Lehrstuhl für Bildverstehen und wissensbasierte Systeme
Institut für Informatik
Technische Universität München

# Observing and Interpreting Complex Human Activities in Everyday Environments

*Jan Bandouch*

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

| | |
|---|---|
| Vorsitzender: | Univ.-Prof. Dr. Daniel Cremers |
| Prüfer der Dissertation: | 1. Univ.-Prof. Michael Beetz, Ph.D. |
| | 2. Prof. Odest Chadwicke Jenkins, Ph.D. (Brown University, Providence, RI, USA) |

Die Dissertation wurde am 14.04.2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 28.10.2010 angenommen.

# Abstract

The ability to automatically observe and interpret human activities is one of the main challenges in computer vision research. Successful methods will provide the foundations for a broad range of novel and advanced applications in human computer interaction, robotics or biomechanical and clinical research, to name just a few. Still, after more than two decades of research, the task remains a hard challenge.

In this thesis we present a novel system for unintrusive observation of human activities that requires no more than three cameras to precisely estimate human fullbody motions in a wide variety of scenarios. It is capable to track a large spectrum of motions, including scenarios where the subject is partially occluded, where it manipulates objects as part of its activities, or where it interacts with the environment or other humans. The accuracy and robustness obtained by our system is the result of the following contributions. First, we take an anthropometric human model and optimize it towards use in a probabilistic tracking framework to provide a detailed biomechanical representation of human shape, posture and motion. Second, we introduce a sophisticated hierarchical sampling strategy for tracking that is embedded in a probabilistic framework and outperforms state-of-the-art Bayesian methods. Third, we show how to track complex manipulation activities in everyday environments using a combination of learned human appearance models and implicit environment models. Fourth, we introduce a method to learn environment- and task-specific models of human motion over time. These models do not only improve the predictive capabilities and thus the efficiency of our tracker, but also provide the basis for the recognition and interpretation of observed activities.

Our system competes with the state-of-the-art for markerless human motion capture, with a mean accuracy of about 2 cm for the joint locations. At the same time, our pose estimates are more detailed due to the increased level of realism for the spine and shoulders in our biomechanical model. In terms of robustness and generality, our system outperforms related methods as is shown through extensive evaluation on today's benchmarks and several challenging sequences ranging from athletic exercises to ergonomic case studies to everyday manipulation tasks. In particular, we have created the first publicly available data set that features markerless fullbody motion capture data of challenging manipulation activities for several subjects.

# Kurzfassung

Das automatische Beobachten und Erkennen von menschlichen Aktivitäten ist eine der größten Herausforderungen im Bereich des automatischen Bildverstehens. Erfolgreiche Lösungsansätze werden eines Tages die Grundlage für innovative Anwendungen *u.a.* im Bereich der Mensch-Maschine-Interaktion, der Robotik oder auch der biomechanischen Bewegungsanalyse bilden. Nach mehr als zwei Jahrzehnten Forschung bleibt das gesteckte Ziel jedoch noch immer eine große Herausforderung.

In dieser Dissertation stellen wir ein neuartiges System zur passiven Beobachtung menschlicher Aktivitäten vor. Es ermöglicht die präzise Vermessung menschlicher Ganzkörperbewegungen in einer Vielzahl unterschiedlicher Szenarien, wobei nicht mehr als drei Kameras benötigt werden. Das System arbeitet zuverlässig mit einem breiten Spektrum an Bewegungen, selbst wenn die zu beobachtende Person teilweise verdeckt wird, wenn sie mit Gegenständen hantiert, oder wenn sie mit der Umgebung oder anderen Personen interagiert. Die Genauigkeit und Robustheit unseres Systems ist auf nachstehende Beiträge dieser Arbeit zurückzuführen. Zum einen verwenden wir ein anthropometrisches Menschmodell, welches wir im Hinblick auf die Verwendbarkeit in probabilistischen Verfahren zur Bewegungsverfolgung optimiert haben. Dies ermöglicht uns eine genaue biomechanische Modellierung des äußeren Erscheinungsbilds sowie der Haltung und Bewegung von Menschen. Des weiteren präsentieren wir ein fortschrittliches hierarchisches Verfahren zur Bayes'schen Bewegungsschätzung, welches verwandte Verfahren in Bezug auf Genauigkeit und Zuverlässigkeit übertrifft. Wir zeigen außerdem, wie man die Vermessung komplexer Manipulationsaktivitäten in Alltagsumgebungen durch das Erlernen menschlicher Erscheinungsbilder sowie durch implizite Umgebungsmodellierung bewerkstelligen kann. Darüber hinaus stellen wir eine automatisierte Methode vor, die das inkrementelle Erlernen umgebungsspezifischer Modelle für menschliche Bewegungen ermöglicht. Diese Modelle können nicht nur zur verbesserten Bewegungsvorhersage und somit zur Effizienzsteigerung unseres Systems verwendet werden, sondern liefern gleichzeitig die Basis für die Erkennung und Interpretation der beobachteten Aktivitäten.

