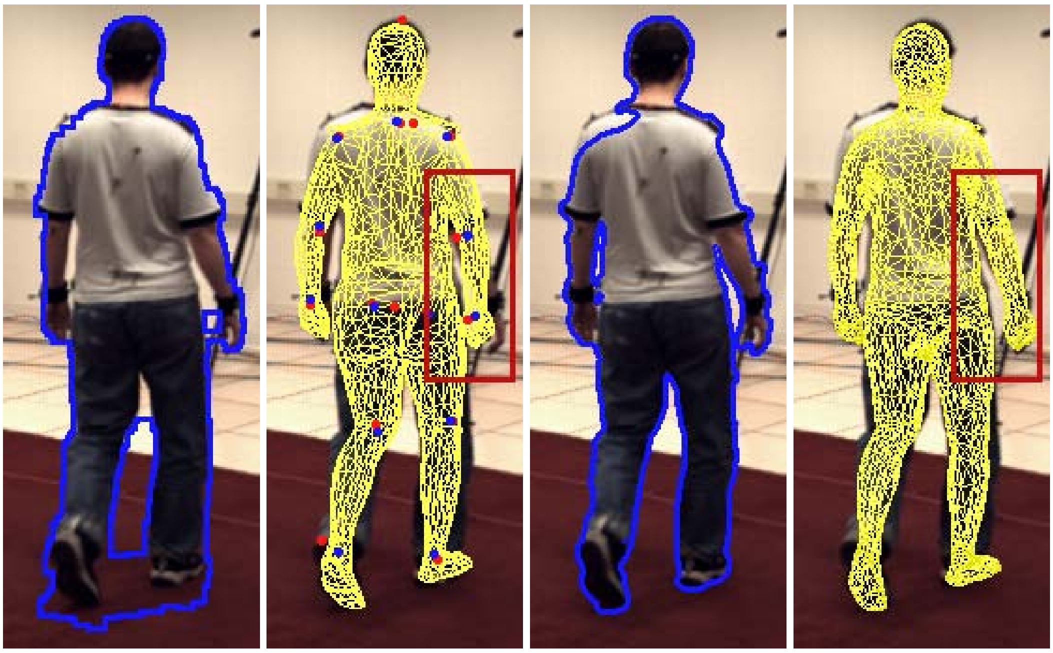
$$\begin{aligned} E(\Phi) = -\int_\Omega & H(\Phi) \ln p_1 + (1 - H(\Phi)) \ln p_2 \, dx \\ & + \vartheta \int_\Omega |\nabla H(\Phi)| \, dx + \lambda \int_\Omega (\Phi - \Phi_\chi)^2 \, dx, \end{aligned} \tag{7.10}$$

where H is a regularized version of the Heaviside step function. The probability densities of the fore- and background, p_1 and p_2, are modeled by local Gaussian densities (3.19) using the color channels L, a, and b that are assumed to be independent as in (6.27). While the first term maximizes the likelihood, the second term, weighted by the fixed parameter $\vartheta = 2$, regulates the smoothness of the contour. The last term penalizes deviations from the projected surface of the smoothed pose x_{t-d}^{smooth} given as level-set function Φ_χ (5.13), where the influence of the shape prior is controlled by the parameter $\lambda = 0.08$. For minimizing (7.10), local optimization is performed with gradient

$$\partial_k \Phi = H'(\Phi)\left(\ln \frac{p_1}{p_2} + \vartheta \operatorname{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)\right) + 2\,\lambda(\Phi_\chi - \Phi) \tag{7.11}$$

and Φ_χ as initial estimate. In contrast to the region-based matching approach in Section 5.2, segmentation and pose estimation are not iterated. Since a good estimate for the pose is already available, the shape prior Φ_χ is determined by x_{t-d}^{smooth} and remains unchanged for a frame.

Pose Estimation



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

Figure 7.7: **From left to right: a)** Silhouette from background subtraction. **b)** Estimate from global optimization. **c)** Silhouette from level-set segmentation. **d)** Improved estimate by local optimization. The right and left arms are better estimated.

The pose x_{t-d}^{smooth} is finally refined by the iterated closest point (ICP) approach introduced in Section 5.1. To this end, 2D-2D correspondences between the zero-level of Φ and $\Phi_\chi(x_{t-d}^{smooth})$ are established by a closest point algorithm [Zha94]. Since the points on the contour of the projected surface of x_{t-d}^{smooth} relate to 3D vertices of the mesh, 3D-2D correspondences between the model and the image can be derived. According to ICP, the pose estimation is performed iteratively where the set of correspondences is updated after each optimization until the pose converges to a local minimum.

For estimating the pose, we seek for the relative transformation that minimizes the error of given 3D-2D correspondences denoted by pairs (X_i, x_i) of homogeneous coordinates. As in Section 5.1, the rigid body motion is modeled as twist: $M = \exp(\theta\hat{\xi})$. A joint j is modeled as zero-pitch screw around a given axis, i.e., the joint motion depends only on the rotation angle θ_j. Hence, a transformation of a point X_i on the limb k_i influenced by n_{k_i} joints is given by

$$X_i' = M(\theta\hat{\xi})M(\theta_{\iota_{k_i}(1)})\ldots M(\theta_{\iota_{k_i}(n_{k_i})})X_i, \tag{7.12}$$

where the mapping ι_{k_i} represents the order of the joints in the kinematic chain. Since each 2D point x_i defines a projection ray that can be represented as Plücker line $L_i = (n_i, m_i)$ [Sto91], the error of a pair (X_i', x_i) is given by the norm of the perpendicular vector between the line L_i and the point X_i'

$$\|\Pi\left(X_i'\right) \times n_i - m_i\|_2, \tag{7.13}$$

where Π denotes the projection from homogeneous coordinates to non-homogeneous coordinates. Using the Taylor approximation $\exp(\theta\hat{\xi}) \approx I + \theta\hat{\xi}$, where I denotes the identity matrix, Equation (7.12) can be linearized. Hence, the sought transformation is obtained by solving the linear least squares problem

$$\underset{\chi}{\operatorname{argmin}} \frac{1}{2} \sum_i \left\| \Pi\left(\left(I + \theta\hat{\xi} + \sum_{j=1}^{n_{k_i}} \theta_{\iota_{k_i}(j)} \hat{\xi}_{\iota_{k_i}(j)} \right) X_i \right) \times n_i - m_i \right\|_2^2, \tag{7.14}$$

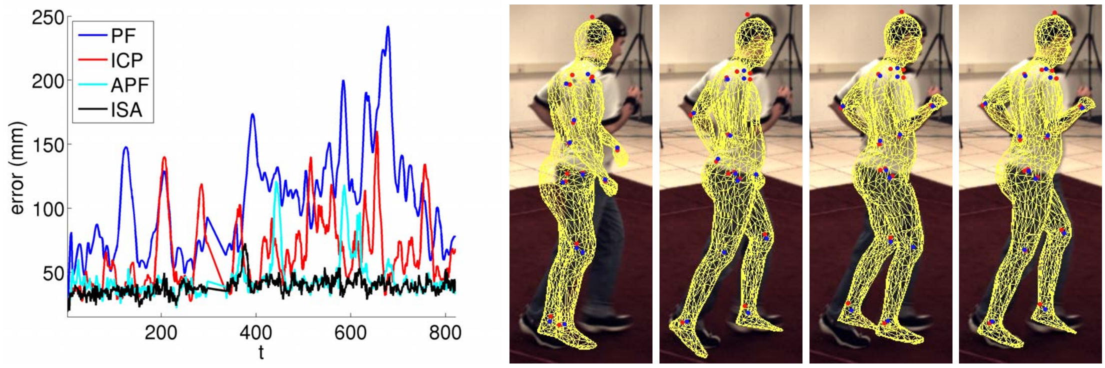
i.e. by solving a system of linear equations.
In order to penalize strong deviations from x_{t-d}^{smooth} and to avoid an underdetermined system, we extend the linear system by an additional equation

$$\alpha\theta_j = \alpha\left(\theta_j^{smooth} - \tilde{\theta}_j\right) \tag{7.15}$$

for each joint j, where $\tilde{\theta}_j$ is the previously estimated absolute joint angle. The parameter α is set relative to the number of correspondences to achieve a constant weighting for each frame. In practice, we use $\alpha = 0.2 \cdot |\{(X_i, x_i)_i\}|$. Since the local optimization provides only a relative transformation, the refined pose x_{t-d} is obtained by applying the relative transformation to the previously estimated pose. We remark that the particular choice of the parameters for local segmentation and optimization influences only marginally the results of the second layer. The values therefore remain fixed in our experiments.

7.1.6 Experiments

For an experimental evaluation of the proposed multi-layer framework, we use the `HumanEva-II` dataset [SB06] that contains two sequences that were captured by 4 calibrated cameras with resolution of 656×490 pixels and 60 fps. The ground truth has been obtained by a marker-based motion capture system that was synchronized with the cameras. The sequences show two different subjects S2 and S4 performing the motions walking, jogging, and balancing. We use the 3D surface mesh model that is available for subject S4 and does not contain the clothing. Both sequences S2 and S4 are tracked with this model although the mesh model does not fit subject S2 as shown in Figure 7.14. Furthermore, we reduced the number of triangles to 5000 and added a skeleton with 28 degrees of freedom to the mesh. Since not all 20 points of the 3D pose from the marker-based system relate to joints of our mesh, we have used the first frame of each sequence to register the 3D markers of the ground-truth to our mesh. In Figure 7.8, the registered markers are shown by red dots and the joint locations by blue dots. For computing the 2D and 3D error, we take the joint locations of the model, if they are available. Otherwise we use the registered markers. Since the joint locations of subject S4 do not fit subject S2, we have used only the registered markers for S2. In order to register the 3D markers as accurately as possible to the model, we have manually segmented the first frame and estimated the initial pose as described in Section 7.1.3. We remark that not only the tracking and initialization contribute to the overall error, but also the registration and the marker-based system introduce some errors. Hence, the reported errors should be regarded only as upper bounds that allow comparison of different approaches. The experiments are split into two sections. While the first section compares filtering approaches to optimization approaches and complements the evaluation in Section 6.3.3, the second section demonstrates the performance of the proposed multi-layer framework.



(a) (b) PF (c) ICP (d) APF (e) ISA

Figure 7.8: **From left to right: a)** Absolute 3D errors for frames $2 - 821$ of sequence S4. While the estimates of the particle filter (PF) are imprecise, local optimization (ICP) gets stuck in local minima. The annealed particle filter (APF) contains two severe errors ($> 100mm$) around frames 440 and 590 yielding a large standard deviation, see Table 7.1. Global stochastic optimization (ISA) performs very well for the entire sequence. **b-e)** Estimates for frame 580 by PF, ICP, APF, and ISA. ICP fails to track the right arm and the legs are disarranged by the APF.

Optimization vs. Filtering

We have compared interacting simulated annealing (ISA) to local optimization (ICP), a standard particle filter (PF) [DFG01], a variant of the smart particle filter (PFICP) [BKMG07], and the annealed particle filter (APF) [DR05]. The comparison is performed on the first 820 frames of sequence S4 using the absolute 3D error as measurement [SB06]. Since the ground truth is corrupted for the frames $298 - 335$, these frames are neglected in the error analysis. For local optimization, we apply the iterative closest point approach described in Section 7.1.5 to the silhouettes obtained by background subtraction, where the prior on physical constraints (6.31) is integrated according to (5.28). ISA, PF, and APF use the same energy model defined in Section 7.1.3. For the particle filter, we employ the weighting function (6.23) with $\beta_t = 1$. This is similar to the assumption that the likelihood is proportional to a product of normal densities. The particles are predicted as described in Section 6.3.2 without using the mutation operator since it is not supported by a filtering framework, i.e., 50% of the particles are shifted according to the predicted mean and the remaining 50% are directly selected. While ISA and APF are executed with 250 particles and 15 iterations, which are called layers for APF, we set the number of particles to 3750 for the particle filter to obtain the same computational cost. Though the smart particle filter as proposed in [BKMG07] uses stochastic meta descent (SMD) [Sch99] for local optimization, any local optimization like ICP can be used in principle. Since our ICP implementation is slower than SMD, we use 16 particles for PFICP to achieve the same computation time as PF according to [BKMG07]. Since neither PF, APF, PFICP, nor ICP are suitable for initialization, the initial pose is provided by ISA.

The errors are plotted in Figures 7.8 a) and 7.9 a). It shows that the global stochastic optimization approach clearly outperforms the local optimization and the particle filter. While ICP gets stuck in local minima, the estimates of PF are imprecise. The annealed particle filter performs better than the standard particle filter but it still produces two severe errors. This is reflected in the standard deviation for APF given in Table 7.1, which is large in comparison to ISA that performs very well for the entire sequence. The result that APF performs better than PF seems to contradict the comparison in [BSB05] where only slightly better results were obtained by APF. The outcome of APF, however, depends strongly on the parameter for adaptive diffusion [DR05] which was not implemented in the previous comparison. The errors for two different settings, namely 0.4 (APF*) and 0.2 (APF), are plotted in Figure 7.9 a). PFICP does not necessary improve ICP where the best result has been achieved with a very large window size for estimating the correction factor. Approaches like PFICP are in general relatively inefficient since the additional optimization step limits the number of particles such that a good approximation of a distribution is infeasible. Furthermore, a lot of computation time is wasted when the particles migrate to the same local minimum.

error (mm)	PF	ICP	PFICP	APF	ISA
avg	104.61	63.86	69.70	44.15	38.58
std dev	40.77	27.07	24.75	15.39	6.54

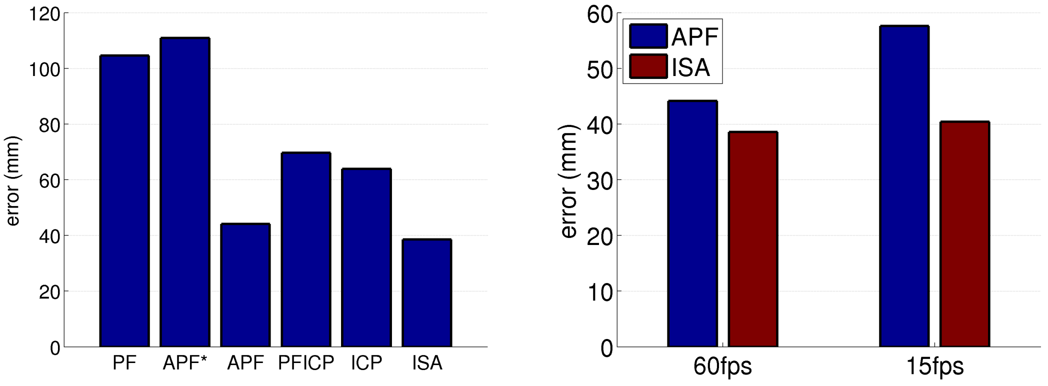
Table 7.1: Averages and standard deviations of the absolute tracking error for frames $2 - 821$ of sequence S4. ISA shows clearly the best results where the standard deviation is significantly lower than for APF.

The performance of APF and ISA on a very fast sequence has been evaluated by reducing the framerate from 60 fps to 15 fps. For the comparison shown in Figure 7.9 b), the parameters for both algorithms are unchanged. While ISA performs very well for 60 Hz and 15 Hz, the error for APF increases by more than 30% when the speed is quadrupled. It might be that the result of APF can be improved by optimizing the parameter for adaptive diffusion on 15 Hz but it is clear that the faster the motion is the more important global optimization becomes.

Although the optimal numbers of particles and iterations for ISA are trade-offs between accuracy and computation cost, Figure 7.10 shows that large numbers of iterations and particles improve the estimates only marginally. Indeed, the error drops until 200 particles and 15 iterations, however after 30 iterations the absolute error is still $36.75mm$. For comparison, an error of $38.58mm$ is obtained by 15 iterations. This indicates that ISA provides estimates near the global optimum in reasonable time, but when more precise estimates are required the ratio between accuracy and computation cost is unsatisfactory.

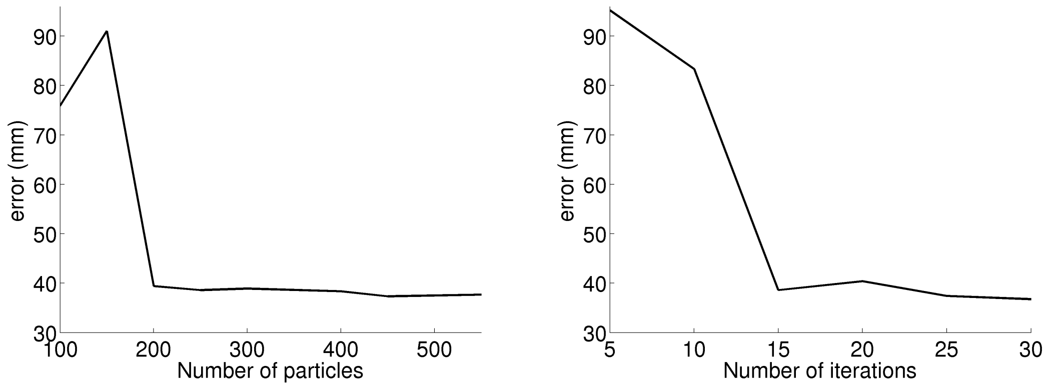
Multi-layer

For evaluating the performance of the proposed multi-layer framework, the absolute 3D tracking errors are measured for the entire sequence S4 that consists of 1257 frames. Figure 7.11 shows that the second layer increases the accuracy of the estimates from the first layer, where 250 particles and 15 iterations are used for ISA and the second layer refines the estimates with a delay of 5 frames. In particular, the largest error around frame 380 is significantly reduced by the second layer. This is reflected by the results given in Table 7.2, where the average error is reduced by 15.9% and the standard deviation by 22.4%. The second layer clearly provides more



(a) (b)

Figure 7.9: Comparison between filtering and optimization approaches. **From left to right: a)** Global stochastic optimization (ISA) provides the best estimates whereas the standard particle filter (PF) and local optimization (ICP) perform poorly. The annealed particle filter (APF) performs better than a combination of particle filtering with local optimization (PFICP) provided that the parameter for adaptive diffusion is well chosen. Otherwise, the error for APF becomes very large. The detailed errors with standard deviations are listed in Table 7.1. **b)** The effect of a very fast movement is simulated by using only every 4th frame of sequence S4 (frames $2 - 821$). This corresponds to a walking and running sequence recorded with 15 fps. While the error increases slightly by 4.68% to $40.39mm$ for ISA, the error for APF rises to $57.63mm$ by 30.5%.



(a) (b)

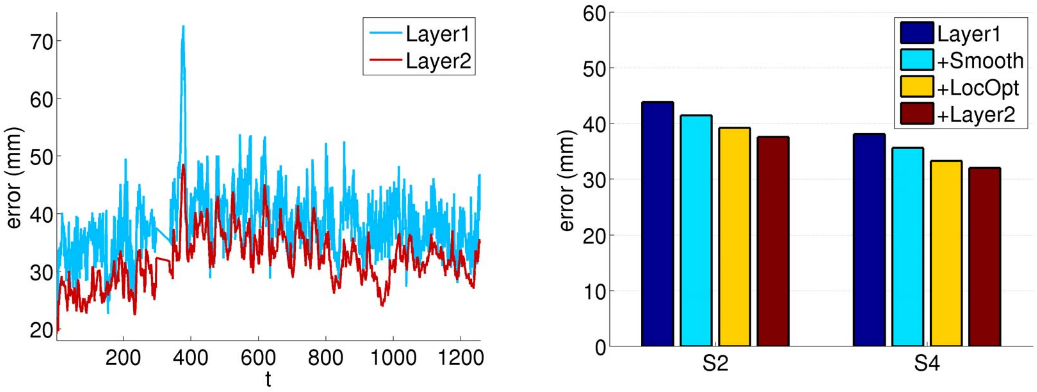
Figure 7.10: Absolute tracking error of global optimization for frames $2 - 821$ of sequence S4. Large numbers of iterations and particles improve the estimates only marginally. **From left to right: a)** Error with respect to the number of particles using 15 iterations. **b)** Error with respect to the number of iterations using 250 particles.

precise estimates, which cannot be achieved by an increased number of particles and iterations in reasonable time; see Figure 7.10. Our current implementation requires 76 seconds per frame for the first layer and 48 seconds per frame for the second layer on a standard computer whereas ISA with 30 iterations would require 152 seconds per frame.

The errors and quantiles for individual joints are provided in Figure 7.12. The quantiles show that most joints, particularly the knees, are very well estimated. It also reveals that the limb extremities, namely wrists and ankles, are more difficult to track since hands and feet are relatively small body parts. The lower quantiles indicate the registration errors of the joint positions, particularly of the ankles. Since the distances between the upper and lower quantiles for the wrists and ankles are similar, the larger error of the ankles might be explained by the registration error.

error (mm)	Layer1	L1+Smooth	L1+LocOpt	L1+Layer2
avg	38.07	35.58	33.23	32.01
std dev	5.84	5.09	5.08	4.53

Table 7.2: Averages and standard deviations of the absolute tracking error for the complete sequence S4 (frames $2 - 1258$). The error of the first layer using only global optimization is significantly reduced by the second layer. Clearly, a coupling of smoothing and local optimization provides more precise results than each of them alone.



(a) (b)

Figure 7.11: **From left to right: a)** Absolute tracking error for the sequence S4 (frames $2 - 1258$). The second layer reduces jitter and increases the accuracy of the estimates from the first layer. In particular, the largest error around frame 380 is significantly reduced by the second layer. **b)** A comparison of the average errors for the complete sequences S2 and S4 shows the improvements of our multi-layer framework. The detailed errors with standard deviations are given in Tables 7.2 and 7.3.

We have also evaluated the impact of coupling local optimization and smoothing for the second layer, which performs better than each of these steps alone. This is shown in Figure 7.11. Tables 7.2 and 7.3 reveal that the accuracy is primarily increased by local optimization whereas smoothing reduces the jitter, as indicated by the decreased standard deviation. The best results for the second layer were achieved with a short delay of 5 frames as plotted in Figure 7.6 b).