Unser System gehört mit einer mittleren Genauigkeit der geschätzten Gelenkpositionen von ungefähr $2$ cm zu den genauesten Systemen zur passiven menschlichen Bewegungserfassung

ohne Marker. Gleichzeitig ist die Haltungsschätzung auf Grund der erhöhten Detaildarstellung unseres Menschmodells besonders im Bereich der Wirbelsäule und der Schultern um einiges detaillierter als bei vergleichbaren Systemen. In Bezug auf Robustheit und allgemeine Anwendbarkeit übertreffen unsere Methoden verwandte Ansätze, wie umfangreiche Versuchsreihen bestätigen. Zu den anspruchsvollen Testsequenzen gehören neben wissenschaftlichen Benchmarks auch Aufnahmen von Bodenturnen, ergonomischen Einstiegsstudien im Automobilbereich, sowie alltäglichen Küchenaktivitäten. Darüber hinaus haben wir im Rahmen unserer Arbeit das erste umfangreiche Datenset mit komplexen Bewegungsdaten von Alltagsaktivitäten veröffentlicht, welches nur mit Hilfe passiver Verfahren zur Bewegungsverfolgung erstellt wurde.

# Acknowledgements

I would like to thank all the great people at the Intelligent Autonomous Systems group in Munich for making my PhD research so much fun despite the hard work. First and foremost I would like to thank my advisor Prof. Michael Beetz for giving me the opportunity to do research in Robotics and Computer Vision, for providing an excellent working environment and for all the personal and technical support throughout the years. Many thanks to Derik for introducing me to research (and to Michael). Cheers to Suat, Freek, Andreas, Radu, Nico v. H., Bernhard, Matthias, Andras, Alexis, Federico, Francisco, Moritz, Dominik, Thomas, Lorenz, Alex, Armin, Ingo, Lars, Dejan, Tetsuyou, Zahid, Murat, Zoltan, Nico B., Mihai, Uli, Christoph, Karinne and many more for endless discussions, evenings and 'kicker' sessions. Special thanks to Andreas, Dejan, Zoltan, Dominik, Moritz and Suat for reviewing parts of this thesis (sorry if I forgot anyone here). Thanks to all my students who showed me the joy (and sometimes pain) of teaching and who helped with many projects, and thanks to Sabine, Oliver and Quirin for shielding me from administrative tasks.

I would especially like to thank Prof. Chad Jenkins for inviting me to Brown University, for making me feel at home there, for introducing me to nonlinear dimensionality reduction and for serving on my committee. Thanks also to Aggeliki, Marek, Bart, Elisa, Silvia and all the other colleagues and friends from RLAB and Prof. Michael Black's group who made my stay in Providence a memorable experience.

I'm very grateful to Prof. Daniel Cremers for leading my thesis committee, and to all members of the scientific community for the many great insights and discussions at conferences and workshops. I would also like to thank Prof. Fernando De la Torre, Prof. Sebastian Thrun and Prof. Masayuki Inaba for inviting me to their labs and showing interest in my work.

Finally, I would like to thank my parents for early guidance and support, my love Monika for everything and much more, and myself for writing this thesis (despite the nice weather outside).