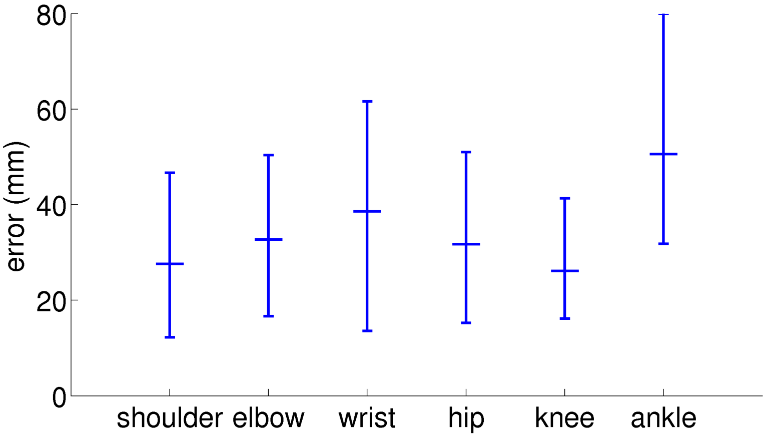


Figure 7.12: Average errors and 0.025-quantiles for individual joints obtained by the multi-layer framework on the entire sequence S4. While the knees are very well estimated, the error bars for the limb extremities such as wrists and ankles are larger than for other joints. The quantiles of the ankles indicate that the ankle joints are not well registered.

Even without delay, the error is slightly reduced compared to applying only local optimization. The computation times are listed in Table 7.4. For convenience, we also provide the error of the second layer in Table 7.5 when a particle filter approach is used as first layer.

error (mm)	Layer1	L1+Smooth	L1+LocOpt	L1+Layer2
avg	43.82	41.44	39.20	37.53
std dev	10.65	9.67	10.05	9.00

Table 7.3: Averages and standard deviations of the absolute tracking error for the complete sequence S2 (frames $1 - 1202$).

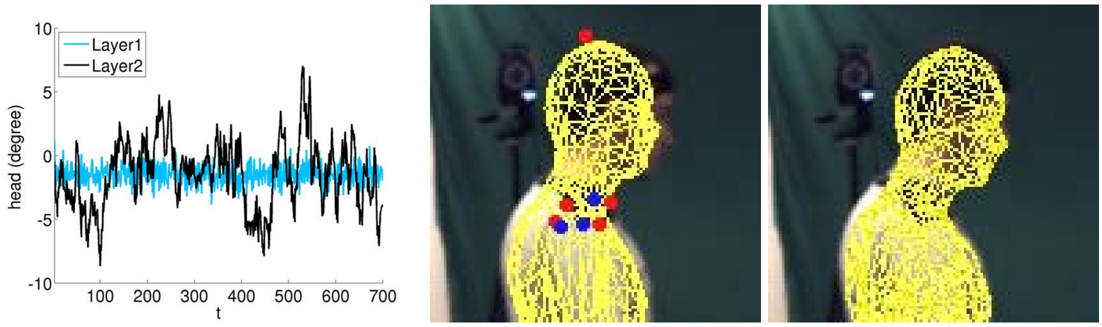
	Layer1	L1+Smooth	L1+LocOpt	L1+Layer2
sec/frame	76	76	124	124

Table 7.4: Overall computation time on a standard PC for a frame with 4 images.

We further applied the multi-layer framework to sequence S2 that consists of 1202 frames. Since we use the 3D surface mesh model of subject S4, the model does not fit subject S2, see Figure 7.14. Nevertheless, competitive results are obtained even though the error is larger by $6mm$ than for sequence S4, see Tables 7.2 and 7.3. The increase of the error seems to be mainly caused by the wrong model since the camera setting and movement are very similar to S4. Particularly, the elbow joints of the model are at the wrong position which causes problems when the elbows are angled. This indicates that our approach would also work with a generic surface model like the SCAPE model [ASK$^+$05, BSB$^+$07]. However, it also reveals that the quality of the surface mesh has a significant impact on the accuracy of the estimates.

error (mm)	PF + L2	PFICP + L2	APF + L2	ISA + L2
avg	82.70	58.38	37.26	32.49
std dev	43.77	25.32	14.67	5.21

Table 7.5: Averages and standard deviations of the absolute tracking error for frames $2 - 821$ of sequence S4. The second layer (L2) improves the results for all sampling approaches. The results without the second layer are given in Table 7.1.



(a) (b) Layer 1 (c) Layer 2

Figure 7.13: Biased estimates. **From left to right: a)** When the physical constraints are modeled by a strong prior, the estimates are biased towards the training data. For this example, only joint samples around zero have been used. Since the second layer does not make use of the prior, the bias is reduced. **b)** Biased estimate of the head by the first layer. **c)** The estimate of the second layer better fits the image data.

The influence of a strong prior is demonstrated in Figure 7.13. To this end, we learned the physical constraints of the head movement only by joint samples around zero. While the estimates from the first layer are biased towards the training data and do not fit the image data, the second layer reduces the bias since it does not rely on the prior. We emphasize that the bias is not completely removed, since the second layer is initialized by the estimates of the first layer, but the example shows that the estimates of our multi-layer framework better fit the image data.
In order to allow comparison to other approaches that have not been mentioned in this section, we provide various error metrics for the sequences S2 and S4 in Tables 7.6 and 7.7. Each sequence is split into three sets, where the first set contains only the walking motion, the second the walking and jogging motion, and the third set the entire sequence consisting of walking, jogging, and balancing. The average errors and standard deviations are given for global stochastic optimization (one layer) and the multi-layer framework (two layers). The 2D errors are computed for cameras C1 and C2. The relative error is computed with respect to the pelvis joint. For a detailed description on the error metrics, we refer to [SB06]. We remark that the relative error is higher than the absolute error. This indicates that the marker for the pelvis joint has not been accurately registered to the surface mesh model. In addition, some estimated human body poses of the multi-layer framework are shown in Figures 7.14 and 7.15.

♯Layers	Dataset	3D (mm)		2D/C1 (pix)		2D/C2 (pix)	
		absolute	relative	absolute	relative	absolute	relative
1	Set1 $(1-350)$	$\mathbf{41.50 \pm 7.98}$	45.78 ± 9.00	5.45 ± 1.49	5.85 ± 1.74	5.54 ± 1.78	5.66 ± 1.84
2	Set1 $(1-350)$	$\mathbf{32.23 \pm 5.71}$	33.49 ± 6.03	4.10 ± 1.11	4.24 ± 1.25	4.38 ± 1.36	4.28 ± 1.33
1	Set2 $(1-700)$	$\mathbf{45.04 \pm 12.85}$	48.36 ± 13.68	5.79 ± 1.89	6.04 ± 2.04	6.07 ± 2.35	6.22 ± 2.41
2	Set2 $(1-700)$	$\mathbf{35.86 \pm 10.73}$	37.62 ± 11.42	4.49 ± 1.44	4.65 ± 1.55	4.85 ± 1.86	4.92 ± 2.01
1	Set3 $(1-1202)$	$\mathbf{43.82 \pm 10.65}$	46.57 ± 11.44	5.61 ± 1.57	5.89 ± 1.71	5.95 ± 1.91	6.14 ± 1.96
2	Set3 $(1-1202)$	$\mathbf{37.53 \pm 9.00}$	39.36 ± 9.70	4.77 ± 1.25	4.99 ± 1.34	5.13 ± 1.55	5.25 ± 1.69

Table 7.6: 3D and 2D errors for subject S2. Accurate results are obtained by our multi-layer framework although the sequence has been tracked with a wrong surface mesh model, see Figure 7.14.

♯Layers	Dataset (Frames)	3D (mm)		2D/C1 (pix)		2D/C2 (pix)	
		absolute	relative	absolute	relative	absolute	relative
1	Set1 $(2-350)$	$\mathbf{34.59 \pm 4.63}$	43.93 ± 8.24	4.48 ± 1.00	5.66 ± 1.69	4.17 ± 0.72	4.93 ± 1.17
2	Set1 $(2-350)$	$\mathbf{27.65 \pm 2.96}$	33.91 ± 4.97	3.58 ± 0.74	4.40 ± 1.03	3.35 ± 0.51	3.91 ± 0.86
1	Set2 $(2-700)$	$\mathbf{38.53 \pm 6.90}$	47.00 ± 10.60	5.14 ± 1.30	6.22 ± 1.90	5.01 ± 1.38	5.70 ± 1.76
2	Set2 $(2-700)$	$\mathbf{32.14 \pm 5.42}$	37.31 ± 6.55	4.34 ± 1.05	5.04 ± 1.21	4.24 ± 1.14	4.72 ± 1.35
1	Set3 $(2-1258)$	$\mathbf{38.07 \pm 5.84}$	45.25 ± 9.13	5.25 ± 1.17	6.12 ± 1.62	5.00 ± 1.12	5.71 ± 1.53
2	Set3 $(2-1258)$	$\mathbf{32.01 \pm 4.53}$	36.01 ± 5.79	4.42 ± 0.92	4.99 ± 1.04	4.30 ± 0.93	4.71 ± 1.10

Table 7.7: 3D and 2D errors for subject S4. The frames $298-335$ are neglected since the ground truth is corrupted for these frames.

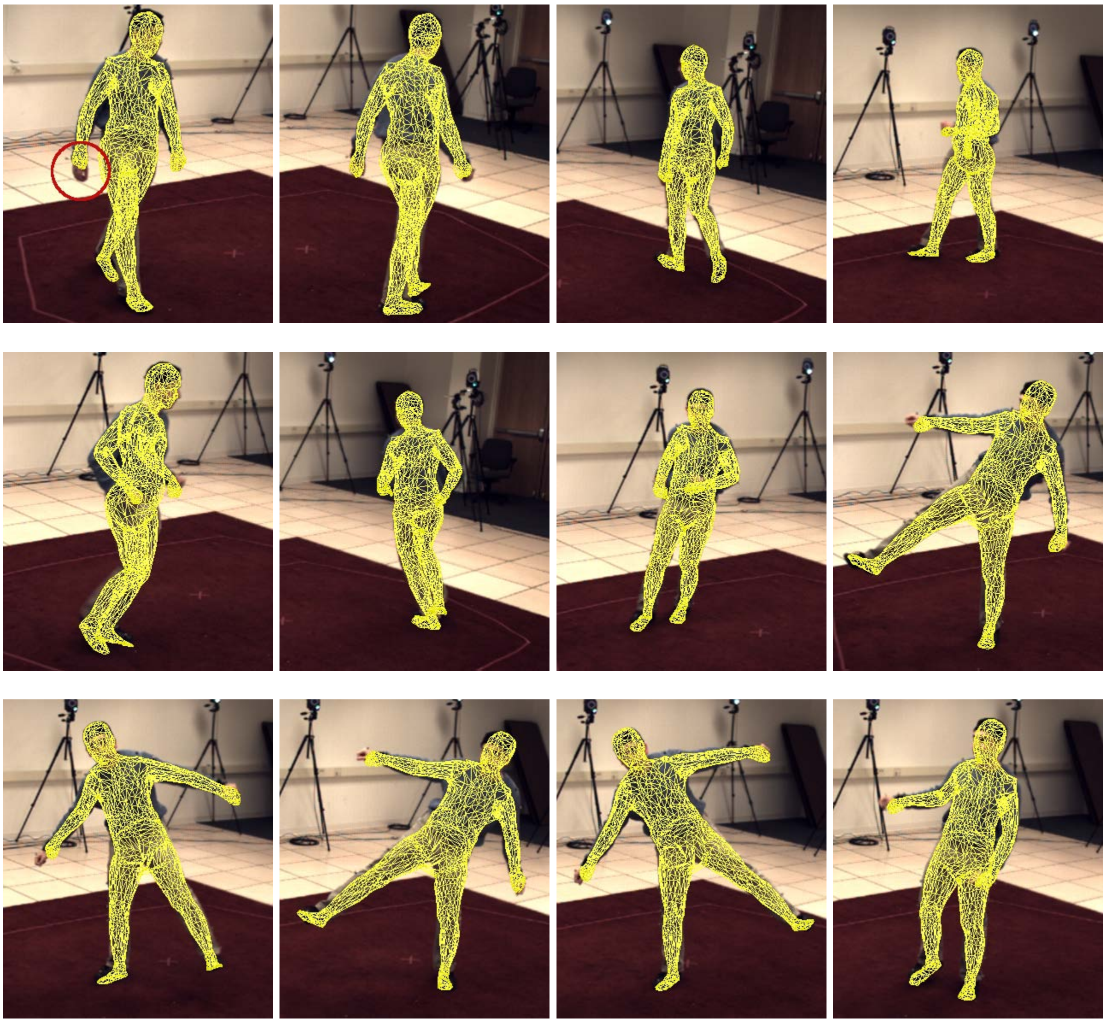


Figure 7.14: Estimates for subject S2. Note that the arms of the surface mesh model are too short, since the model of subject S4 has been used for tracking (*top left*). **From top left to bottom right:** The meshes of the estimates are projected on the images of camera C1 for frames 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, and 1200.

7.1.7 Summary

In this section, we have compared optimization and filtering approaches for model-based human motion capture that do not rely on prior knowledge on the dynamics. A quantitative error analysis has revealed that interacting simulated annealing provides significantly better estimates than an iterative closest point approach, a standard particle filter, a variant of the smart particle filter, or the annealed particle filter. While ISA provides robust and relatively accurate estimates of the human pose, an even higher precision is only achieved at the expense of high computational cost. To address this problem, we have introduced a multi-layer framework that combines the advantages of global stochastic optimization, local optimization, and filtering. While the first layer relies on ISA, the second layer refines the estimates where filtering and local optimization

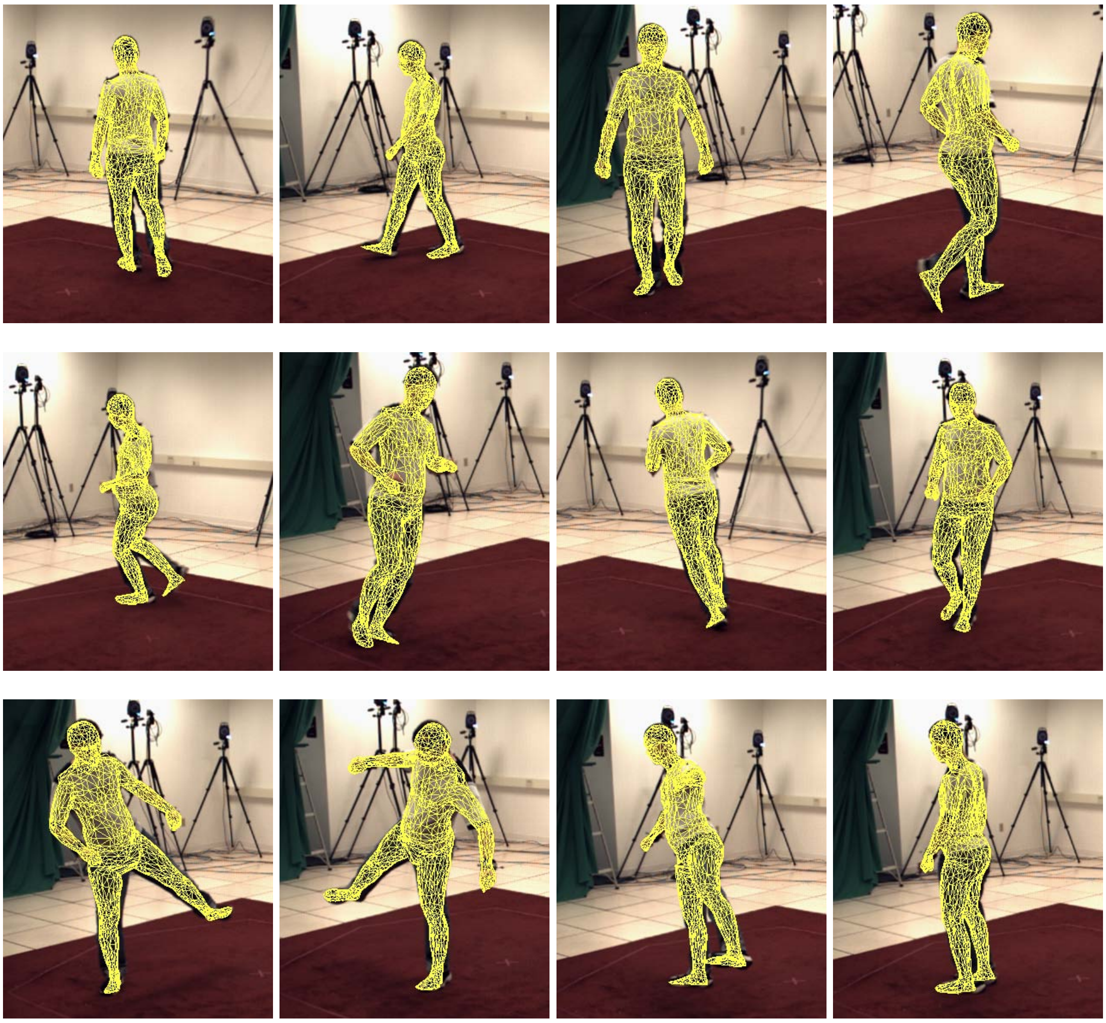


Figure 7.15: Estimates for subject S4. **From top left to bottom right:** The meshes of the estimates are projected on the images of camera C2 for frames 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, and 1200.

are coupled. The second layer not only increases the accuracy, but also reduces jitter and potential bias from the first layer. The latter is an important issue particularly in medical applications. In practice, the two layers can be run in parallel such that the processing time is not increased. An additional speed-up by a factor of $12 - 20$ has been observed for the first layer when the evaluation of the energy is performed on a GPU.

Since the described approach is based on a fixed surface model, its general applicability has some limitations. Although good results are obtained even with a wrong surface model, we have demonstrated that the quality of the surface mesh has an impact on the accuracy of the estimates. A solution to be investigated is to adapt a generic human model to the image data. The framework could also be combined with motion priors, which might be useful in monocular scenarios. The multi-layer framework is appealing in this case, since the motion priors would reduce the search space for ISA and the second layer would be necessary to reduce the bias introduced by the priors.

7.2 Skeleton Tracking and Surface Estimation

The human motion capture approaches discussed so far use a skeleton-based deformation to approximate the surface deformations. Ideally, one expects to both estimate an articulated rigid-body skeleton that explains the overall motion of the character, as well as the potentially non-rigid deformation of the surface, e.g. caused by tissue or garment. On the one end of the spectrum, many current automatic approaches track only a skeleton model which poses strong restrictions on the subject, like tight clothing. Since garment motion, for instance, is non-rigid and rarely aligned with the motion of the underlying articulated body, these algorithms often dramatically fail if the subject wears wide clothing like a dress. On the other end of the spectrum, there are methods which capture a faithfully deforming 3D surface of the subject, but do not provide an underlying skeleton.

In contrast, our approach captures both skeletal motion as well as an accurately deforming surface of an animal or human by fitting a body model to multi-view image data. Our body model is a combination of a bone skeleton with joints, as well as a surface whose deformation is only loosely coupled with the skeleton motion. We can accurately capture both detailed surface deformations and motion of wide apparel, which are essential for realistic character animations. At the same time, the skeleton provides a low-dimensional motion parameterization which facilitates tracking of fast movements. Our captured performances can be easily edited and used in animation frameworks typical for games and movies, which are almost exclusively skeleton-based.

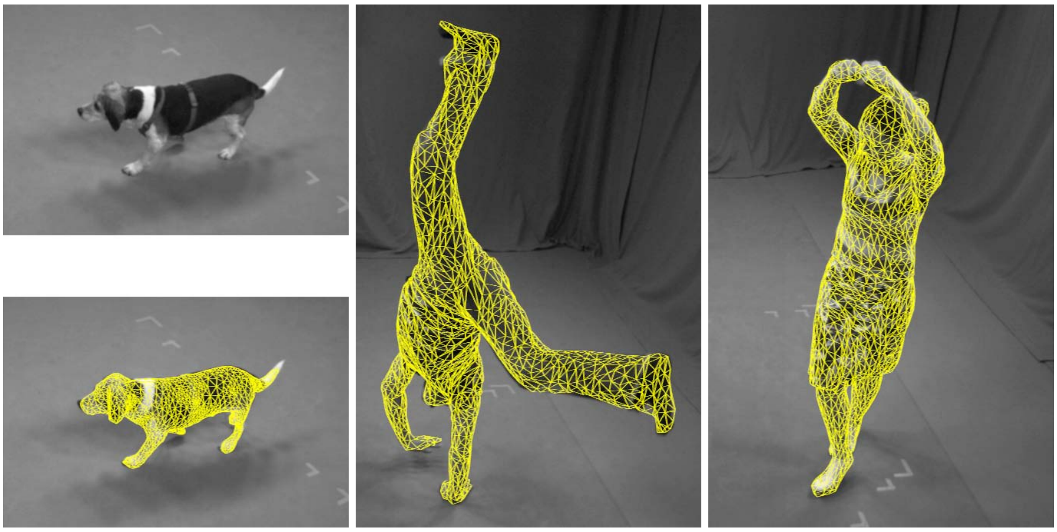


Figure 7.16: Our approach captures the motion of animals and humans accurately even in the case of rapid movements and wide apparel. The images show three examples of estimated surfaces that are superimposed on the images.

Finally, our approach exceeds the performance of related methods from the literature since both accurate skeleton and surface motion are found fully-automatically. This is achieved by the following properties:

- Our approach recovers the movement of the skeleton and the temporal deformation of the 3D surface in an interleaved manner. To find the body pose in the current frame, we first optimize the skeletal pose and use simple approximate skinning to deform the detailed surface of the previous time step into the current time step. Once converged, the fine surface deformation at the current time step is computed without limiting the deformation to comply with the skeleton. This improves also the skeleton estimation and avoids errors caused by wide apparel since the refined surface model of the previous frame provides a good approximation of the surface at the current frame.

- Since skeleton-based pose estimation is more constrained than surface estimation, our approach is less sensitive to silhouette noise than comparable visual hull approaches and runs even on medium quality multi-view sequences like the HumanEva benchmark [SB06]. The reliability and accuracy of our approach is demonstrated on 12 sequences that consist of over 5000 frames with 9 different subjects (including a dog) performing a wide range of motions and wearing a variety of clothing.
- Since local optimization methods get stuck in local minima and cannot recover from errors, they cannot track challenging sequences without manual interaction. In order to overcome the limitations of local optimization, we exploit the tree structure of the skeleton to split the optimization problem into a local and a lower dimensional global optimization problem.

The optimization scheme is motivated by the observation that local optimization is efficient and accurately tracks most frames of a sequence. However, it fails completely in some frames where the motion is fast or the silhouettes are noisy. The error often starts at a certain limb or branch of the skeleton and is propagated through the kinematic chain over time until the target is irrevocably lost. Our approach interferes before the error spreads. It detects misaligned limbs after local optimization and re-estimates the affected branch by global optimization. Since global optimization is only performed for few frames and for a lower dimensional search space, the approach is suitable for large data sets and high dimensional skeleton models with over 35 degrees of freedom.

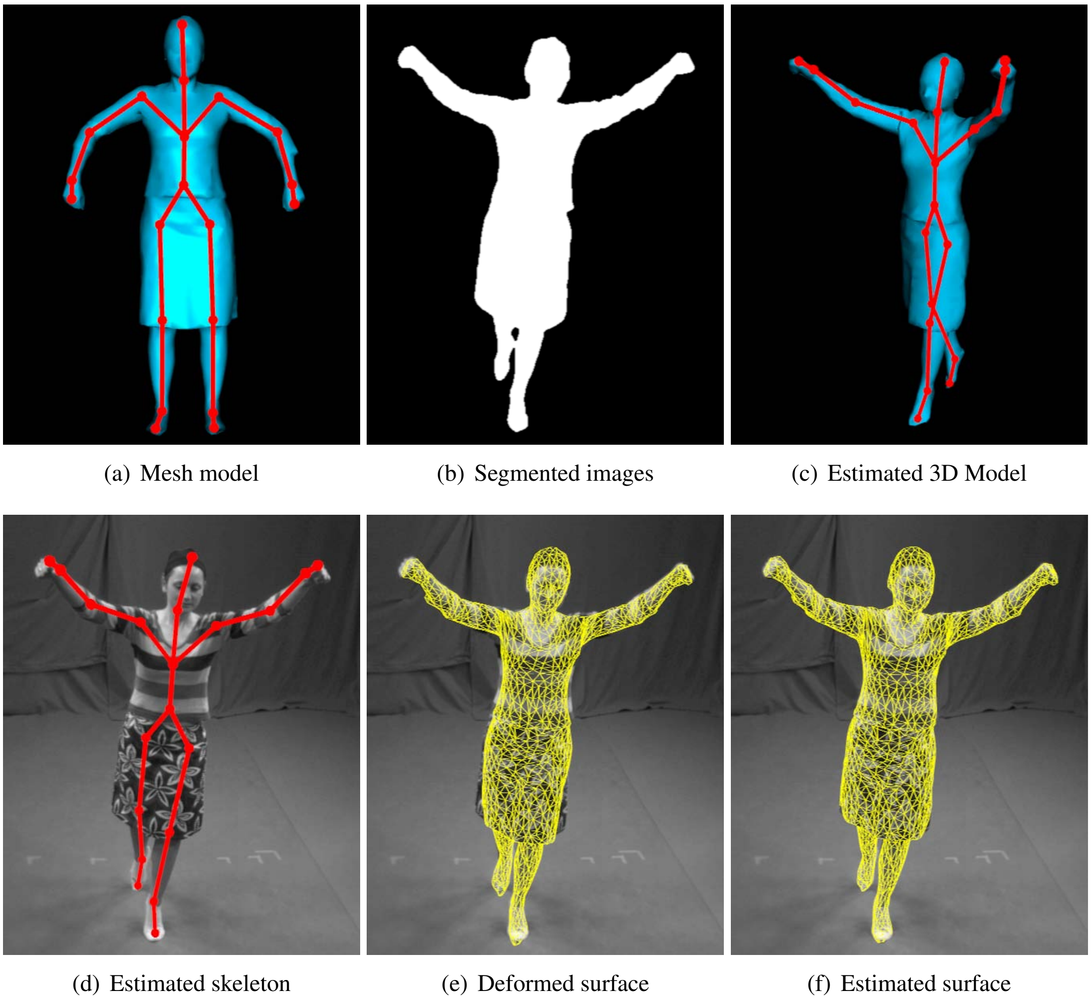
7.2.1 Beyond Articulated Models

Since articulated models are not very realistic models of the human body, implicit surfaces based on metaballs [PF03], shape-from-silhouette model acquisition [CBK05], or the learned SCAPE body model [ASK$^+$05, BSB$^+$07] have been proposed. While these approaches model the human body without clothing, Balan and Black [BB08] have used SCAPE to estimate the human body underneath clothes from a set of images. Tracking humans wearing more general apparel has been addressed in [RKP$^+$07] where a physical model of the cloth is assumed to be known.

In contrast to skeleton-based approaches, 3D surface estimation methods are able to capture time-varying geometry in detail. Many approaches like [SH03, VZBH08] rely on the visual hull but suffer from topology changes that occur frequently in shape-from-silhouette reconstructions. Mesh-based tracking approaches as proposed in [ATSS07] and [AST$^+$08] provide frame-to-frame correspondences with a consistent topology. Fitting a mesh model to silhouettes and stereo, however, requires a large amount of correspondences to optimize the high dimensional parameter space of a 3D mesh. This, in turn, makes them more demanding on processing time and image quality than skeleton-based methods.

Our approach is most similar to the work of Vlasic et al. [VBMP08] where a two-pass approach has been proposed. In the first pass, a skeleton is geometrically fit into the visual hull for each frame. The second pass deforms a template model according to the estimated skeleton and refines the template to fit the silhouettes. Despite of visual appealing results, a considerable amount of manual interaction is required, namely up to every 20th frame, to correct the errors of the skeleton estimation. The errors are caused by fitting the skeleton to the visual hull via local optimization without taking a complete surface model or texture information into account. Moreover, their visual hull approach is sensitive to silhouette errors. In contrast, our local-global optimization makes for a fully-automatic approach that also works on data of poor image quality.

7.2.2 Overview



(a) Mesh model

(b) Segmented images

(c) Estimated 3D Model

(d) Estimated skeleton

(e) Deformed surface

(f) Estimated surface

Figure 7.17: Having an articulated template model **(a)** and silhouettes **(b)** from several views, our methods tracks the skeleton and estimates the time-varying surface consistently without supervision by the user **(c)**. Using the estimated surface of the previous frame, the pose of the skeleton **(d)** is optimized such that the deformed surface **(e)** fits the image data. Since skeleton-based pose estimation is not able to capture garment motion **(e)**, the surface is refined to fit the silhouettes **(f)**.

The performance of an animal or human is captured by synchronized and calibrated cameras and the silhouettes are typically extracted by background subtraction or chroma-keying. Our body model comprises of two components, a 3D triangle mesh surface model $\mathcal{M}$ with 3D vertex locations X_i and an underlying bone skeleton as shown in Figure 7.17 a). We assume that a 3D surface model $\mathcal{M}$ of the tracked subject in a static pose is available. It might be acquired by a static full-body laser scan, by shape-from-silhouette methods, or from a human shape database as SCAPE. In our experiments, we demonstrate results for all three cases, but would like to note that model acquisition is outside of the scope of this work as discussed in Section 1.6. A kinematic skeleton is then inserted into the 3D mesh. In our case, an object-specific skeleton with usually around 36 degrees-of-freedom is generated by manually marking the joint positions. Thereafter, weights $w_{k,i}$ are automatically computed for each X_i, which describe the association of X_i with each bone k [BP07]. The weights allow us to do skinning, i.e. a simple approximation of non-linear surface deformation based on the skeleton pose. Weighted skinning is used to interpolate the joint transformations on a per-vertex-basis. We use quaternion blend skinning [KCvO07] which produces less artifacts than linear blend skinning methods.

An outline of the processing pipeline is given in Figure 7.17. Starting with the estimated mesh and skeleton from the previous frame, the skeleton pose is optimized as described in Section 7.2.3 such that the projection of the deformed surface fits the image data in a globally optimal way (Figure 7.17 d)). Since this step only captures deformations that can be approximated by articulated surface skinning (Figure 7.17 e)), subsequently the non-rigid surface is refined as described in Section 7.2.4 (Figure 7.17 f)). The estimated refined surface and skeleton pose serve as initialization for the next frame to be tracked.

7.2.3 Skeleton-based Pose Estimation

Since local pose optimization is prone to errors and global pose optimization in high dimensional spaces is expensive, our method estimates poses in two phases. The first phase searches for the nearest local minimum of an energy functional that assesses the model-to-image alignment based on silhouettes and texture features. To this end, the whole articulated skeleton is optimized locally. Subsequently, misaligned bones are detected by evaluating the energy E_k of each rigid body part. When the energy exceeds a given threshold, the affected limb is labeled as misaligned. In addition, the preceding limb in the kinematic chain is also labeled when the joint between the limbs has less than three degrees of freedom (e.g. knee or elbow). For instance, a wrong estimate of the shank might be caused by a rotation error along the axis of the thigh. Then the labeling process is continued such that all bones until the end of the branch are labeled as illustrated in Figure 7.18. Thereafter, the labeled bones are re-estimated by global optimization.

Local Optimization

The articulated pose is represented by a set of twists $\theta_j\hat{\xi}_j$ as in Section 5.1. A transformation of a vertex X_i, which is associated with bone k_i and influenced by n_{k_i} out of totally N joints, is given by

$$T_\chi X_i = \exp\left(\theta\hat{\xi}\right)\exp\left(\theta_{\iota_{k_i}(1)}\hat{\xi}_{\iota_{k_i}(1)}\right)\dots\exp\left(\theta_{\iota_{k_i}(n_{k_i})}\hat{\xi}_{\iota_{k_i}(n_{k_i})}\right)X_i, \tag{7.16}$$

where the mapping ι_{k_i} represents the order of the joints in the kinematic chain. Since the joint motion depends only on the joint angle θ_j, the state of a kinematic chain is defined by a parameter vector $\chi := (\theta\xi, \Theta) \in \mathbb{R}^d$ that consists of the six parameters for the global twist $\theta\hat{\xi}$ and the joint angles $\Theta := (\theta_1, \dots, \theta_N)$.

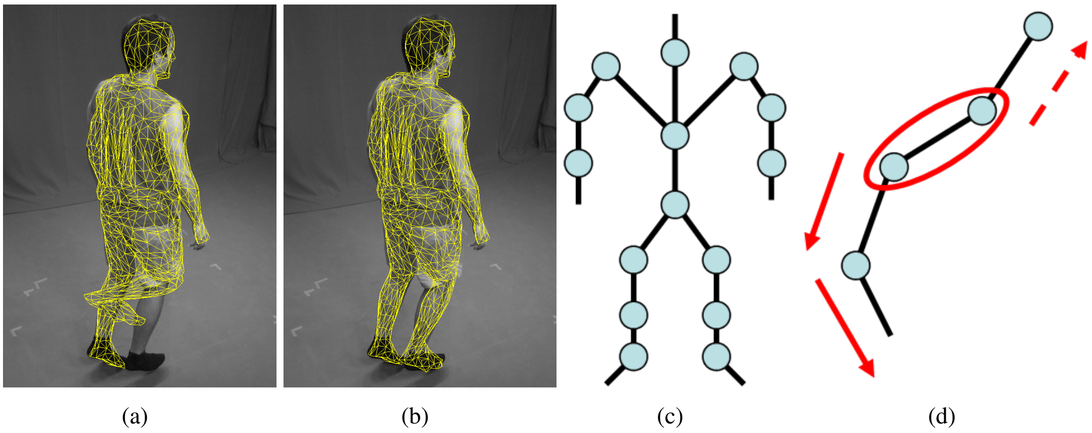


Figure 7.18: Although local optimization is prone to errors, often only a single branch of the kinematic chain is affected **(a)**. This reduces the computational burden for global optimization since it can be performed in a lower dimensional subspace to correct the estimation error **(b)**. After detecting misaligned limbs *(red circle)*, the kinematic chain is traversed *(red arrows)* to label bones and associated joints that have to be globally optimized **(c,d)**.

For estimating the parameters χ, a sufficient set of point correspondences between the 3D model X_i and the current frame x_i is needed. For the local optimization, we rely on silhouette contours and texture as in Section 5.5. Contour correspondences are established between the projected surface and the image silhouette by searching for closest points between the respective contours. Texture correspondences between two frames are obtained by matching SIFT features [Low04]. In both cases, the 2D correspondences are associated with a projected model vertex X_i yielding the 3D-2D correspondences (X_i, x_i). In the contour case, x_i is the point on the image contour closest to the projected vertex location in the current frame. In the texture case, x_i is the 2D location in the current frame that is associated with the same SIFT feature as the projected vertex X_i in the previous frame. Since each 2D point x_i defines a projection ray that can be represented as Plücker line $L_i = (n_i, m_i)$ [Sto91], the error of a pair $(T_\chi X_i, x_i)$ is given by the norm of the perpendicular vector between the line L_i and the transformed point $T_\chi X_i$:

$$\left\| \Pi \left(T_\chi X_i \right) \times n_i - m_i \right\|_2 , \tag{7.17}$$

where Π denotes the projection from homogeneous coordinates to non-homogeneous coordinates. Using Equations (7.16) and (7.17), one obtains the weighted least squares problem

$$\underset{\chi}{\operatorname{argmin}} \frac{1}{2} \sum_i w_i \left\| \Pi \left(T_\chi V_i \right) \times n_i - m_i \right\|_2^2 \tag{7.18}$$

that can be solved iteratively and linearized by using the Taylor approximation $\exp(\theta\hat{\xi}) \approx I + \theta\hat{\xi}$, where I denotes the identity matrix, as in Section 5.5.2. In order to stabilize the optimization, the linear system is regularized by

$$\beta\theta_j = \beta \left(\hat{\theta}_j - \tilde{\theta}_j \right) , \tag{7.19}$$

where $\hat{\theta}_j$ is the predicted angle from a linear 3rd order autoregression (5.22), $\tilde{\theta}_j$ is the previously estimated absolute joint angle, and β is a small constant. The pose $\hat{\chi}$ represented by all

$\hat{\theta}_j$ can be regarded as a conservative prediction for the current frame. Since the optimization regards the limbs as rigid structures, the mesh is updated between the iterations by quaternion blending [KCvO07] to approximate smooth surface deformation.
While contour correspondences are all weighted equally with $w_i^C = 1$, the texture correspondences have higher weights w_i^T during the first iteration since they can handle large displacements. For the first iteration, we set the weights such that $\sum_i w_i^T = \alpha \sum_i w_i^C$ with $\alpha = 2.0$, i.e. the impact of the texture features is twice as high as the contour correspondences. After the first iteration, the solution already converges to the nearest local minimum such that the texture features can be down-weighted by $\alpha = 0.1$. In addition, obvious outliers are discarded by thresholding the re-projection error of the texture correspondences similar to Section 5.5.1.
After the local optimization has converged to a solution χ, the error for each limb is evaluated individually. Since each correspondence is associated with one limb k, the limb-specific energy is obtained by

$$E_k(\chi) = \frac{1}{K} \sum_{\{i;k_i=k\}} \|\Pi\left(T_\chi X_i\right) \times n_i - m_i\|_2^2 \, , \tag{7.20}$$

where only contour correspondences are used and $K = |\{i; k_i = k\}|$. If at least one limb exceeds the predefined upper bound of the energy function, the second phase of the optimization, global optimization, is initiated.

Global Optimization

After labeling the joints of the misaligned limbs as illustrated in Figure 7.18, the parameter space of the skeleton pose $\mathbb{R}^d$ is projected onto a lower dimensional search space $P(\chi) \to \tilde{\chi} \in \mathbb{R}^m$ with $m \leq d$ by keeping the parameters of the non-labeled joints fixed. In order to find the optimal solution for $\tilde{\chi}$, we minimize the energy

$$\underset{\tilde{\chi}}{\operatorname{argmin}} \left\{ E_S(P^{-1}(\tilde{\chi})) + \gamma \, E_R(\tilde{\chi}) \right\} . \tag{7.21}$$

While the first term measures the silhouette consistency between the projected surface and the image, the second term penalizes strong deviations from the predicted pose and serves as a weak smoothness prior weighted by $\gamma = 0.01$.
The silhouette functional $E_S(P^{-1}(\tilde{\chi}))$ is a modification of the Hamming distance. Using the inverse mapping $\chi = P^{-1}(\tilde{\chi})$ as new pose, the surface model is deformed by quaternion blend skinning and projected onto the image plane for each camera view c. As in Section 6.3.1, the consistency error for a single view is obtained by the pixel-wise differences between the projected surface $S_c^p(\chi)$ in model pose χ and the binary silhouette image S_c:

$$E_S^c(\chi) = \frac{1}{area(S_c^p)} \sum_p |S_c^p(\chi)(p) - S_c(p)| + \frac{1}{area(S_c)} \sum_q |S_c(q) - S_c^p(\chi)(q)| \, , \tag{7.22}$$

where the sums with respect to p and q are only computed over the silhouette areas of $S_c^p(\chi)$ and S_c, respectively. In order to penalize pixel mismatches that are far away from the silhouette, a Chamfer distance transform is previously applied to the silhouette image. The silhouette term E_S is finally the average of E_S^c over all views.
The second term of the energy function (7.21) introduces a smoothness constraint by penalizing deviations from the predicted pose $\hat{\chi}$ in the lower dimensional space:

$$E_R(\tilde{\chi}) = \|\tilde{\chi} - P(\hat{\chi})\|_2^2. \tag{7.23}$$

Since we seek for the globally optimal solution for $\tilde{\chi} \in \mathbb{R}^m$, we use the particle-based global optimization from Chapter 6. The method is appropriate to our optimization scheme since the computational effort can be adapted to the dimensions of the search space and the optimization can be initiated with several hypotheses. In our setting, each particle represents a single vector $\tilde{\chi}$ in the search space that can be mapped to a skeleton pose by the inverse projection P^{-1}. The computational effort depends on two parameters, namely the number of iterations and the number of particles. While the latter needs to be scaled with the search space, the number of iterations can be fixed. In our experiments, we have used 15 iterations and $20 * m$ particles with a maximum of 300 particles. These limits are necessary to have an upper bound for the computation time per frame. Furthermore, the optimization is performed on the whole search space when more than 50% of the joints are affected. It usually happens when the torso rotation is not well estimated by the local optimization which is, however, rarely the case.
The initial set of particles is constructed from two hypotheses, the pose after the local optimization and the predicted pose. To this end, we uniformly interpolate between the two poses and diffuse the particles by a Gaussian kernel.

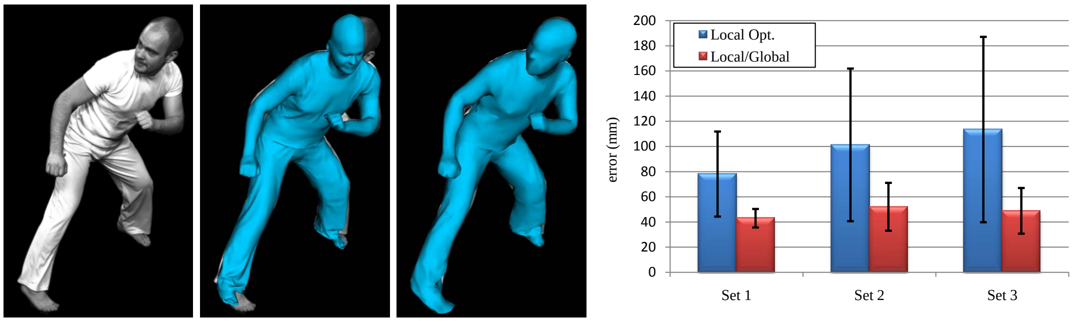
7.2.4 Surface Estimation

Since quaternion blend skinning is based on the overly simplistic assumption that the surface deformation is explained only in terms of an underlying skeleton, the positions of all vertices need to be refined to fit the image data better as illustrated in Figures 7.17 e) and f). To this end, we abandon the coupling of vertices to underlying bones and refine the surface by an algorithm that is related to the techniques used by de Aguiar et al. [AST+08] and Vlasic et al. [VBMP08]. We also use a Laplacian deformation framework (see [BS08] for a comprehensive overview) to move the silhouette rim vertices of our mesh (vertices that should project onto the silhouette contour in one camera) towards the corresponding silhouette contours of our images. In contrast to previous work, we do not formulate deformation constraints in 3D, i.e., we do not require contour vertices on the model $\mathcal{M}$ to move towards specific 3D points found via reprojection. Instead, we constrain the projection of the vertices to lie on 2D positions on the image silhouette boundary. This makes the linear system to be solved for the refined surface more complex, as we have to solve for all three dimensions concurrently rather than sequentially, as is possible in the previous works. But on the other hand this gives the deformation further degrees of freedom to adapt to our constraints in the best way possible. We reconstructed the refined surface by solving the least-squares system

$$\underset{X}{\operatorname{argmin}} \left\{ \|LX - \delta\|_2^2 + \alpha \|C_{sil}X - q_{sil}\|_2^2 \right\}. \tag{7.24}$$

Here, L is the cotangent Laplacian matrix and δ are the differential coordinates of our current mesh with vertices X [BS08]. The second term in our energy function defines the silhouette constraints and their weighting factor α. Matrix C_{sil} and vector q_{sil} are assembled from individual constraints that take the following form: Given the 3×4 projection matrix M^c of a camera c split into its translation vector T^c and the remaining 3×3 transformation N^c, the target screen space coordinates $x_i = (x_{i,u}, x_{i,v})$, and the 3D position X_i of a vertex on the 3D silhouette rim of $\mathcal{M}$, we can express a silhouette alignment constraint using two linear equations:

$$\begin{aligned} (N_1^c - x_{i,u}N_3^c)X_i &= -T_1^c + x_{i,u}T_3^c \\ (N_2^c - x_{i,v}N_3^c)X_i &= -T_2^c + x_{i,v}T_3^c \end{aligned} \tag{7.25}$$



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

Figure 7.19: **Left:** Visual comparison of our approach with [AST+08]. **a)** Input image. **b)** Tracked surface mesh from [AST+08]. **c)** Tracked surface mesh with lower resolution obtained by our method. While [AST+08] handles loose clothes better, our approach estimates the human pose more reliably. **Rightmost: d)** Comparison of our optimization scheme with local optimization. The bars show the average error and standard deviation of the joint positions of the skeleton for the S4 sequence of the `HumanEva-II` benchmark. The three sets cover the frames $2-350$ (walking), $2-700$ (walking+jogging), and $2-1258$ (walking+jogging+balancing). While our approach recovers accurately the joint positions over the whole sequence, the error for local optimization is significantly larger.