*To Moni*

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# List of Symbols

### Basic Notation

| | |
|---|---|
| $\mathbf{A}$ | Matrix (bold uppercase letters) |
| $\mathbf{a}$ | Vector (bold lowercase letters) |
| $a$ | Scalar (regular lowercase letters) |
| $\hat{\mathbf{a}}$ | Vector estimate |
| $\bar{\mathbf{a}}$ | Mean vector |
| $.^{\mathrm{T}}$ | Superscript denoting matrix/vector transpose |
| $\mathbf{A}^{-1}$ | Superscript denoting inverse of matrix $\mathbf{A}$ |
| $\{\dots\}$ | Listing of elements in a set |
| $(\dots)^{\mathrm{T}}$ | Listing of vector components |
| $\langle\dots\rangle$ | Listing of tuple components (ordered set) |
| $.^{(i)}$ | Superscript denoting the index of the $i$-th element in a set |
| $\emptyset$ | Empty set |
| $\mathcal{O}(n)$ | Landau notation for describing the time complexity of algorithms |

### Linear Algebra

| | |
|---|---|
| $\mathbb{R}^n$ | N-dimensional real coordinate space |
| $\mathbf{H}$ | Homogeneous Euclidean transformation matrix (6 d.o.f. pose in $\mathbb{R}^3$) |
| $\mathbf{p} = (x,\, y,\, z,\, 1)^{\mathrm{T}}$ | Homogeneous vector representation of a point (in $\mathbb{R}^3$) |
| $\mathbf{T}$ | Homogeneous translation matrix (in $\mathbb{R}^3$) |
| $\mathbf{t} = (tx,\, ty,\, tz)^{\mathrm{T}}$ | Translation parameter vector corresponding to $\mathbf{T}$ |
| $\mathbf{R}$ | Homogeneous (combined) rotation matrix (in $\mathbb{R}^3$) |
| $\mathbf{R}_x$ | Homogeneous rotation matrix around the $x$-axis only (in $\mathbb{R}^3$) |
| $\mathbf{r} = (rx,\, ry,\, rz)^{\mathrm{T}}$ | Euler rotation parameter vector corresponding to $\mathbf{R}$ |
| $\mathbf{P}$ | Homogeneous permutation/reflection matrix (in $\mathbb{R}^3$) |
| $\mathbf{I}$ | Identity matrix |
| $\mathbf{e}$ | Eigenvector |
| $\lambda$ | Eigenvalue |

| | |
|---|---|
| $\mathbf{U}$ | Matrix consisting of column Eigenvectors |
| $\mathbf{\Lambda}$ | Diagonal matrix consisting of Eigenvalues |

**Probabilistic State Estimation**

| | |
|---|---|
| $p(\mathbf{a})$ | Probability density function for variable $\mathbf{a}$ |
| $p(\mathbf{a}, \mathbf{b})$ | Joint probability density function over variables $\mathbf{a}$ and $\mathbf{b}$ |
| $p\big(\mathbf{a} \,\big|\, \mathbf{b}\big)$ | Conditional probability density function for variable $\mathbf{a}$ given $\mathbf{b}$ |
| $\boldsymbol{\mu}$ | Mean of a distribution |
| $\boldsymbol{\sigma}^2$ | Variance of a distribution |
| $\mathbf{\Sigma}$ | Covariance (matrix) of a distribution |
| $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ | Multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$ |
| $\mathbf{x}_t$ | State vector at time $t$ |
| $\mathbf{y}_t$ | Observation vector at time $t$ |
| $\mathbf{x}_{0:t}$ | State sequence from time $0$ to $t$ |
| $\mathbf{y}_{0:t}$ | Observation sequence from time $0$ to $t$ |
| $\mathcal{X}$ | State space for $\mathbf{x}$ |
| $\mathcal{P}_k$ | $k$-th sub-partition of the state space $\mathcal{X}$ |
| $\mathcal{S}_t$ | Particle set at time $t$ |
| $\mathcal{S}_{t,m}$ | Particle set at time $t$ and iteration $m$ |
| $\langle \mathbf{x}_t^{(i)}, w_t^{(i)} \rangle$ | $i$-th particle from particle set $\mathcal{S}_t$ at time $t$ consisting of state estimate $\mathbf{x}_t^{(i)}$ and corresponding weight $w_t^{(i)}$ |
| $N$ | Particle count |
| $M$ | Iteration or layer count |
| $d$ | State dimensionality |
| $c_d$ | Effective particle count per dimension |
| $\mathcal{D}$ | Survival diagnostic |
| $e_s^{(i)}$ | Shape-based observation error for $i$-th particle |
| $e_a^{(i)}$ | Appearance-based observation error for $i$-th particle |
| $w_s^{(i)}$ | Shape-based particle weight for $i$-th particle |
| $w_a^{(i)}$ | Appearance-based particle weight for $i$-th particle |
| $\langle \alpha_M, \ldots, \alpha_1 \rangle$ | Diffusion variance scheme in the *Annealed Particle Filter* (APF) |
| $\langle \beta_M, \ldots, \beta_1 \rangle$ | Annealing scheme in APF |