Here the subscripts of N_i and T_i correspond to the respective rows of the matrix or entry of the vector. These equations force the vertex to lie somewhere on the ray going through the camera's center of projection and the pixel position x_i. Since the error of this constraint is depth-dependent and thus not linear in the image plane, we weight each constraint such that the error is 1 for a single pixel difference at the original vertex position.

Enforcing too high weights for our constraints may lead to an over-adaptation in presence of inaccurate silhouettes. We therefore perform several iterations of the deformation, using lower weights. As the silhouette rim points may change after a deformation, we have to recalculate them following each deformation. In all our experiments, we performed 8 iterations and used weights of $\alpha = 0.5$.

The estimation for the next frame is then initiated with the estimated skeleton and an adapted surface model which is obtained by a linear vertex interpolation between the mesh from skeleton-pose estimation $X_i^{t,p}$ and the refined mesh $X_i^{t,r}$, i.e. $X_i^{t+1} = \lambda X_i^{t,r} + (1-\lambda)X_i^{t,p}$. In general, a small value $\lambda = 0.1$ is sufficient and enforces mesh consistency.

We finally remark that the surface estimation uses 2D constraints while the skeleton-based pose estimation (Section 7.2.3) uses 3D constraints. In both cases 3D constraints can be computed faster, but 2D constraints are more accurate. We therefore resort to 3D constraints during skeleton-based pose estimation which only produces an approximate pose and surface estimate, but use 2D constraints during refinement where accuracy matters.

7.2.5 Experiments

For a quantitative and qualitative evaluation of our approach, we have recorded new sequences and used public available datasets for a comparison to the related methods [AST+08] and [VBMP08]. Altogether, we demonstrate the reliability and accuracy of our method on 12 se-

Sequence	Frames	Views	Model	%DOF
Handstand	401	8	Scan	3.3%
Wheel	281	8	Scan	0.2%
Dance	574	8	Scan	4.0%
Skirt	721	8	Scan	0.2%
Dog	60	8	Scan	98.3%
Lock [SH03]	250	8	S-f-S	33.9%
Capoeira1 [AST$^+$08]	499	8	Scan	3.4%
Capoeira2 [AST$^+$08]	269	8	Scan	11.8%
Jazz Dance [AST$^+$08]	359	8	Scan	43.8%
Skirt1 [AST$^+$08]	437	8	Scan	7.2%
Skirt2 [AST$^+$08]	430	8	Scan	6.5%
HuEvaII S4 [SB06]	1258	4	SCAPE	79.3%

Table 7.8: Sequences used for evaluation. The first 5 sequences are newly recorded. The other sequences are public available datasets. The sequences cover a wide range of motion, apparel, subjects, and recording settings. The last column gives the average dimensionality of the search space for the global optimization in percentage of the full search space.

quences with 9 different subjects. An overview of the sequences is given in Table 7.8. The number of available camera views ranges from 4 to 8 cameras and the 3D surface models have been acquired by a static full body laser scan, by a shape-from-silhouette method, or by the SCAPE model. While our newly recorded sequences have been captured with 40Hz at $1004x1004$ pixel resolution, the other sequences are recorded with the settings: 25Hz and $1920x1080$ pixel resolution [SH03], 25Hz and $1004x1004$ pixel resolution [AST$^+$08], or 60Hz and $656x490$ pixel resolution [SB06]. Despite of the different recording settings, the sequences cover various challenging movements from rapid capoeira moves over dancing sequences to a handstand where visual hull approaches are usually prone to topology changes. Furthermore, we have addressed scale issues by capturing the motion of a small dog and wide apparel by three skirt sequences where skeleton-based approaches usually fail. The last column in Table 7.8 gives the achieved dimensionality reduction of the search space for global optimization and indicates the reduced computation time. On the newly recorded sequences, the surface estimation requires 1.7 seconds per frame, the local optimization 3 seconds per frame, and the global optimization 14 seconds for each DOF (maximal 214 seconds per frame). For the skirt sequence, the average computation time for all steps is 9 seconds per frame whereas global optimization without local optimization takes 216 seconds per frame using 15 iterations and 300 particles.
The examples in Figures 7.20, 7.21, and 7.22 show that our approach accurately estimates both skeleton and surface deformation. Even the challenging lock sequence [SH03] can be tracked fully automatically whereas the approach [VBMP08] requires a manual pose correction for 13 out of 250 frames. A visual comparison with a mesh-based method [AST$^+$08] is shown in Figure 7.19. Since this method does not rely on a skeleton, it is free of skinning artifacts and estimates apparel surfaces more accurately. The prior skeleton model in our approach, on the other hand, makes pose recovery of the extremities more accurate. In addition, our method is

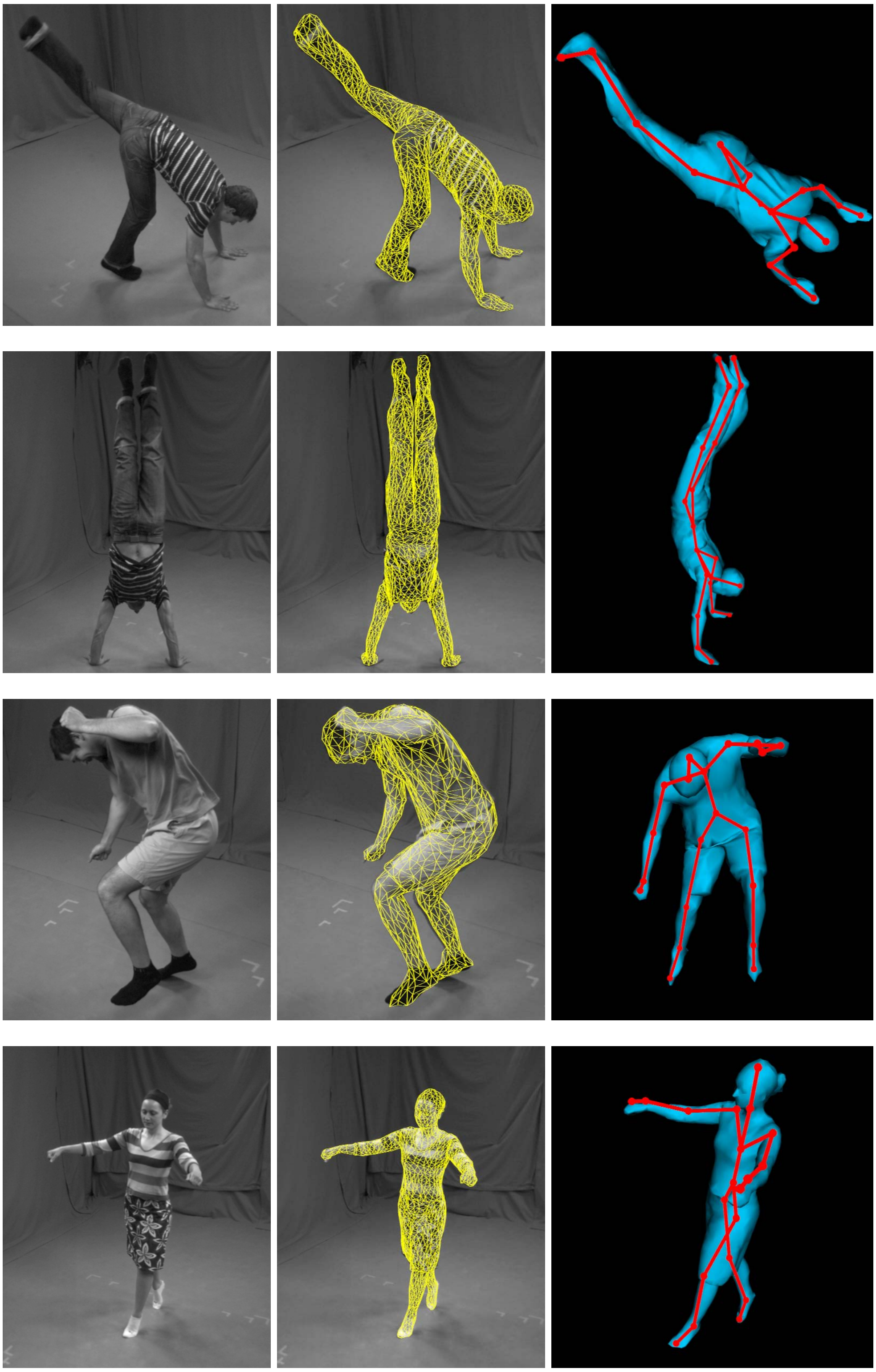


Figure 7.20: Various results of our tracking method. The three pictures of the subjects show input image, adapted mesh overlay, and 3D model with estimated skeleton from a different viewpoint respectively.

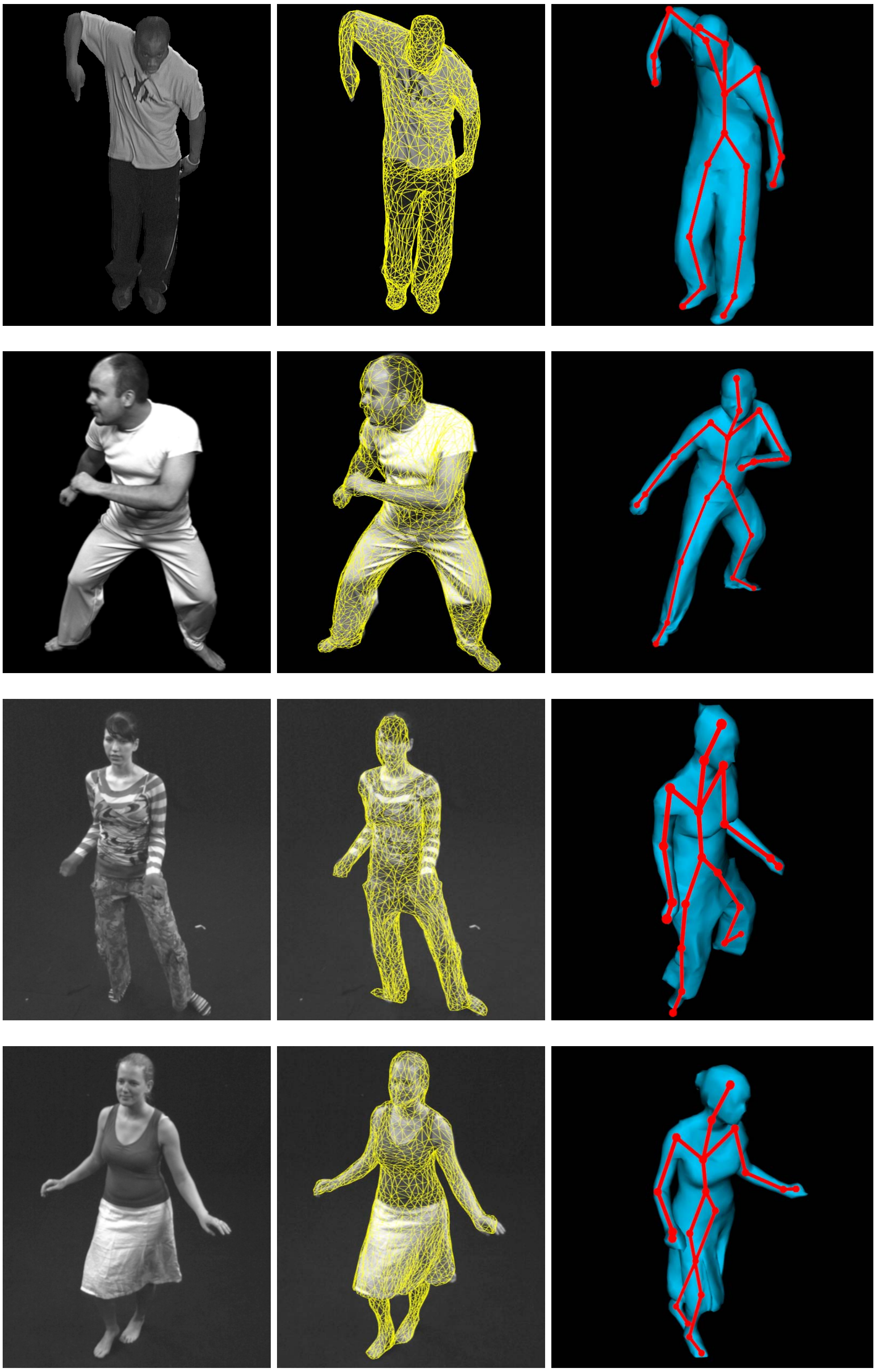


Figure 7.21: Various results of our tracking method. The three pictures of the subjects show input image, adapted mesh overlay, and 3D model with estimated skeleton from a different viewpoint respectively.

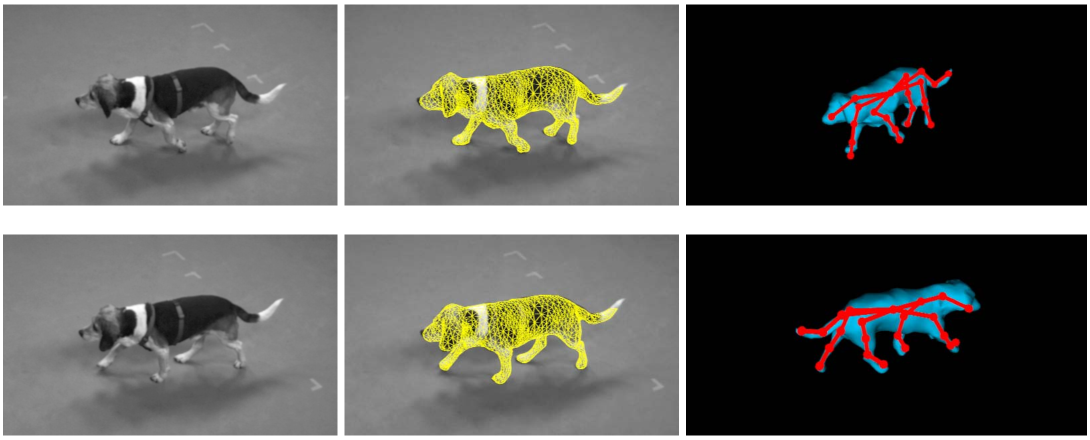


Figure 7.22: Results of our tracking method for the dog sequence. The three pictures of the subjects show input image, adapted mesh overlay, and 3D model with estimated skeleton from a different viewpoint respectively.

over a magnitude faster than the mesh-based approach [AST$^+$08], which requires approximately 10 minutes per frame.

In contrast to [AST$^+$08] and [VBMP08], our algorithm can also handle medium-resolution multi-view sequences with extremely noisy silhouettes like the `HumanEva-II` benchmark [SB06]. The dataset provides a ground truth for 3D joint positions of the skeleton that has been obtained by a marker-based motion capture system that was synchronized with the cameras. The sequence S4 with three subsets contains the motions walking, jogging, and balancing. The average errors for all three subsets are given in Figure 7.19 d). The plot shows that our method provides accurate estimates for the skeleton pose, but it also demonstrates the significant improvement of our optimization scheme compared to local optimization. We finally remark that the errors in Figure 7.19 d) are larger than the one given in Table 7.7 and that the jazz dance sequence taken from [AST$^+$08] also contains some inaccurate estimates for the feet. The errors do not result from the optimization itself, but a silhouette problem in the data. Therefore the functional being optimized, which is dominated by this error-corrupted term, may lead to problems. Since the silhouettes from [SB06] and [AST$^+$08] are very noisy around the feet due to shadows, the feet are not always perfectly estimated. This could be improved by using additional cues or a pose prior as in Section 7.1.

7.3 Summary

In this chapter, we have introduced two tracking systems for model-based human motion capture. Both systems combine the techniques from the previous chapters and outperform the current state-of-the-art. While the first approach has been developed for motion analysis scenarios with noisy silhouette data and is rigorously evaluated on the `HumanEva-II` benchmark [SB06], the second approach demonstrates that global optimization can be efficiently employed even on high quality video footage as shown on a large variety of sequences.

Although interacting simulated annealing provides robust and relatively accurate estimates of the human pose, an even higher precision is only achieved at the expense of high computational cost. To address this problem, we have introduced a multi-layer framework that combines the advantages of global stochastic optimization, local optimization, and filtering. While the first layer relies on stochastic global optimization, the second layer refines the estimates where filtering and local optimization are coupled. The second layer not only increases the accuracy, but also reduces jitter and potential bias from the first layer. Although the approach significantly outperforms the current state-of-the-art on the HumanEva-II benchmark, the computation time limits its application on high quality data, i.e. sequences captured with more than 4 high resolution cameras in a controlled environment. Another aspect that is inherent in any skeleton-based pose estimation method is the limited capacity to handle non-rigid surface deformations. Since garment motion, for instance, is non-rigid and rarely aligned with the motion of the underlying articulated body, these methods often fail if the subject wears wide clothing like a dress.

Hence, we have presented an approach that recovers skeleton pose and surface motion fully-automatically from a multi-view video sequence. To this end, the skeleton motion and the temporal surface deformation are captured in an interleaved manner that improves both accurate skeleton and detailed surface estimation. In addition, we have introduced an optimization scheme for skeleton-based pose estimation that makes automatic processing of large data sets feasible. It reduces the computational burden for global optimization in high dimensional spaces by splitting the skeleton-specific optimization problem into a local optimization problem and a lower dimensional global optimization problem. The reliability of our approach has been evaluated on a large variety of sequences including the HumanEva-II benchmark. The proposed method exceeds the performance of related methods since it allows both accurate skeleton estimation for subjects wearing wide apparel and surface estimation without topology changes for fast movements and noisy silhouettes. This simplifies the acquisition of marker-less motion capture data for applications like character animation and motion analysis. The current limitation of this approach comes from the fact that the pose needs to be explained by the functional being optimized. This means that only body parts can be estimated that are visible. When the feet are occluded by a very long dress, for instance, the estimated feet pose is somehow arbitrary. This could be addressed by taking additional prior knowledge into account, like a motion prior. Furthermore, the approach could also be combined with the multi-layer framework to improve the accuracy on noisy silhouette data.

8

Conclusions

In this work, we have addressed the question *"What is the best way to determine the sequence of human poses that fits a given image sequence best?"* by discussing the following two subquestions in the context of markerless human motion capture with skeleton-based shape models:

1. *What are good cues for human motion capture?*
2. *Is human motion capture a filtering or an optimization problem?*

8.1 What are good cues for human motion capture?

Local optimization can solve human motion capture accurately but it cannot recover from errors. Hence, cues are required that guide the local optimization to the true pose. Furthermore, they need to be robust to occlusions, illumination changes, and clutter and they need to be reliable for homogeneous and structured surfaces. Since no single cue for motion capture like silhouettes, edges, color, motion, and texture meets the demands, a multi-cue integration is necessary for tracking complex objects like humans. For instance, a region-based approach that couples segmentation and pose estimation [RBW07] works well for homogeneous objects but it usually requires many iterations until convergence, which makes the approach very expensive. Particularly for large transformations from frame to frame, the segmentation and consequently the pose estimation usually get stuck in a local optimum. Another problem is ambiguous solutions for symmetric objects.

Hence, we have extended the region-based approach with motion cues. Motion cues are complementary to silhouettes since they perform better on sufficiently structured objects. Furthermore, they can handle large transformations between successive frames and consequently reduce the number of required iterations for optimization. Since the impact of a cue should be large in situations when its extraction is reliable, and small, if the information is likely to be erroneous, we have proposed an adaptive weighting scheme that combines the complementary cues. We have also compared dense and sparse features, namely dense optical flow and local descriptors. The highest accuracy and robustness has been achieved by using sparse and dense features at the same time. Although both are motion cues, they have different strengths. While the estimated optical flow might not be exact in difficult situations, it provides at least enough correspondences for a unique approximate solution. Matches between local descriptors are usually more reliable, but their number is sometimes not sufficient to estimate the pose.

Besides image-based cues, prior knowledge is another important source of information for markerless human motion capture. Although a surface model with an underlying skeleton reduces the search space significantly, it does not take into account physical restrictions on the kinematic model. For instance, both anatomical limits of joints like knees and elbows and unrealistic self-intersections constrain the search space. Instead of modeling the physical restrictions as hard constraints, we allow for the simplification and approximation of the kinematic model by integrating this prior knowledge as soft constraints into filtering and optimization approaches.

Even though the combination of surface-region matching, optical flow, and SIFT tracking provides precise estimates for rigid and articulated objects with homogeneous and structured surfaces, it does not solve the drift problem. Since motion cues rely on image features that are tracked over time, the accumulation of small errors results in a drift away from the target object that cannot be compensated by the region-based features. Hence, we have addressed the drift problem for human motion capture and tracking in the presence of multiple moving objects where the error accumulation becomes even more problematic due to occlusions. To this end, we have proposed an analysis-by-synthesis framework for articulated models that relies on a combination of region-based and motion-based cues. A comparison with other model-based approaches for rigid objects has revealed that the proposed method handles the drift problem better. Our experiments have demonstrated that our framework is not restricted to a single rigid object but tackles the drift problem also for multiple moving objects and humans in challenging scenes containing fast movements, occlusions, and clutter.

Furthermore, we have demonstrated that the proposed multi-cue framework tackles challenging real-world problems and opens up new opportunities for analyzing human motion. For instance, the analysis of crash test videos is an important task for the automotive industry in order to improve the passive safety components of cars. In contrast to conventional marker-based systems which provide only sparse 3D measurements, our approach estimates all six degrees of freedom of dummy body parts like the head. This opens up new opportunities for analyzing pedestrian crashes where many biomechanical effects are not fully understood.

Although the analysis-by-synthesis framework tackles major challenges in human motion capture, there are still some limitations. Drift is only prevented as long as there are enough correspondences between the synthesized and the original image. This means that the approach assumes that there is enough texture information to establish these correspondences. However, even in the worst case where no correspondences are available, the method still behaves as a region-based approach. In general, the method benefits from high resolution images whereas the framerate is less important since large transformations are captured by patch-based matching. Since high-definition cameras are already widely used in contrast to high-speed cameras, assuming a high image resolution is not very restrictive. Another limitation is given by the clothing. Although the approach does not require tight-fitting apparel, it cannot handle arbitrary deformable surfaces or wide apparel like skirts. Since the skeleton-based surface deformation is only an approximation of the real surface deformation, the accuracy of the correspondences between the textured surface and the original image depends on the quality of the approximation. Furthermore, changes of the illumination are implicitly handled by the robustness of the used features. For handling illumination changes that exceed the abilities of the features, the textured model needs to be extended such that the lighting environment is also taken into account.

8.2 Is human motion capture a filtering or an optimization problem?

Markerless human motion capture can be regarded as a filtering or an optimization problem. While the filtering approaches rely often on particle filters, the optimization problem is commonly solved by iterative methods like gradient descent. Hence, we have studied the strength and weakness of filtering and optimization strategies for human motion capture with skeleton-based shape models where we also introduce a novel stochastic global optimization technique.

Filtering The main problem for applying filtering techniques to human motion capture is the need for accurate models. In particular, modeling the human dynamics with Markov kernels is very challenging. While many learning techniques provide solutions only for certain motion patterns, and therefore very application specific solutions, a general motion model is a crucial step towards commercial applications. We have proposed a motion pattern independent prior that compensates at least to some extent for the weak dynamical model. The prior ensures that particles representing a familiar pose are favored such that the state space becomes more constrained. Even though it improves particle filter approaches, it is not a replacement for a good dynamical model. Without such a model, accurate results for human motion capture in reasonable time are hard to achieve. The main advantages of filtering approaches over optimization approaches are the abilities to handle noise and to resolve ambiguities over time. However, a synthetic example has indicated that pixel noise is not a serious problem in the context of motion capture with shape models.

Local Optimization In order to estimate the human pose precisely and accurately, local optimization requires strong cues. Since local optimization itself is very fast, the extraction of the cues dominates the computing time. However, when the sources for feature extraction are limited, only weak cues are available. This might also occur due to low contrast or low size of the object despite high-resolution images. Without strong cues, however, local optima become more critical and need to be resolved by global optimization methods. Another global optimization problem is the initialization of model-based tracking approaches where the pose cannot be predicted from the previous frame. The initialization problem occurs also for texture acquisition which is needed for the analysis-by-synthesis framework. Although static 3D scan devices equipped with cameras or photogrammetric reconstruction techniques can acquire shape and texture, it is more convenient to acquire the texture directly from the video stream since the surface color is likely to change from sequence to sequence in contrast to the human body shape. Therefore, there is a need for global optimization techniques that meet the demands of human pose estimation and tracking.

Global Optimization We have shown that global stochastic optimization, namely *interacting simulated annealing*, is a promising alternative to existing filtering and local optimization approaches for markerless human motion capture. It not only solves the difficult initialization problem that is relevant for texture acquisition as well, but also applies to pose tracking where stable results are achieved with a remarkable accuracy. Indeed, a quantitative comparison with several optimization and particle filtering approaches has revealed that our tracking framework gives significantly better results even for challenging scenes where the silhouette information is unreliable. Since the framework is easy to implement and requires neither excessive preprocessing nor strong assumptions, it is a very general solution to human motion tracking that can be specialized further.

The experiments demonstrate that regarding human motion capture as an optimization problem avoids the modeling problem of filtering approaches. As long as the problem cannot be well approximated by the filtering equations that model the observations and the temporal evolution of the pose, filtering approaches perform poorly for this task. Furthermore, the observation that image noise is not very serious for model-based human motion capture has been supported by our experiments where accurate results have been achieved by global optimization despite significant errors in the silhouettes. Hence, human motion capture with skeleton-based shape models can be very well modeled as an optimization problem.

Local optimization may perform better than global optimization for sequences where local optima are not critical - but this requires high resolution image data where strong cues can be extracted. Apart from that, global optimization is the method of choice where *interacting simulated annealing* takes a leading role. Since interacting simulated annealing approximates a distribution instead of a single value, it has several advantages for pose tracking in comparison to standard optimization techniques. The optimization can be initialized with a large set of hypotheses, up to one hypothesis for each particle. This increases the variety among the particles and helps to recover from large errors or ambiguities as they occur for a small number of cameras. Furthermore, the proposed tracking system preserves the uncertainty that arises from the prediction and optimization and takes it into account for the next frame. This increases the robustness and efficiency of the approach compared to other techniques like fast simulated annealing as we have demonstrated in our experiments.

Optimization and Filtering Although interacting simulated annealing performs very well compared to local optimization and filtering approaches, there are still some limitations that need to be mentioned. When the estimates are observed over time, some jitter is noticeable which is typical for stochastic approaches like interacting simulated annealing that sample from a distribution of interest. Variations between estimates of two frames might also occur, when the tracker recovers from an ambiguity in the previous frame. Moreover, interacting simulated annealing provides estimates close to the global optimum in reasonable time, but more precise estimates are only achieved at a very high cost.

To address these problems, we have introduced a multi-layer framework that combines the advantages of global stochastic optimization, local optimization, and filtering. While the first layer relies on stochastic global optimization, the second layer refines the estimates where filtering and local optimization are coupled. The second layer not only increases the accuracy, but also reduces jitter and potential bias from the first layer. Although the approach significantly outperforms the current state-of-the-art on the `HumanEva-II` benchmark, the computation time limits its application on high quality data, i.e. sequences captured with more than 4 high resolution cameras in a controlled environment. Another aspect that is inherent in any skeleton-based pose estimation method is the limited capacity to handle non-rigid surface deformations. Since garment motion, for instance, is non-rigid and rarely aligned with the motion of the underlying articulated body, these methods often fail if the subject wears wide clothing like a dress.

In order to address these issues, we have presented an approach that recovers skeleton pose and surface motion fully-automatically from a multi-view video sequence. To this end, the skeleton motion and the temporal surface deformation are captured in an interleaved manner that improves both accurate skeleton and detailed surface estimation. In addition, we have introduced an optimization scheme for skeleton-based pose estimation that makes automatic processing of large data sets feasible. It reduces the computational burden for global optimization in high dimensional spaces by splitting the skeleton-specific optimization problem into a local optimization problem and a lower dimensional global optimization problem. The reliability of our approach has been evaluated on a large variety of sequences including the `HumanEva-II` benchmark. The proposed method exceeds the performance of related methods since it allows both accurate skeleton estimation for subjects wearing wide apparel and surface estimation without topology changes for fast movements and noisy silhouettes. This simplifies the acquisition of markerless motion capture data for applications like character animation and motion analysis. The current limitation of this approach comes from the fact that the pose needs to be explained by the functional being optimized. This means that only body parts can be estimated that are visible. When the feet are occluded by a very long dress, for instance, the estimated feet pose is arbitrary. This could be addressed by taking into account additional prior knowledge, like a motion prior. Furthermore, the approach could also be combined with the multi-layer framework to improve the accuracy on noisy silhouette data.

8.3 What is the best way for estimating human poses?

We eventually emphasize that the results and conclusions need to be viewed in the context of the stated assumptions, namely the availability of a skeleton-based shape model and at least two calibrated and synchronized cameras. While interacting simulated annealing in combination with filtering and local optimization performs very well under these assumptions, this might not be the case for human motion capture in general, even though the convergence properties of the approaches are generally valid. Nevertheless, we have proposed three human motion capture systems, namely an analysis-by-synthesis framework, a multi-layer framework, and a framework for surface and skeleton estimation, that cover a wide range of settings that are relevant for real-world applications. The systems not only simplify the acquisition of markerless motion capture data for applications like character animation and motion analysis but also open up new possibilities as in the crash test example. We finally remark that the presented work goes beyond human motion capture since it deals with solving a complex optimization problem in a 30-40 dimensional search space efficiently. In particular, the global optimization example and the discussion on the asymptotic behavior of interacting simulated annealing in Section 6.1 are of general interest.

8.4 Future Challenges

In this section, we briefly discuss extensions, related problems, and aspects that are out of scope of this work. Getting rid of the assumptions on the model and the camera setting is clearly the biggest challenge.

Calibrated and Synchronized Cameras Even though indoor and outdoor sequences can be captured with pre-calibrated and synchronized cameras, human motion capture with off-the-shelf handheld video cameras is an interesting low-budget alternative without the need of expensive hardware. This allows the acquisition of markerless motion capture data not only by nearly anyone but also in arbitrary environments since no heavy equipment is required. While hardware synchronization is basically a matter of money, moving cameras increase the size of the capture area since the cameras can follow the subject. This is particularly relevant for sports science where the athletes typically perform on a large area. In order to capture human motion with unsynchronized and moving cameras, the cameras need to be calibrated and registered in a common world coordinate system [THWS08]. The synchronization of the video streams can be achieved via the audio streams recorded by the cameras. Although the calibration and synchronization is not as accurate as in a static setup, a variant of the multi-cue approach from Section 5.3 can be used to estimate the human poses as shown in Figure 8.1. A more detailed description of the approach is given in [HRT$^+$09]. This could be further generalized by taking video clips from the internet that show different views of a sports event.

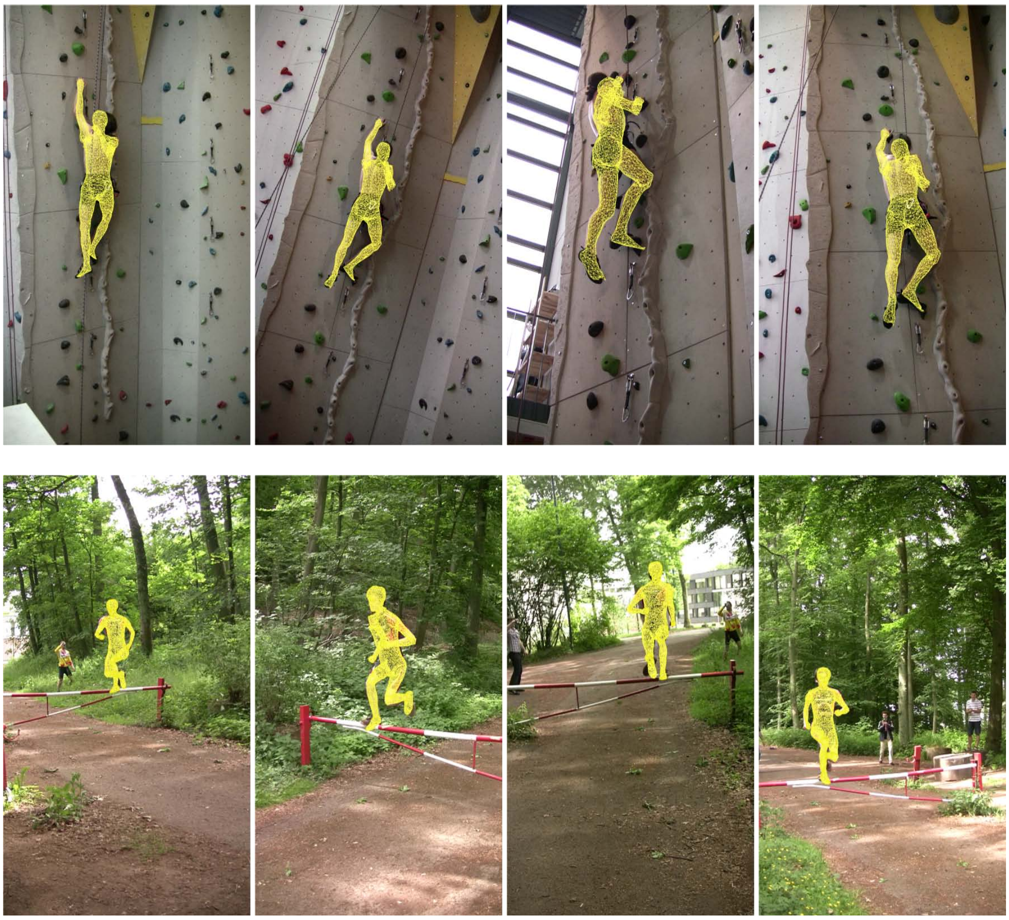


Figure 8.1: Two examples for human motion capture from sequences that have been recorded with four handheld cameras. The four views for one time instance are shown. **Top:** Indoor-climbing. **Bottom:** Running and jumping on a forest trail.

Surface Model Although the acquisition of a surface model is out of the scope of this work, our approaches cope with models acquired by a static full body laser scan, shape-from-silhouette methods, or human shape databases. The type of model basically depends on the application, but it is not essential for the presented algorithms. In principle, the initialization problem can be extended by estimating not only the pose but also the shape using the approach from Section 6.2. To this end, the SCAPE model [ASK^{+}05] or related datasets [MCA07, HSS^{+}09] can be used. The shape parameters increase the search space only by a few dimensions since the space of human shapes can be mapped to a low dimensional manifold. These datasets, however, are only available for humans and not for other subjects like animals.

Skeleton and Topology In this work, we have assumed that a skeleton-based shape model is available that already provides the skeleton and the topology of the subject. The type and degrees of freedom of the skeleton depend on the application. For instance, a skeleton for animation [BP07] differs from a biomechanical skeleton model [RFDC05]. Skeletons for biomechanical studies have usually more degrees of freedom than the one we have used in our experiments, but some degrees are very constrained. Since it is straightforward to integrate these constraints into our approaches, we expect that our algorithms work for biomechanical skeleton models as well as for standard skeleton models. Since we have assumed that the skeleton and the topology of the subject are known, our approaches do not suffer from topology changes in contrast to visual hull approaches. In a more general setting, however, the topology is not known a priori. In this case, the topology and the underlying skeleton need to be estimated from the video footage. This means that one still obtains a consistent topology as it is needed for many applications, but the topology is estimated from the evolution of the surface [VZBH08]. Using the approach from Section 7.2, one might begin with an initial reconstruction of the topology and the skeleton from the images of the first frame. The refinement from Section 7.2 needs then to be generalized to allow for skeleton and topology changes.

Deformable Surfaces The motion of arbitrary deformable objects without an underlying skeleton can still be approximated by a skeleton-based surface model. As in Section 7.2, the surface deformations can be split into large scale deformations that are captured by the global skeleton pose estimation, and small scale deformations that are captured by surface refinement. We also emphasize that interacting simulated annealing works with any representation of the surface. Although the human skeleton is a natural representation as illustrated in Figure 1.2, other representations that reduce the dimensions of the search space are also suitable as long as the mappings between the surface mesh and the search space can be computed very efficiently.

Details The camera resolution usually prevents capturing very small details like fingers and facial expressions which are important for character animation. This could be addressed by representing the 3D hand shapes and 3D faces by examples where a mapping between the image data and the examples needs to be learned. Moreover, the estimated examples need to be merged with the surface in order to obtain a seamless mesh. Another approach is an active camera system where additional cameras focus only on one body part. To this end, the cameras need to follow specific body parts like the head or one hand. A camera calibration as in Figure 8.1 might not even be necessary since the positions of the body parts are already known.

Motion Transfer Motion transfer is a very important issue for character animation. Our captured performances can be easily edited and used in animation frameworks typical for games and movies, which are almost exclusively skeleton-based. The realism, however, is limited by the skeleton-based deformation. While the approach from Section 7.2 reconstructs skeleton motion and detailed time-varying surface geometry, the physically plausible cloth behavior is not maintained when the skeleton is edited. Nevertheless, our approach provides a convenient opportunity to acquire a rigged fully-animatable virtual double of a real person that comprises of a skeleton-based representation for the actual body parts and a physically-based simulation model for the apparel. Having such a model, new animations can be created with new body motion and cloth deformation that look as realistic as in the captured sequence as outlined in Figure 8.2. To this end, the captured data needs to be analyzed to identify non-rigidly deforming pieces of apparel on the geometry. The parameters of a physically-based cloth simulation model like [MHHR07] can then be estimated for each piece of apparel from the data obtained by our approach from Section 7.2. This is also an elegant way to get rid of skinning artifacts that might be still visible after the surface refinement.

Multi Objects Although we have focused on tracking single subjects, the extension to multiple objects is straightforward. Indeed, the sequences in Sections 5.5 and 5.6 already contain mul-

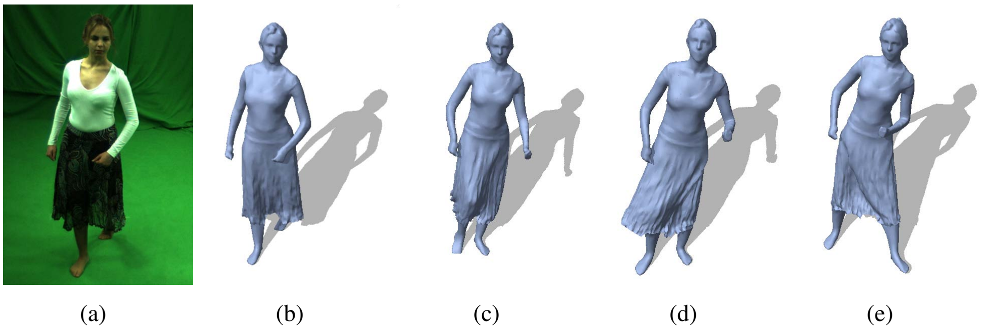


Figure 8.2: **a)** Input frame from captured sequence. **b)** The reconstructed animatable model simulating the same pose. **c)-e)** Subsequent frames of a newly created animation for which only motion parameters of the underlying skeleton were given.

tiple moving objects. Furthermore, tracking two objects simultaneously is simpler than tracking a single object without the knowledge of the second object.

Cues The cues that have been discussed in Chapter 5 are visual cues that can be extracted from the image data of a standard camera. However, additional cues from other devices like time-of-flight cameras, acceleration sensors, or infrared cameras can also be used. In particular for the crash test analysis example in Section 5.6, the fusion of visual information and the data from the acceleration sensor is suitable.

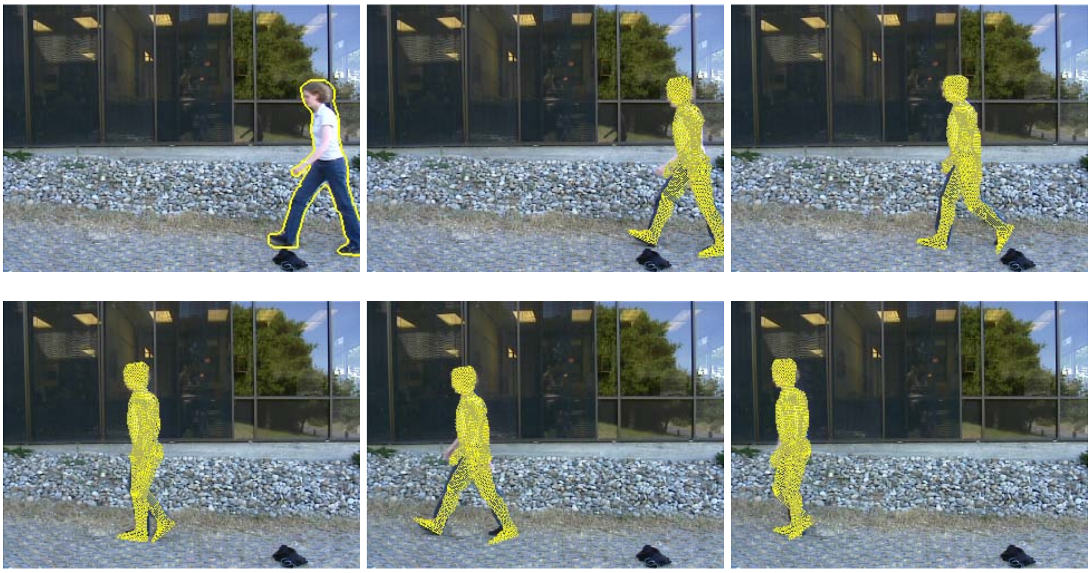


Figure 8.3: Monocular standard test sequence with self-occlusions. Input image with estimated contour and tracking results.