**Kalman Filter**

| | |
|---|---|
| $\mathbf{A}$ | Kalman state transition matrix |

| | |
|---|---|
| $\mathbf{C}$ | Kalman measurement matrix |
| $\boldsymbol{\epsilon}$ | Kalman process noise vector |
| $\boldsymbol{\delta}$ | Kalman measurement noise vector |
| $\mathbf{Q}$ | Kalman state transition covariance matrix |
| $\mathbf{R}$ | Kalman measurement covariance matrix |
| $\mathbf{K}$ | Kalman gain matrix |
| $g$ | Extended Kalman Filter state transition function |
| $h$ | Extended Kalman Filter measurement function |
| $\mathbf{G}$ | Extended Kalman Filter state Jacobian matrix |
| $\mathbf{H}$ | Extended Kalman Filter measurement Jacobian matrix |

**Color-/Shape-Based Image Operations**

| | |
|---|---|
| $I_F$ | Binary image foreground region mask |
| $I_P$ | Binary model projection region mask |
| $I_B$ | Binary blocking layer region mask |
| $I_H$ | Binary human appearance-like region mask |
| $\mathrm{XOR}(I_a, I_b)$ | Bitwise logical XOR operator between two binary region masks |
| $\mathrm{AND}(I_a, I_b)$ | Bitwise logical AND operator between two binary region masks |
| $\mathrm{OR}(I_a, I_b)$ | Bitwise logical OR operator between two binary region masks |
| $\mathrm{DIFF}(I_a, I_b)$ | Bitwise logical difference operator between two binary region masks |
| $\mathrm{NOT}(I_a)$ | Bitwise logical negation operator on a binary region masks |
| $\mathrm{COUNT}(I_a)$ | Operator returning the sum of all pixels set in a binary region mask |
| $\mathcal{C}_{HLS}$ | Bitmask representation of color in $HLS$ colorspace |
| $\mathcal{D}_{HLS}$ | Bitmask representation of color distribution in $HLS$ colorspace |
| $C_I$ | Color bitmask representation of current image |
| $C_P$ | Color bitmask representation of rendered model projection |
| $\mathrm{SIMILAR}(C_a, C_b)$ | Operator returning a binary region mask consisting of all pixels that are similar in color between $C_a$ and $C_b$ |

**Human Model Related**

| | |
|---|---|
| $\mathtt{bp}$ | Human body part identifier (see list of abbreviations for valid body parts) |
| $\overleftarrow{\mathtt{bp}}$ | Hierarchical predecessor to body part $\mathtt{bp}$ |
| $\overrightarrow{\mathtt{bp}}$ | Hierarchical successor(s) to body part $\mathtt{bp}$ |
| $\mathbf{H}_{\mathtt{bp}}$ | Part coordinate system (PCS) of body part $\mathtt{bp}$ |
| $t$, $n$, $b$ | Axis directions (*tangential*, *normal*, *binormal*) in PCS |