Monocular and Motion Priors The presented approaches are not limited to a multi-camera setting. Only the computation of the bounding box for the pose initialization in Section 6.2

requires at least two camera views, but this could be generalized by specifying the view frustum for the camera, i.e., the depth range needs to be given. Since projections on a GPU are anyway specified by a view frustum, it does not impose an additional restriction. However, priors become necessary in monocular scenes where image cues for some body parts are missing. Figure 8.3 shows an example where the right arm of the person is fully occluded. Since motion priors usually impose strong restrictions on the type of motions that can be captured, we have not used these priors even though they can be integrated in our approaches similar to the learned constraints of the skeleton from Section 4.2. Figure 8.3 shows tracking results where the system from Section 5.3 has been supported by a kernel density estimate on a set of walking motions [BRCS07]. When motion priors are needed or desired, the multi-layer framework from Section 7.1 is appealing for several reasons. While interacting simulated annealing is powerful enough to track sequences without motion priors, these priors can simplify the search and reduce the computation time. When the motion pattern can be detected on-the-fly, the computational effort can be adapted depending on whether the captured performance is part of the learned motion prior or not. This would improve over existing methods with motion priors that dramatically fail when the performed motion is not part of the training data. Furthermore, the second layer would reduce the bias introduced by the priors.
For applications where only one camera view is available, it is questionable whether 3D human motion capture data is needed at all. Tracking humans in monocular sequences is usually only possible with a certain amount of learning and very limited depth accuracy. However, when a large training set is required anyway, learning the mapping from the 2D image data directly to the space of interest might be more suitable than model fitting with strong priors. Indeed, bottom-up approaches rely on detectors for humans or body parts [FDLF07, RFZ07, ARS08] and they work very well for monocular sequences. Since detectors basically produce many hypotheses including false positives as shown in Figure 8.4, particle-based approaches like interacting simulated annealing or nonparametric belief propagation [SBR$^+$04] are very suitable to integrate this information for tracking, even for several camera views.

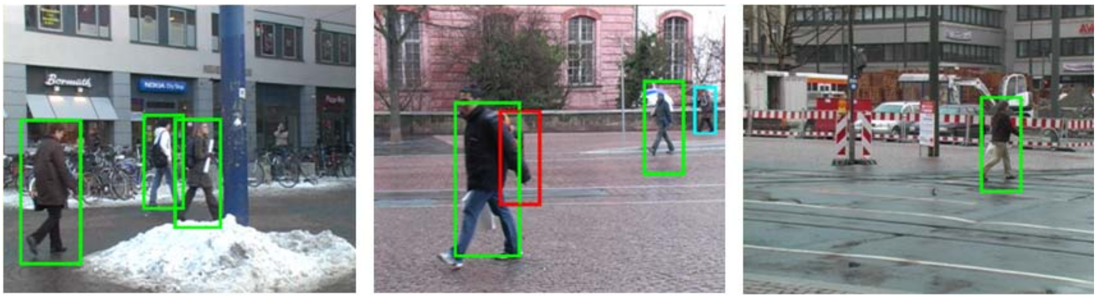


Figure 8.4: Detected pedestrians in monocular scenes [GL09]. The image in the middle contains correctly detected pedestrians *(green)*, false positives *(red)*, and missed detections *(cyan)*.

Real-time Our current implementation of interacting simulated annealing does not achieve real-time performance on a single-core PC. Although a partial implementation on a GPU has reduced the computation time to around 2 seconds per frame for the `Maria` sequences from Section 6.3.2, the required number of energy evaluations per frame is still too large for real-time performance, namely 15 iterations × 200 particles × 5 views. Since, however, the particles and the camera views can be evaluated in parallel, the computation time might be further reduced by using a cluster of PCs, at least one for each camera view. Instead of increasing the hardware requirements, a lower dimensional representation of the surface model than the one obtained by the skeleton might reduce the number of required particles, and consequently the computation time. One might also consider other kernels than a Gaussian kernel for the mutation step to increase the performance, similar to related simulated annealing approaches where other statistics have improved the convergence.

Evaluation Quantitative evaluations are very important to measure the progress in human motion capture. Since the ultimate goal is the design of a markerless tracking system that captures the motion of a human in an outdoor scene with the accuracy of a commercial marker-based system in an indoor scene, the accuracy must be compared to marker-based systems. One of the standard datasets that use marker-based motion capture data as ground-truth is the recently released `HumanEva-II` benchmark [SB06]. Although there are only very few approaches at the moment that can track the sequences S2 and S4 with a reasonable accuracy, one can expect that an overfitting will occur over the next 5 years due to the limited variation of human motion. Hence, larger datasets are required that cover not only more types of motion but also several settings from one camera to eight cameras. The quality of the ground-truth of the benchmark is also affected by the placement of the markers and the relative motion of skin and cloth with respect to the underlying bones. Instead of comparing the 3D positions of the joint centers estimated by the marker-based system, one could also directly compare the marker positions. This would also reveal whether limb rotations along the axis are correctly estimated. While the skeleton pose can be validated by a standard marker-based system, approaches that estimate the surface like our approach from Section 7.2 require more advanced techniques. Since a large number of markers strongly interferes with the experiments, the ground-truth could be acquired by tracking a hidden fluorescent texture, as it is already used for commercial systems [Mov08]. Another alternative are synthetic sequences that are generated by ray tracing and achieve movie quality.

Applications We have presented approaches that acquire accurate markerless motion capture data that can be used for a large variety of applications like action recognition, rehabilitation, sports science, biomechanical research, or character animation. Hence, it is interesting to see how the approaches fit the demands of the applications, help to make current processes more efficient, and open up new opportunities. Besides human motion capture, the systems might also be applied to other problems like robotics where filtering or local optimization have shortcomings. In general, some results of this work might be relevant to any application where a complex optimization problem needs to be solved efficiently.

Bibliography

[Ack87] D. Ackley. *A connectionist machine for genetic hillclimbing*. Kluwer, Boston, 1987.

[ACP02] B. Allen, B. Curless, and Z. Popović. Articulated body deformation from range scan data. *ACM Transactions on Graphics*, 21(3):612–619, 2002.

[Ari07] Aristotle. *On the Motion of Animals*. eBooks@Adelaide, 2007. Translated by A. Farquharson. Originally published 350 B.C.E.

[Ari08] Ariel dynamics: Apas - ariel performance analysis system. http://www.arielnet.com, March 2008.

[ARS08] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[ASK$^+$05] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Transactions on Graphics*, 24(3):408–416, 2005.

[AST$^+$08] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *International Conference on Computer Graphics and Interactive Techniques*, pages 1–10, 2008.

[AT06] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58, 2006.

[ATSS07] E. De Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[BA83] P. Burt and E. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[BA96] M. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision Image Understanding*, 63(1):75–104, 1996.

[Bau90] H. Bauer. *Maß- und Integrationstheorie*. de Gruyter, 1990.

[Bau91] H. Bauer. *Wahrscheinlichkeitstheorie*. de Gruyter, 4 edition, 1991.

[Bau96] H. Bauer. *Probability Theory*. de Gruyter, Baton Rouge, 1996.

[BB06] A. Balan and M. Black. An adaptive appearance model approach for model-based articulated object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 758–765, 2006.

[BB08] A. Balan and M. Black. The naked truth: Estimating body shape under clothing. In *European Conference on Computer Vision*, pages 15–29, 2008.

[BBHS07] A. Balan, M. Black, H. Haussecker, and L. Sigal. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. In *International Conference on Computer Vision*, pages 1–8, 2007.

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36, 2004.

[BC08] L. Ballan and G. Cortelazzo. Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In *International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.

[BETG08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[BF87] W. Braune and O. Fischer. *The Human Gait*. Springer, 1987. Translated by P. Maquet. Originally published 1898: *Der Gang des Menschen*.

[BFB94] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.

[Bil95] P. Billingsley. *Probability and Measure*. Wiley Series in Probability and Mathematical Statistics. Wiley, 3 edition, 1995.

[Bir98] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–237, 1998.

[BJ98] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[BKMG07] M. Bray, E. Koller-Meier, and L. Van Gool. Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding*, 106(1):116–129, 2007.

[BKT06] M. Bray, P. Kohli, and P. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *European Conference on Computer Vision*, pages 642–655, 2006.

[BM92] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[BM98] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8–15, 1998.

[BMP04] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, 2004.

[Bor86] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3), 1986.

[Bor89] A. Borelli. *On the Movement of Animals*. Springer, 1989. Translated by P. Maquet. Originally published 1680: *De Motu Animalium*.

[Bou08] J.-Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc, March 2008.

[BP07] I. Baran and J. Popović. Automatic rigging and animation of 3d characters. *ACM Transactions on Graphics*, 26(3):72, 2007.

[BRCS06] T. Brox, B. Rosenhahn, D. Cremers, and H.-P. Seidel. High accuracy optical flow serves 3-d pose tracking: Exploiting contour and flow based constraints. In *European Conference on Computer Vision*, pages 98–111, 2006.

[BRCS07] T. Brox, B. Rosenhahn, D. Cremers, and H.-P. Seidel. Nonparametric density estimation with adaptive anisotropic kernels for human motion tracking. In *International Workshop on Human Motion*, volume 4814 of *LNCS*, pages 152–165. Springer, 2007.

[BRDW03] T. Brox, M. Rousson, R. Deriche, and J. Weickert. Unsupervised segmentation incorporating colour, texture, and motion. In *Computer Analysis of Images and Patterns*, volume 2756 of *LNCS*, pages 353–360. Springer, 2003.

[Bre97] C. Bregler. Learning and recognizing human dynamics in video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[BRGC09] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers. Combined region- and motion-based 3d tracking of rigid and articulated objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[BRKC06] T. Brox, B. Rosenhahn, U. Kersting, and D. Cremers. Nonparametric density estimation for human pose tracking. In *Pattern Recognition (DAGM)*, volume 4174 of *LNCS*, pages 546–555. Springer, 2006.

[Bro71] D. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[BRW05] T. Brox, B. Rosenhahn, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose estimation. In *Pattern Recognition (DAGM)*, volume 3663 of *LNCS*, pages 109–116. Springer, 2005.

[BS93] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[BS08] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, 2008.

[BSB05] A. Balan, L. Sigal, and M. Black. A quantitative evaluation of video-based 3d person tracking. In *IEEE Workshop on VS-PETS*, pages 349–356, 2005.

[BSB+07] A. Balan, L. Sigal, M. Black, J. Davis, and H. Haussecker. Detailed human shape and pose from images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[BW05] A. Bruhn and J. Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *International Conference on Computer Vision*, pages 749–755, 2005.

[BW06a] T. Brox and J. Weickert. A TV flow based local scale estimate and its application to texture discrimination. *Journal of Visual Communication and Image Representation*, 17(5):1053–1073, 2006.

[BW06b] A. Bruhn and J. Weickert. Confidence measures for variational optic flow methods. In R. Klette, R. Kozera, L. Noakes, and J. Weickert, editors, *Geometric Properties for Incomplete Data*, pages 283–297. Springer, 2006.

[CBK05] G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part ii: Applications to human modeling and markerless motion tracking. *International Journal of Computer Vision*, 63(3):225–245, 2005.

[CF01] K. Choo and D. Fleet. People tracking using hybrid monte carlo filtering. In *International Conference on Computer Vision*, pages 321–328, 2001.

[CG99] D. Crisan and M. Grunwald. Large deviation comparison of branching algorithms versus resampling algorithms: Application to discrete time stochastic filtering. Technical Report TR1999-9, Statistical Laboratory, Cambridge University, U.K., 1999.

[CKBH00] G. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2720, 2000.

[CL04] P. Chigansky and R. Liptser. Stability of nonlinear filters in nonmixing case. *The Annals of Applied Probability*, 14(4):2038–2056, 2004.

[CMC+06] S. Corazza, L. Mündermann, A. Chaudhari, T. Demattio, C. Cobelli, and T. Andriacchi. A markerless motion capture system to study musculoskeletal biomechanics: Visual hull and simulated annealing approach. *Annals of Biomedical Engineering*, 34(6):1019–1029, 2006.

[CML99] D. Crisan, P. Del Moral, and T. Lyons. Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields*, 5(3):293–319, 1999.

[CMU08] CMU. Graphics lab motion capture database, 2008. http://mocap.cs.cmu.edu.

[CR99] T.-J. Cham and J. Rehg. A multiple hypothesis approach to figure tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 239–245, 1999.

[CRD07] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007.

[CRM00] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 142–149, 2000.

[CSA00] M. Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000.

[CTMS03] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM Transactions on Graphics*, 22(3):569–577, 2003.

[CV01] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[DBR00] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1144–1149, 2000.

[DC01] T. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *International Conference on Computer Vision*, pages 315–320, 2001.

[DCM05] R. Douc, O. Cappe, and E. Moulines. Comparison of resampling schemes for particle filtering. In *International Symposium on Image and Signal Processing and Analysis*, pages 64–69, 2005.

[DDDS03] P. David, D. DeMenthon, R. Duraiswami, and H. Samet. Simultaneous pose and correspondence determination using line feature. In *International Conference on Computer Vision*, pages 424–431, 2003.

[DFG01] A. Doucet, N. De Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[DM00] D. Decarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Jorunal of Computer Vision*, 38(2):99–127, 2000.

[DR05] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, 2005.

[FDLF07] A. Fossati, M. Dimitrijevic, V. Lepetit, and P. Fua. Bridging the gap between detection and tracking for 3d monocular video-based motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[FH00] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 20662073, 2000.

[FH04] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical Report TR2004-1963, Cornell Computing and Information Science, 2004.

[FNGD00] J. De Freitas, M. Niranjan, A. Gee, and A. Doucet. Sequential monte carlo methods to train neural network models. *Neural Computation*, 12(4):955–993, 2000.

[FWA03] D. Fisher, M. Williams, and T. Andriacchi. The therapeutic potential for changing patterns of locomotion: An application to the acl deficient knee. In *ASME Bioengineering Conference*, 2003.

[Gab46] D. Gabor. Theory of communication. *Journal of the Institute for Electrical Engineers*, 93(3):429–459, 1946.

[Gav96] D. Gavrila. *Vision-based 3-D Tracking of Humans in Action*. PhD thesis, University of Maryland, College Park, 1996.

[Gav99] D. Gavrila. The visual analysis of human movement: a survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[GB98] H. Gregory and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[GBG08] S. Gehrig, H. Badino, and J. Gall. *Human Motion - Understanding, Modeling, Capture and Animation*, chapter Accurate and Model-Free Pose Estimation of Crash Test Dummies, pages 453–473. Springer, 2008.

[GBP06] S. Gehrig, H. Badino, and P. Paysan. Accurate and model-free pose estimation of small objects for crash video analysis. In *Britsh Machine Vision Conference*, 2006.

[GBUP95] L. Goncalves, E. Di Bernardo, E. Ursella, and P. Perona. Monocular tracking of the human arm in 3d. In *International Conference on Computer Vision*, pages 764–770, 1995.

[GD96] D. Gavrila and L. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[Gid95] B. Gidas. *Topics in Contemporary Probability and Its Applications*, chapter 7: Metropolis-type Monte Carlo Simulation Algorithms and Simulated Annealing, pages 159–232. CRC Press, Boca Raton, 1995.

[GL09] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[GO04] F. Le Gland and N. Oudjane. Stability and uniform approximation of nonlinear filters using the hilbert metric and application to particle filters. *The Annals of Applied Probability*, 14(1):144–187, 2004.

[Gol80] H. Goldstein. *Classical Mechanics*. Addison-Wesley, Reading, MA, 2 edition, 1980.

[GPS$^+$07] J. Gall, J. Potthoff, C. Schnörr, B. Rosenhahn, and H.-P. Seidel. Interacting and annealing particle filters: Mathematics and a recipe for applications. *Journal of Mathematical Imaging and Vision*, 28(1):1–18, 2007.

[Gra18] H. Gray. *Anatomy of the Human Body.* Philadelphia: Lea Febiger, 1918. Online published by Bartleby.com, 2000. www.bartleby.com/107.

[GRBS06] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Learning for multi-view 3d tracking in the context of particle filters. In *International Symposium on Visual Computing*, volume 4292 of *LNCS*, pages 59–69. Springer, 2006.

[GRBS08] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and filtering for human motion capture – a multi-layer framework. *International Journal of Computer Vision*, 2008.

[GRGS08] J. Gall, B. Rosenhahn, S. Gehrig, and H.-P. Seidel. Model-based motion capture for crash test video analysis. In *Pattern Recognition*, volume 5096 of *LNCS*, pages 92–101. Springer, 2008.

[GRS06] J. Gall, B. Rosenhahn, and H.-P. Seidel. Robust pose estimation with 3d textured models. In *IEEE Pacific-Rim Symposium on Image and Video Technology*, volume 4319 of *LNCS*, pages 84–95. Springer, 2006.

[GRS07] J. Gall, B. Rosenhahn, and H.-P. Seidel. Clustered stochastic optimization for object recognition and pose estimation. In *Pattern Recognition*, volume 4713 of *LNCS*, pages 32–41. Springer, 2007.

[GRS08a] J. Gall, B. Rosenhahn, and H.-P. Seidel. Drift-free tracking of rigid and articulated objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[GRS08b] J. Gall, B. Rosenhahn, and H.-P. Seidel. *Human Motion - Understanding, Modeling, Capture and Animation*, chapter An Introduction to Interacting Simulated Annealing, pages 319–343. Springer, 2008.

[GSA$^+$09] J. Gall, C. Stoll, E. De Aguiar, B. Rosenhahn, C. Theobalt, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[GSD03] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. In *International Conference on Computer Vision*, pages 641–648, 2003.

[GSS93] N. Gordon, D. Salmond, and A. Smith. Novel approach to non-linear/non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, 1993.

[HBG+00] A. Hilton, D. Beresford, D. Gentils, R. Smith, W. Sun, and J. Illingworth. Whole-body modelling of people from multi-view images to populate virtual worlds. *Visual Computer, International Journal of Computer Graphics*, 16(7):411–436, 2000.

[HM96] S. Herbert and L. McKernan, editors. *Who's Who of Victorian Cinema: A Worldwide Survey*. BFI Publishing, 1996.

[Hog83] D. Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[HRT+09] N. Hasler, B. Rosenhahn, T. Thormählen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[HS81] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17, 185-203 1981.

[HS97] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, 1997.

[HSS+09] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. *Computer Graphics Forum*, 2(28), 2009.

[IB96] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, pages 343–356, 1996.

[IB98] M. Isard and A. Blake. A smoothing filter for condensation. In *European Conference on Computer Vision*, pages 767–781, 1998.

[IF01] S. Ioffe and D. Forsyth. Human tracking with mixtures of trees. In *International Conference on Computer Vision*, pages 690–695, 2001.

[JBY96] S. Ju, M. Black, and Y. Yacoob. Cardboard people: a parameterized model of articulated image motion. In *International Conference on Automatic Face and Gesture Recognition*, pages 38–44, 1996.

[JFEM03] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, 2003.

[Joh76] G. Johansson. Visual motion perception. *Scientific American*, 232:76–88, 1976.

[Kal60] R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[KBG05] R. Kehl, M. Bray, and L. Van Gool. Full body tracking from multiple views using stochastic sampling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 129–136, 2005.

[KCvO07] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan. Skinning with dual quaternions. In *Symposium on Interactive 3D graphics and games*, pages 39–46, 2007.

[KFY$^+$05] J. Kim, J. Fisher, A. Yezzi, M. Cetin, and A. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Transactions on Image Processing*, 14(10):1486–1502, 2005.

[KJV83] S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[KM96] I. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 81–87, 1996.

[KM98] I. Kakadiaris and D. Metaxas. Three-dimensional human body model acquisition from multiple views. *International Journal of Computer Vision*, 30(3):191–218, 1998.

[KMI93] Y. Kameda, M. Minoh, and K. Ikeda. Three dimensional pose estimation of an articulated object from its silhouette image. In *Asian Conference on Computer Vision*, pages 612–615, 1993.

[KRH08] D. Knossow, R. Ronfard, and R. Horaud. Human motion tracking with a kinematic parameterization of extremal contours. *International Journal of Computer Vision*, 79(3):247–269, 2008.

[KS04] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 506–513, 2004.

[LC85] H. Lee and Z. Chen. Determination of 3d human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 30(2):148–168, 1985.

[LCF00] J. Lewis, M. Cordner, and N. Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Conference on Computer Graphics and Interactive Techniques*, pages 165–172, 2000.

[LE07] C.-S. Lee and A. Elgammal. Modeling view and posture manifolds for tracking. In *International Conference on Computer Vision*, pages 1–8, 2007.

[LESC04] G. Loy, M. Eriksson, J. Sullivan, and S. Carlsson. Monocular 3d reconstruction of human motion in long action sequences. In *European Conference on Computer Vision*, pages 442–455, 2004.

[LK81] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[LLF05] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 775–781, 2005.

[LN06] M. Lee and R. Nevatia. Human pose tracking using multi-level structured models. In *European Conference on Computer Vision*, pages 368–381, 2006.

[Low87] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[Low91] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

[Low99] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, 1999.

[Low04] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LPF04] V. Lepetit, J. Pilet, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–250, 2004.

[LRD99] F. Lerasle, G. Rives, and M. Dhome. Tracking of human limbs by multiocular vision. *Computer Vision and Image Understanding*, 75(3):229–246, 1999.

[LRF93] H. Li, P. Roivainen, and R. Forcheimer. 3-d motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):545–555, 1993.

[LSLF08] P. Lagger, M. Salzmann, V. Lepetit, and P. Fua. 3d pose refinement from reflections. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[LYST06] R. Li, M. Yang, S. Sclaroff, and T. Tian. Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers. In *European Conference on Computer Vision*, pages 137–150, 2006.

[Mac00] J. MacCormick. *Probabilistic models and stochastic algorithms for visual tracking*. PhD thesis, University of Oxford, 2000.

[Mar73] E.-J. Marey. *La Machine Animale, Locomotion Terrestre et Aérienne*. G. Baillière, Paris, 1873.

[Mar94] E.-J. Marey. *Le mouvement*. G. Masson, Paris, 1894.

[MB04] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[MBCM99] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2d-3d model-based approach. In *International Conference on Computer Vision*, pages 262–268, 1999.

[MCA07] L. Mündermann, S. Corazza, and T. Andriacchi. Accurately measuring human movement using articulated icp with soft-joint constraints and a repository of articulated models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[MDFW00] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. A. Wan. The unscented particle filter. In *Advances in Neural Information Processing Systems*, pages 584–590, 2000.

[MDN97] D. Meyer, J. Denzler, and H. Niemann. Model based extraction of articulated objects in image sequences for gait analysis. In *International Conference on Image Processing*, page 7881, 1997.

[MG01a] T. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

[MG01b] P. Del Moral and A. Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. *Annales de l'Institut Henri Poincaré (B), Probabilités et statistiques*, 37(2):155–194, 2001.

[MHHR07] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.

[MHK06] T. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[MI00] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *European Conference on Computer Vision*, pages 3–19, 2000.

[MLS94] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL, 1994.

[MM00] P. Del Moral and L. Miclo. Branching and interacting particle systems approximations of feynman-kac formulae with applications to non linear filtering. In *Séminaire de Probabilités XXXIV*, number 1729 in Lecture Notes in Mathematics, pages 1–145. Springer, 2000.

[MM06] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1052–1062, 2006.

[MMG06] B. Merry, P. Marais, and J. Gain. Animation space: A truly linear framework for character animation. *ACM Transactions on Graphics*, 25(4):1400–1423, 2006.

[Mor96] P. Del Moral. Non linear filtering: Interacting particle solution. *Markov Processes and Related Fields*, 2(4):555–580, 1996.

[Mor98] P. Del Moral. Measure-valued processes and interacting particle systems. application to nonlinear filtering problems. *Annals of Applied Probability*, 8(2):438–495, 1998.

[Mor04] P. Del Moral. *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. Springer, New York, 2004.

[Mot08] Motion analysis corporation: Eagle and hawk. http://www.motionanalysis.com, March 2008.

[Mov08] Mova contour reality capture system. http://www.mova.com, March 2008.

[MP96] E. Memin and P. Perez. Robust discontinuity-preserving model for estimating optical flow. *International Conference on Pattern Recognition*, pages 920–924, 1996.

[MP06] K. Moon and V. Pavlovic. Impact of dynamics on subspace embedding and tracking of sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 198–205, 2006.

[MR98] R. Mukundan and K. Ramakrishnan. *Moment Functions in Image Analysis: Theory and Application*. World Scientific Publishing, 1998.

[MS89] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[MS02] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142, 2002.

[MS03] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 257–263, 2003.

[MS04] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[MTHC01] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Articulated body posture estimation from multi-camera voxel data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 455–460, 2001.

[MTHC03] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3):199–223, 2003.

[MTLT88] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics Interface*, pages 26–33, 1988.

[MTS$^+$05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1):43–72, 2005.

[Muy57] E. Muybridge. *Animals in Motion*. Dover Publications, New York, 1957.

[OB80] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):522–536, 1980.

[OMG07] Oxford metrics group. annual report. http://www.omg3d.com/html/documents/OMG_2007AR.pdf, 2007.

[Org08] Organic motion: Stage. http://www.organicmotion.com, March 2008.

[PA98] X. Pennec and N. Ayache. Uniform distribution, distance and expectation problems for geometric features processing. *Journal of Mathematical Imaging and Vision*, 9(1):49–67, 1998.

[Par62] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[PBB$^+$06] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, 2006.

[PBRT99] J. Puzicha, J. Buhmann, Y. Rubner, and C. Tomasi. Empirical evaluation of dissimilarity measures for color and texture. In *International Conference on Computer Vision*, pages 1165–1172, 1999.

[PD02] N. Paragios and R. Deriche. Geodesic active regions: a new paradigm to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, 13(1/2):249–268, 2002.

[Pen98] X. Pennec. Computing the mean of geometric features: Application to the mean rotation. Technical Report RR–3371, INRIA, Sophia Antipolis, France, March 1998.

[Pen06] X. Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 2006.

[PF03] R. Plankers and P. Fua. Articulated soft objects for multiview shape and motion capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1182–1187, 2003.

[PH91] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):730–742, 1991.

[Pop07] R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1-2):4–18, 2007.

[PS99] M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

[Qua08] Qualisys: Oqus and proreflex. http://www.qualisys.com, March 2008.

[Ras80] R. Rashid. Toward a system for the interpretation of moving light display. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):574–581, 1980.

[RBCS06] B. Rosenhahn, T. Brox, D. Cremers, and H.-P. Seidel. A comparison of shape matching methods for contour based pose estimation. In *Combinatorial Image Analysis*, volume 4040 of *LNCS*, pages 263–276. Springer, 2006.

[RBN$^+$97] C. Reinschmidt, A. Van Den Bogert, B. Nigg, A. Lundberg, and N. Murphy. Effect of skin movement on the analysis of skeletal knee joint motion during running. *Journal of Biomechanics*, 30(7):729–732, 1997.

[RBS$^+$06] B. Rosenhahn, T. Brox, D. Smith, J. Gurney, and R. Klette. A system for markerless human motion estimation. *Künstliche Intelligenz*, 1:45–51, 2006.

[RBS07] B. Rosenhahn, T. Brox, and H.-P. Seidel. Scaled motion dynamics for markerless motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[RBW07] B. Rosenhahn, T. Brox, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose tracking. *International Journal of Computer Vision*, 73(3):243–262, 2007.

[RFDC05] L. Reveret, L. Favreau, C. Depraz, and M.-P. Cani. Morphable model of quadrupeds skeletons for animating 3d animals. In *Symposium on Computer Animation*, pages 135–142, 2005.

[RFZ07] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):65–81, 2007.

[RK94] J. Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *European Conference on Computer Vision*, pages 35–46, 1994.

[RKM08] B. Rosenhahn, R. Klette, and D. Metaxas, editors. *Human Motion – Understanding, Modelling, Capture and Animation*, volume 36 of *Computational Imaging and Vision*. Springer, Netherlands, 2008.

[RKP$^+$07] B. Rosenhahn, U. Kersting, K. Powell, R. Klette, G. Klette, and H.-P. Seidel. A system for articulated tracking incorporating a clothing model. *Machine Vision and Applications*, 18(1):25–40, 2007.

[Roh97] K. Rohr. *Human Movement Analysis Based on Explicit Motion Models (in Motion-Based Recognition by Shah, M. and Jain, R. (Eds.))*, chapter 8, pages 171–198. Kluwer Academic Publishers, Boston, 1997.

[Ros56] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837, 1956.

[RW01] L. Rogers and D. Williams. *Diffusions, Markov Processes and Martingales*, volume 1. Cambridge University Press, Cambridge, 2 edition, 2001.

[SB03] H. Sidenbladh and M. Black. Learning the statistics of people in images and video. *International Journal of Computer Vision*, 54(1-3):183–209, 2003.

[SB06] L. Sigal and M. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University, 2006.

[SBF00] H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, pages 702–718, 2000.

[SBR$^+$04] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard. Tracking loose-limbed people. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 421–428, 2004.

[SBS02] H. Sidenbladh, M. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *European Conference on Computer Vision*, volume 1, pages 784–800, 2002.

[Sch99] N. Schraudolph. Local gain adaptation in stochastic gradient descent. In *International Conference on Artificial Neural Networks*, pages 569–574, 1999.

[SE99] S. Sangwine and T. Ell. Hypercomplex auto- and cross-correlation of color images. In *IEEE International Conference on Image Processing*, pages 319–322, 1999.

[SFPG07] S. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc. Feature tracking and matching in video using programmable graphics hardware. *Machine Vision and Applications*, 2007.

[SH87] H. Szu and R. Hartley. Fast simulated annealing. *Physics Letters A*, 122:157–162, 1987.

[SH03] J. Starck and A. Hilton. Model-based multiple view reconstruction of people. In *International Conference on Computer Vision*, pages 915– 922, 2003.

[She98] F. Shevlin. Analysis of orientation problems using plucker lines. In *International Conference on Pattern Recognition*, pages 685–689, 1998.

[Sil86] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

[Sim08] Simi reality motion systems: Simi motion. http://www.simi.com, March 2008.

[SKM07] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Bme : Discriminative density propagation for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):2030–2044, 2007.

[SM99] P. Sturm and S. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1432–1437, 1999.

[ST94] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[ST03] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 22(6):371–391, 2003.

[Sto91] J. Stolfi. *Oriented Projective Geometry: A Framework for Geometric Computation*. Academic Press, Boston, 1991.

[SVD03] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *International Conference on Computer Vision*, pages 750–757, 2003.

[TCMS04] 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(6):333–351, 2004.

[TDDS06] L. Taycher, D. Demirdjian, T. Darrell, and G. Shakhnarovich. Conditional random people: Tracking humans with crfs and grid filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–229, 2006.

[THWS08] T. Thormählen, N. Hasler, M. Wand, and H.-P. Seidel. Merging of unconnected feature tracks for robust camera motion estimation from video. In *European Conference on Visual Media Production*, 2008.

[TLF08] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[TS96] C. Tsallis and D. Stariolo. Generalized simulated annealing. *Physica A*, 233:395–406, 1996.

[Tsa87] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

[UF04] R. Urtasun and P. Fua. 3d human body tracking using deterministic temporal motion models. In *European Conference on Computer Vision*, pages 92–106, 2004.

[UFF06] R. Urtasun, D. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 238–245, 2006.

[VBMP08] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics*, 27(3):1–9, 2008.

[Vic08] Oxford metrics group: Vicon mx and vicon motus video. http://www.vicon.com, March 2008.

[Vit08] Vitus smart 3d-bodyscanner. http://www.vitronic.de, March 2008.

[VLF04a] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *International Symposium on Mixed and Augmented Reality*, pages 48–57, 2004.

[VLF04b] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1391–1391, 2004.

[VZBH08] K. Varanasi, A. Zaharescu, E. Boyer, and R. Horaud. Temporal surface tracking using mesh evolution. In *European Conference on Computer Vision*, pages 30–43, 2008.

[WADP97] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, 1997.

[WN99] S. Wachter and H.-H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3):174–192, 1999.

[WR96] C. Williams and C. Rasmussen. Gaussian processes for regression. In *Advances in Neural Information Processing Systems*, 1996.

[WR06] P. Wang and J. Rehg. A modular approach to the analysis and evaluation of particle filters for figure tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 790–797, 2006.

[WRV98] J. Weickert, B. Ter Haar Romeny, and M Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7:398–410, 1998.

[WSLG07] O. Weber, O. Sorkine, Y. Lipman, and C. Gotsman. Context-aware skeletal shape deformation. *Computer Graphics Forum*, 26(3), 2007.

[WW92] W. Weber and E. Weber. *Mechanics of the Human Walking Apparatus*. Springer, Berlin, 1992. Translated by R. Furlong and P. Maquet. Originally published 1836: *Über die Mechanik der menschlichen Gehwerkzeuge*.

[XL07] X. Xu and B. Li. Learning motion correlation for tracking articulated human body with a rao-blackwellised particle filter. In *International Conference on Computer Vision*, pages 1–8, 2007.

[YBS07] S. Yoshizawa, A. Belyaev, and H.-P. Seidel. Skeleton-based variational mesh deformations. *Computer Graphics Forum*, 26(3):255–264, 2007.

[YK91] M. Yamamoto and K. Koshikawa. Human motion analysis based on a robot arm model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 664–665, 1991.

[YWP06] H. Yang, G. Welch, and W. Pollefeys. Illumination insensitive model-based 3d object tracking and texture refinement. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 869–876, 2006.

[Zha94] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[Zha99] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *International Conference on Computer Vision*, pages 666–673, 1999.

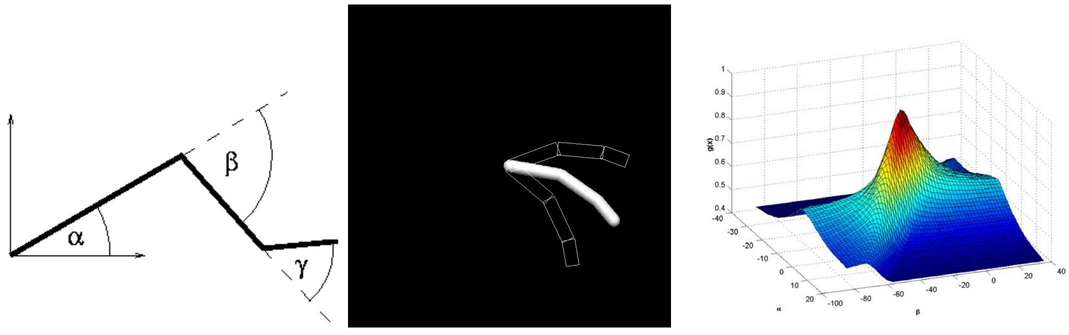
[ZRS05] R. Zayer, C. Rössel, and H.-P. Seidel. Discrete tensorial quasi-harmonic maps. In *International Conference on Shape Modeling and Applications*, pages 276–285, 2005.

A

Appendix

A.1 Parameter Evaluation for ISA by means of Synthesized Sequences

For an exhaustive experimental evaluation of the parameters of ISA, an articulated arm with three degrees of freedom is tracked. The aim of this section is not to find “the best” parameters since these depend on the specific application. Rather, we reveal the general impact of the parameters on the performance using an experimental setting that is typical for human motion capture. The evaluation results provide a general guidance and a good starting point for finding the optimal setting for a particular application. Parameter settings for full-body human pose estimation are given in Sections 6.2 and 6.3. Furthermore, we compare the two selection kernels discussed in Section 6.1.5. ISA with $\epsilon_t = 0$ (6.7) is denoted by ISA_0 and with $\epsilon_t(\eta_t) = 1/(n\,\langle\eta_t, \exp(-\beta_t\,V)\rangle)$ (6.8) by $ISA_{1/n}$. In Section A.1.2, we demonstrate the influence of the mixing condition that is essential for the convergence of ISA as shown in Theorem 6.1.2 and Theorem 6.1.3.



(a) (b) (c)

Figure A.1: **From left to right: a)** The pose of the arm is described by the vector $x = (\alpha, \beta, \gamma)^T$. **b)** Image of the synthesized arm with superimposed templates for two different values of α. **c)** Graph of $\exp(-V(x))$ over (α, β).

A.1.1 Toy Example

Experimental Set-up and Implementation Details The arm consists of three limbs and three joints. The position of the arm is described by $x^T = (\alpha, \beta, \gamma) \in E$, where

$E := [-170, 170] \times [-125, 125] \times [-125, 125]$ as depicted in Figure A.1 a). For evaluation, a sequence of 201 synthetic images is generated, see Figure A.1 b). X_0 is uniformly distributed in E yielding an unknown arm position at the beginning. The angles α_{t+1}, β_{t+1}, and γ_{t+1} are sampled from Gaussian distributions on E with mean α_t, β_t, and γ_t and variance $\sigma_\alpha = 20$, $\sigma_\beta = 40$, and $\sigma_\gamma = 30$, respectively. This sequence *(Seq_1)* is difficult for tracking since the velocity and the direction of the movement may change from frame to frame. In a second sequence *(Seq_2)*, the arm moves from position $(-30, -80, -40)^T$ to $(50, 30, 20)^T$ and back with constant speed as illustrated in Figure A.2 a). Moreover, we added some Gaussian noise to each position vector.

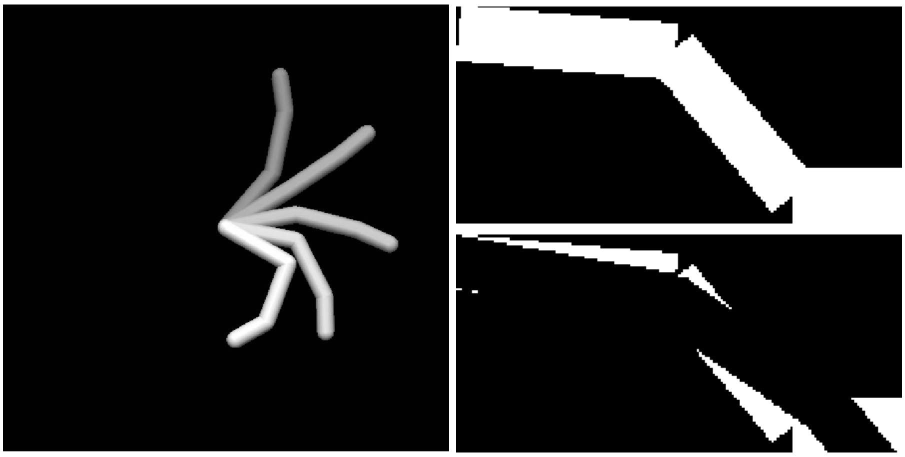

(a) (b)

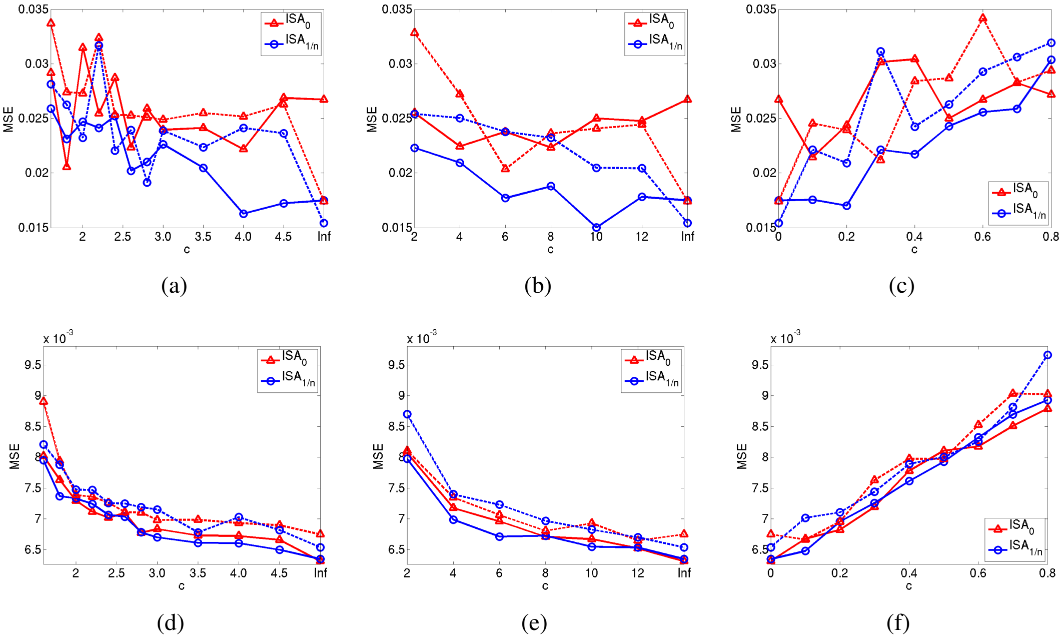
Figure A.2: **a)** Motion sequence Seq_2. **b)** Template's silhouette (*top*). Error map (*bottom*).

In order to compute the Boltzmann-Gibbs measures $\exp(-\beta_t V)$, the image is converted to a binary image by thresholding. This image is compared with the silhouette of each arm template that is determined by a particle $x_t^{(i)}$ as shown in Figure A.2 b). An error map is obtained by a pixelwise AND operation between the inverted binary image and the template's silhouette. The weighting functions are then calculated by $g_t := \exp(-\beta_t N_e / N_p)$, where N_p denotes the number of pixels of the template's silhouette and N_e the sum of the pixel values in the error map. The graph of the weighting function is plotted in Figure A.1 c). We have observed in our experiments that $\sup_t(\exp(\beta_t \ \mathrm{osc}(V)) \approx 40$. This means that the selection kernel (6.8) is valid if the number of particles is greater than 40.

In the following, we evaluate the performance of ISA_0 and $ISA_{1/n}$ in combination with different annealing schemes, variance schemes, number of annealing runs, and number of particles. The simulations for Seq_1 and Seq_2 have been repeated 50 and 40 times, respectively. The error of an estimate $\sum_i \pi_t^{(i)} x_t^{(i)}$ is measured by $1 - \exp(-N_e/N_p)$. The averages of the mean square errors (MSE) for each sequence indicate the performance.

Since in real world applications the measurements are noisy due to clutter, film grain, bad lighting conditions, CCD camera noise, etc., we have also added strong noise to the weighting functions by $\exp(-\beta_t \vartheta(N_e + W_t^{(i)})/N_p)$, where $\vartheta(N) = \max(0, \min(N, N_p))$ and $W_t^{(i)}$ are independent zero-mean Gaussian random variables with variance 40000. For comparison, $N_p \approx 4000$.

GPF vs. ISA We assume that the dynamics for Seq_1 are known. Hence, the algorithms are initialized by the uniform distribution on E and the prediction step (6.10) is performed according to the Gaussian transitions used for the arm simulation. By contrast, the dynamical model is not used for tracking Seq_2. The initial distribution is instead the uniform distribution on $[-20, -40] \times [-60, -100] \times [-20, -60] \subset E$ and the transitions kernels are the same as for Seq_1. In order to provide a fair comparison between a generic particle filter (GPF, Algorithm 1) with n_T particles and ISA with various annealing schemes, the number of particles is given by $n = \lfloor n_T/T \rfloor$ where T denotes the number of annealing runs. GPF with $n_T = 250$ achieves a MSE of $\mathbf{0.04386}$ for Seq_1 and $\mathbf{0.04481}$ for the noisy sequence. Seq_2 has been tracked with 225 particles and MSE of $\mathbf{0.01099}$ and $\mathbf{0.01157}$, respectively.



(a) (b) (c)

(d) (e) (f)

Figure A.3: Performance for different annealing schemes with $T = 5$. Average of the MSE for the sequences Seq_1 **(top)** and Seq_2 **(bottom)** with noisy measurements *(dashed)* and without noise *(solid)*. **a, d)** $\beta_t = \alpha\,(1 - c^{-(t+1)})$. **b, e)** $\beta_t = \alpha\,\ln(t + c)/\ln(T + c - 1)$. **c, f)** $\beta_t = \alpha\,((t + 1)/T)^c$. **Top:** The curves for the geometric annealing schemes are unstable and the best result is obtained by $ISA_{1/n}$ with a logarithmic scheme. **Bottom:** The error decreases when $\beta_t \to \alpha$. The impact of the selection kernel and noise is small.

Annealing Schemes We have evaluated the performance of various annealing schemes $0 \leq \beta_0 \leq \cdots \leq \beta_{T-1}$ with fixed length $T = 5$. While the particles are diffused between the annealing steps for Seq_1 by Gaussian kernels with $\sigma_\alpha = 20$, $\sigma_\beta = 40$, and $\sigma_\gamma = 30$, we set $\sigma_\alpha = \sigma_\beta = \sigma_\gamma = 5$ for Seq_2. In Figure A.3, the MSEs for the annealing schemes with decreasing increments

$$
\begin{aligned}
\beta_t &= \alpha\,(1 - c^{-(t+1)}) && (\textit{geometric}),\\
\beta_t &= \alpha\,\ln(t + c)/\ln(T + c - 1) && (\textit{logarithmic}),\\
\beta_t &= \alpha\,((t + 1)/T)^c && (\textit{polynomial})
\end{aligned}
$$

β_t	ISA_0	$ISA_{1/n}$	ISA_0	$ISA_{1/n}$
	Seq_1		Seq_1 with noise	
$\alpha\,(t+1)/T$	0.03634	0.03029	0.03220	0.02809
$\alpha\,1.2^{t+1-T}$	**0.02819**	**0.02302**	**0.03185**	**0.02609**
$\alpha\,1.8^{t+1-T}$	0.04214	0.05128	0.03891	0.04452
	Seq_2		Seq_2 with noise	
$\alpha\,(t+1)/T$	0.01006	0.00948	0.00988	0.01026
$\alpha\,1.2^{(t+1-T)}$	**0.00818**	**0.00805**	**0.00827**	**0.00858**
$\alpha\,1.8^{(t+1-T)}$	0.01514	0.01501	0.01557	0.01543

Table A.1: MSE error for annealing schemes with constant and increasing increments ($T = 5$). The schemes are outperformed by the annealing schemes given in Figure A.3.

are given. The schemes are normalized such that $\beta_{T-1} = \alpha = 4$. When c tends to infinity or to 0, in the case of a polynomial scheme, $\beta_t \rightarrow \alpha$ for all $0 \leq t < T$.

The diagrams show that the geometric annealing schemes are unstable in the sense that the curves of the MSE with respect to c contain many local optima, particularly for Seq_1. It makes the optimization of the scheme for a particular application quite difficult. The logarithmic schemes performed best where the lowest MSE for Seq_1, namely 0.01501, was achieved by $ISA_{1/n}$ with $c = 10$. In comparison, the errors for Seq_2 are significant lower and the scheme with $\beta_t = \alpha$ performs best since the motion is simple and local maxima rarely occur. Furthermore, the difference between the two selection kernels is small. The impact of noise on the results is also minor when the dynamics are simple in contrast to the more difficult sequence. The observation that the error for Seq_1 with noise significantly declines as c goes to infinity indicates that the other parameters are not well chosen for this noisy sequence. The results for schemes with constant or increasing increments in Table A.1 reveal that these schemes are outperformed by the schemes given in Figure A.3. We use henceforth a polynomial annealing scheme with $c = 0.1$ since both ISA_0 and $ISA_{1/n}$ perform well for the scheme.

Variance Schemes During the mutation step of ISA, the particles are diffused according to a Gaussian distribution where the variance for each annealing step is defined by a variance scheme. The errors for constant schemes are given in Table A.2, for deterministic schemes in Tables A.3 and A.4, and for dynamic schemes (6.14) in Figure A.4. The first column of Tables A.3 and A.4 contains the reference variance that is reduced for each annealing step by the decreasing scheme given in the second column. We give three examples where $\iota \in \{\alpha, \beta, \gamma\}$: $(-d_\alpha\ -d_\beta\ -d_\gamma)$ means that $\sigma^2_{\iota,t} = \sigma^2_{\iota,t-1} - d_\iota$. The decreasing scheme $-0\,d^1\,d^2\,d^3$ gives the variance scheme $\sigma^2_{\iota,t} = \sigma^2_{\iota,t-1} - d^t$. The scheme $\sigma^2_{\iota,t} = d^{t+1}\sigma^2_\iota$ is denoted by $\times d^1\,d^2\,d^3\,d^4$.

The dynamic variance schemes are not only easier to handle since they depend only on one parameter c, but they also outperform the deterministic schemes provided that an appropriate parameter c is chosen. The best result for Seq_1 with MSE 0.01175 was obtained by $ISA_{1/n}$ with parameter $c = 0.3$. In comparison to the GPF, the MSE was reduced by more than 73%. We see that the error for Seq_2 was not significantly improved when comparing the best settings for constant, deterministic, and dynamic schemes. It indicates that the flow of Feynman-Kac distributions locates the global minimum and that the error is mainly caused by the particle

$(\sigma_\alpha^2\,\sigma_\beta^2\,\sigma_\gamma^2)$	ISA_0	$ISA_{1/n}$	ISA_0	$ISA_{1/n}$
	Seq_1		Seq_1 with noise	
$(15\,35\,25)$	0.02527	0.01985	0.02787	0.02573
$(20\,40\,30)$	**0.02145**	**0.01756**	0.02453	**0.02213**
$(25\,45\,35)$	0.02341	0.02011	0.02506	0.02357
$(15\,40\,35)$	0.02238	0.01891	**0.02035**	0.02510
$(25\,40\,25)$	0.02240	0.01905	0.02622	0.02345
	Seq_2		Seq_2 with noise	
$(0.5\,0.5\,0.5)$	0.00637	0.00631	0.00643	0.00664
$(2\,2\,2)$	0.00612	0.00627	**0.00639**	0.00652
$(5\,5\,5)$	0.00668	0.00648	0.00666	0.00702
$(0.5\,2\,5)$	**0.00611**	**0.00626**	0.00643	**0.00629**
$(5\,2\,0.5)$	0.00661	0.00674	0.00674	0.00695

Table A.2: MSE error for constant variance schemes. The decreasing schemes perform better (Tables A.3 and A.4).

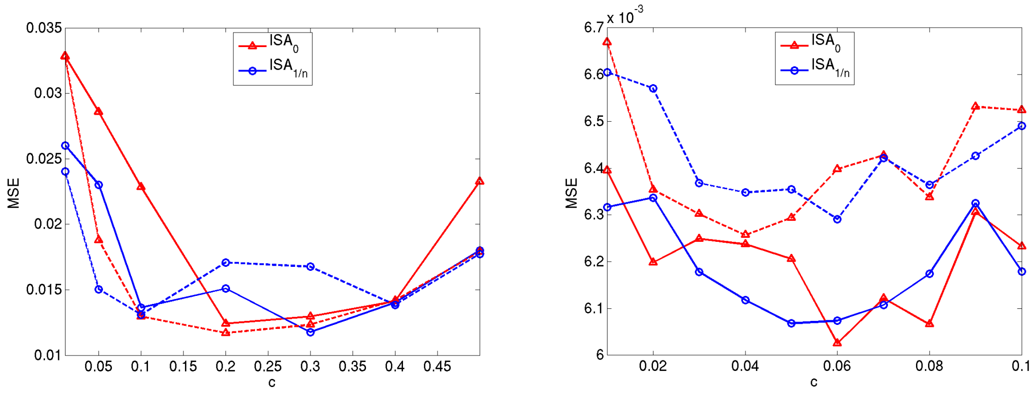
$(\sigma_\alpha^2\,\sigma_\beta^2\,\sigma_\gamma^2)$	Decreasing scheme	ISA_0	$ISA_{1/n}$	ISA_0	$ISA_{1/n}$
		Seq_1		Seq_1 with noise	
$(32\,49\,36)$	$(-4\;-3\;-2)$	0.01997	0.01920	0.02437	0.02335
$(32\,58\,54)$	$(-4\;-6\;-8)$	0.02243	0.02485	0.02480	0.02093
$(32\,70\,54)$	$(-4\;-10\;-8)$	0.02048	0.02066	0.02332	0.02411
$(32\,52\,42)$	$(-4\;-4\;-4)$	0.02193	0.01919	0.02489	**0.01795**
$(29\,52\,45)$	$(-3\;-4\;-5)$	0.01989	**0.01666**	0.02029	0.02074
$(23\,47\,35)$	$\times\frac{\beta_3}{\alpha}\,\frac{\beta_2}{\alpha}\,\frac{\beta_1}{\alpha}\,\frac{\beta_0}{\alpha}$	0.02230	0.01950	0.02654	0.02203
$(27\,47\,37)$	$-0\,1.5\,1.5^2\,1.5^3$	0.02187	0.02324	**0.01807**	0.02328
$(27\,47\,37)$	$-0\,1.5^3\,1.5^2\,1.5$	0.02048	0.02219	0.02398	0.02109
$(48\,97\,73)$	$\times 0.8\,0.8^2\,0.8^3\,0.8^4$	0.02140	0.02030	0.02099	0.02326
$(30\,60\,45)$	$\times 0.9\,0.9^2\,0.9^3\,0.9^4$	**0.01907**	0.01690	0.02470	0.02142

Table A.3: MSE error for deterministic variance schemes. The schemes are outperformed by dynamic variance schemes (Figure A.4).

$(\sigma_\alpha^2\,\sigma_\beta^2\,\sigma_\gamma^2)$	Decreasing scheme	ISA_0	$ISA_{1/n}$	ISA_0	$ISA_{1/n}$
		Seq_2		Seq_2 with noise	
$(3.5\,5\,8)$	$(-1\ -1\ -1)$	0.00619	0.00632	**0.00635**	**0.00629**
$(5\,5\,5)$	$(-1.5\ -1.5\ -1.5)$	0.00614	0.00623	0.00640	0.00656
$(3.5\,5\,6.5)$	$(-1\ -1.5\ -2)$	**0.00606**	0.00626	0.00641	0.00642
$(6.5\,5\,3.5)$	$(-2\ -1.5\ -1)$	0.00648	0.00654	0.00651	0.00656
$(7.5\,7.5\,7.5)$	$-0\,1.5\,1.5^2\,1.5^3$	0.00649	0.00657	0.00662	0.00662
$(7.5\,7.5\,7.5)$	$-0\,1.5^3\,1.5^2\,1.5$	0.00636	0.00638	0.00646	0.00657
$(1.2\,1.2\,1.2)$	$\times 0.8\,0.8^2\,0.8^3\,0.8^4$	0.00622	0.00623	0.00649	0.00639
$(.75\,.75\,.75)$	$\times 0.9\,0.9^2\,0.9^3\,0.9^4$	0.00631	**0.00607**	0.00636	0.00641

Table A.4: MSE error for deterministic variance schemes. The best dynamic variance schemes (Figure A.4) perform as well as the best deterministic variance schemes.

approximation. Hence, an improvement is only expected by reducing the number of annealing runs yielding more particles for approximation or by increasing n_T.

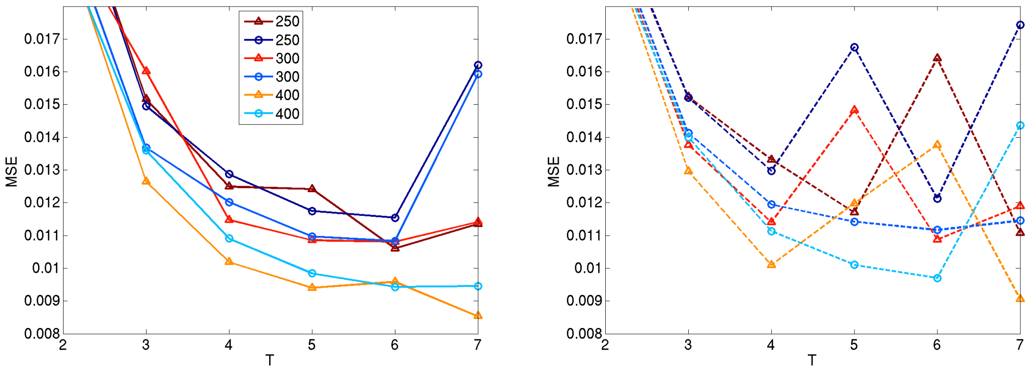


(a) (b)

Figure A.4: Performance for dynamic variance schemes with different values of c in the presence of noise *(dashed)* and without noise *(solid)*. **From left to right: a)** MSE for Seq_1. The error is significantly reduced in comparison to deterministic schemes (Tables A.2 and A.3). The best result is obtained by $ISA_{1/n}$ with $c = 0.3$. **b)** MSE for Seq_2. The best dynamic variance schemes perform as well as the best deterministic variance schemes (Tables A.2 and A.4).

Number of Annealing Runs and Particles The influence of the number of annealing runs for different values of n_T is plotted in Figures A.5 and A.6. Seq_1 has been tracked by ISA_0 and $ISA_{1/n}$ with a dynamic scheme with $c = 0.2$ and $c = 0.3$, respectively. The parameters for Seq_2 are 0.06 and 0.05, respectively. The curves for $ISA_{1/n}$ are quite stable with a unique optimal parameter $T = 6$ independent of n_T and noise, see Figure A.5. By contrast, the curves for ISA_0 contain deep local minima, in particular when the sequence was disturbed by noise.

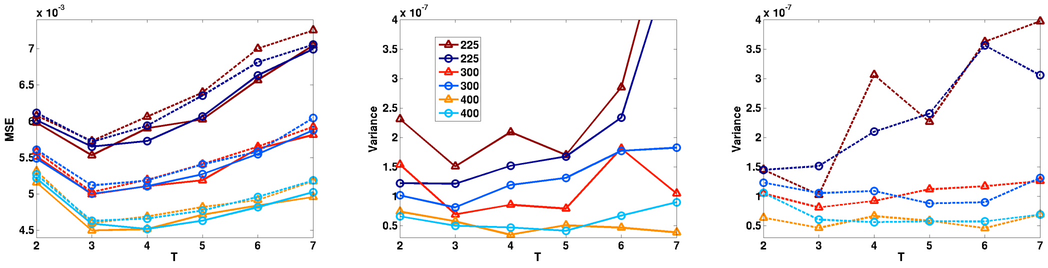
Moreover, one can observe at $T = 7$ that the error for $ISA_{1/n}$ increases significantly when the number of particles is not clearly greater than $\sup_t(\exp(\beta_t \operatorname{osc}(V))$. This shows the impact of the condition on the results. The MSEs for Seq_2 are given in Figure A.6. The error is reduced by decreasing the number of annealing runs and by increasing n_T as expected whereas the differences between ISA_0 and $ISA_{1/n}$ are minimal. It also demonstrates the robustness of ISA to noise. As comparison, the error of GPF was hardly reduced by increasing n_T. The MSE was still above 0.043 and 0.01 for Seq_1 and Seq_2, respectively.



(a)

(b)

Figure A.5: Performance of ISA_0 *(triangles)* and $ISA_{1/n}$ *(circles)* for different numbers of annealing runs T with $n_T = 250$, 300, and 400. The curves for $ISA_{1/n}$ are more stable with a unique optimal parameter $T = 6$, but the error increases at $T = 7$. More annealing runs are required than for Seq_2 (Figure A.6). **From left to right: a)** MSE for Seq_1 without noise. **b)** MSE for Seq_1 with noise.



(a)

(b)

(c)

Figure A.6: Performance of ISA_0 *(triangles)* and $ISA_{1/n}$ *(circles)* for different numbers of annealing runs T with $n_T = 225$, 300, and 400. **From left to right: a)** MSE for Seq_2 with noisy measurements *(dashed)* and without noise *(solid)*. The error decreases with increasing n_T whereas the differences between ISA_0 and $ISA_{1/n}$ are minimal. The error is only slightly affected by noise. **b)** Variance of MSE for Seq_2 without noise. The variance also decreases with increasing n_T. The curves for $ISA_{1/n}$ are more stable. **c)** Variance with noise.

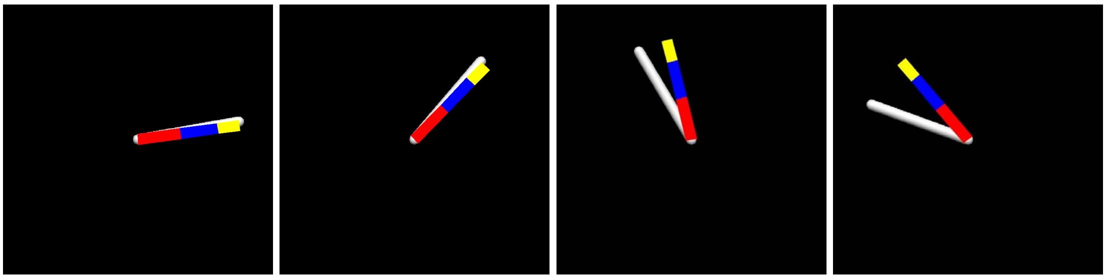


Figure A.7: When the mixing condition is not satisfied, ISA loses track of the articulated arm after some time and is not able to recover. **From left to right:** $t = 1$, 5, 158 and 165.

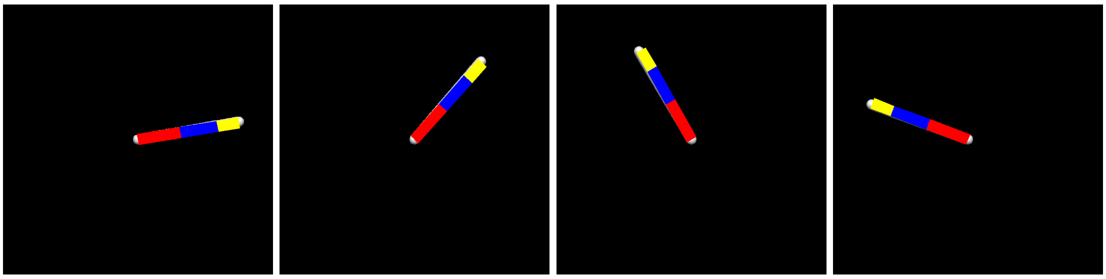


Figure A.8: When the mixing condition is satisfied, ISA is able to track the articulated arm. **From left to right:** $t = 1$, 5, 158, and 165.

A.1.2 Mixing Condition

In this section, we illustrate the impact of the mixing condition that is essential for the convergence results given in Sections 3.4.3 and 6.1.5. For this purpose, we track a stiff arm, i.e. $x = \alpha$. We suppose that the arm movement is given by the process $X_t := X_{t-1} + V_t$, where $X_0 := 0$ and V_t are i.i.d. uniform random variables on $[-10, 10]$. Let us examine the events where $V_t \in [9.75, 10]$ for $1 \leq t \leq 400$. Even though the probability that this occurs is very small, it is strictly greater than zero.

For the simulations, we have used ISA_0 with a prediction step (6.10) and parameters $n = 100$, $T = 2$, $\beta_0 = 3.2$. The initial distribution is δ_0 and the mutation kernels $K_t(x, \cdot)$ are uniform distributions on $[x - 2, x + 2]$. When uniform kernels have been chosen for prediction in accordance with the process X_t, ISA has not been capable of tracking the articulated arm as shown in Figure A.7. The algorithm loses track of the arm after some time and is not able to recover afterwards. For comparison, the uniform kernels have been replaced by Gaussian kernels with variance 100, which satisfy the mixing condition since the state space is bounded. In this case, the arm has been successfully tracked over a sequence of 400 images, see Figure A.8. We carried out the simulations 25 times. This shows that interacting particle systems may fail when the mixing condition is not satisfied, even though the particles are correctly predicted according to the dynamics.

B

Publications

The work presented in this thesis was published in the following papers.

[1] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. *Learning for multi-view 3d tracking in the context of particle filters.* In International Symposium on Visual Computing, volume 4292 of LNCS, pages 59–69. Springer, 2006.

[2] J. Gall, B. Rosenhahn, and H.-P. Seidel. *Robust pose estimation with 3d textured models.* In IEEE Pacific-Rim Symposium on Image and Video Technology, volume 4319 of LNCS, pages 84–95. Springer, 2006.

[3] J. Gall, J. Potthoff, C. Schnörr, B. Rosenhahn, and H.-P. Seidel. *Interacting and annealing particle filters: Mathematics and a recipe for applications.* Journal of Mathematical Imaging and Vision, 28(1):1–18, 2007.

[4] J. Gall, B. Rosenhahn, and H.-P. Seidel. *Clustered stochastic optimization for object recognition and pose estimation.* In Pattern Recognition, volume 4713 of LNCS, pages 32–41. Springer, 2007.

[5] S. Gehrig, H. Badino, and J. Gall. *Accurate and Model-Free Pose Estimation of Crash Test Dummies.* Human Motion - Understanding, Modeling, Capture and Animation, Computational Imaging and Vision, Springer, Vol 36, pages 453–473. Springer, 2008.

[6] J. Gall, B. Rosenhahn, and H.-P. Seidel. *An Introduction to Interacting Simulated Annealing.* Human Motion - Understanding, Modeling, Capture and Animation, Computational Imaging and Vision, Springer, Vol 36, pages 319–343. Springer, 2008.

[7] J. Gall, B. Rosenhahn, and H.-P. Seidel. *Drift-free tracking of rigid and articulated objects.* In IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[8] J. Gall, B. Rosenhahn, S. Gehrig, and H.-P. Seidel. *Model-based motion capture for crash test video analysis.* In Pattern Recognition, volume 5096 of LNCS, pages 92–101. Springer, 2008.

[9] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. *Optimization and filtering for human motion capture – a multi-layer framework.* International Journal of Computer Vision, 2008.

[10] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers. *Combined region- and motion-based 3d tracking of rigid and articulated objects.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009.

[11] N. Hasler, B. Rosenhahn, T. Thormählen, M. Wand, J. Gall, and H.-P. Seidel. *Markerless motion capture with unsynchronized moving cameras.* In IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[12] J. Gall, C. Stoll, E. De Aguiar, B. Rosenhahn, C. Theobalt, and H.-P. Seidel. *Motion capture using joint skeleton tracking and surface estimation.* In IEEE Conference on Computer Vision and Pattern Recognition, 2009.