| | |
|---|---|
| $\boldsymbol{\psi}$ | Pose parameter vector for human model |
| $\mathbf{r}'_{\mathtt{bp}}$ | Rotation parameter vector for body part $\mathtt{bp}$ |
| $\tilde{\mathbf{r}}_{\mathtt{bp}}$ | Base rotation vector for body part $\mathtt{bp}$ |
| $\mathbf{r}_{\mathtt{bp}}$ | Final rotation vector for body part $\mathtt{bp}$ |
| $\mathbf{rmin}_{\mathtt{bp}}$ | Minimal angular rotation limits for body part $\mathtt{bp}$ |
| $\mathbf{rmax}_{\mathtt{bp}}$ | Maximal angular rotation limits for body part $\mathtt{bp}$ |
| $\boldsymbol{\phi}$ | Shape parameter vector for human model (combined) |
| $\boldsymbol{\phi}_I$ | Inner shape parameter vector for human model |
| $\boldsymbol{\phi}_O$ | Outer shape parameter vector for human model |
| $l_i$ | Anthropometric inner length values for human model |
| $\mathbf{s}_i$ | Absolute scale parameters for the $i$-th outer slice in the human model |
| $\boldsymbol{\varphi}$ | Learned reduced shape parameter vector for human model (combined) |
| $\mathbf{vmax}_{\mathtt{bp}}$ | Maximal angular velocities for body part $\mathtt{bp}$ |
| $\mathbf{R}^{\star}_{\mathtt{bp}}$ | $3\times3$ rotation matrix corresponding to PCS of body part $\mathtt{bp}$ |
| $\mathbf{t}^{\star}_{\mathtt{bp}}$ | $3\times1$ translation vector corresponding to PCS of body part $\mathtt{bp}$ |
| $s$ | Size of temporal pose history for motion snippets |
| $\boldsymbol{\varsigma}^{(i)}$ | Motion snippet parameter vector encoding the $i$-th human pose and its short-term temporal pose history of size $s$ |
| $\boldsymbol{\rho}^{(i,j)}$ | Pose parameter vector encoding all joint locations of pose $i+j$ in coordinates relative to the pelvis PCS of pose $i$ |
| $\boldsymbol{\tau}^{(i,j)}_{\mathtt{bp}}$ | Joint location for body part $\mathtt{bp}$ of pose $i+j$ in coordinates relative to the pelvis PCS of pose $i$ |

# List of Abbreviations

| | |
|---|---|
| AIS | Annealed Importance Sampling |
| APF | Annealed Particle Filter |
| ATN | Adjacent Temporal Neighbor |
| AuxPF | Auxiliary Particle Filter |
| BIHS | Branched Iterative Hierarchical Sampling |
| CAD | Computer Aided Design |
| CCD | Charge-Coupled Device |
| CG | Computer Graphics |
| CRF | Conditional Random Field |
| CSD | Covariance Scaled Diffusion |
| CTN | Common Temporal Neighbor |
| CV | Computer Vision |
| d.o.f. | Degrees of freedom |
| EKF | Extended Kalman Filter |
| EM | Expectation-Maximization |
| f.p.s. | Frames per second |
| GPDM | Gaussian Process Dynamical Model |
| GPU | Graphics Processing Unit |
| GUI | Graphical User Interface |
| HCI | Human-Computer-Interaction |
| HMM | Hidden Markov Model |
| HLS | Hue-Luminance-Saturation (colorspace) |
| HOG | Histogram of Oriented Gradients |
| HRI | Human-Robot-Interaction |
| HSV | Hue-Saturation-Value (colorspace) |
| ICP | Iterative Closest Point |
| IHS | Iterative Hierarchical Sampling |
| Isomap | Isometric Feature Mapping |

| | |
|---|---|
| MCMC | Markov Chain Monte Carlo |
| MDS | Multidimensional Scaling |
| MAP | Maximum a Posteriori |
| Lab | Lab (colorspace) |
| PCA | Principal Component Analysis |
| PCS | Part Coordinate System |
| pdf | Probability density function |
| PS | Partitioned Sampling |
| RFID | Radio Frequency Identification |
| RGB | Red-Green-Blue (colorspace) |
| RLE | Run-Length-Encoding |
| SCS | Slice Coordinate System |
| SIFT | Scale-Invariant Feature Transform |
| SIR | Sampling Importance Resampling |
| ST-Isomap | Spatio-Temporal Isometric Feature Mapping |
| SVD | Singular Value Decomposition |
| SVM | Support Vector Machine |
| TUM | Technische Universität München |
| UKF | Unscented Kalman Filter |
| UPF | Unscented Particle Filter |
| VSD | Variance Scaled Diffusion |

## Human Body Part Identifiers (hierarchical order)

| | | | | | |
|---|---|---|---|---|---|
| `BEC` | Pelvis | `OSL` | Left Thigh | `UAL` | Left Forearm |
| `ULW` | Lower Lumbar Spine | `USL` | Left Lower Leg | `HAL` | Left Hand |
| `OLW` | Upper Lumbar Spine | `FUL` | Left Foot | `FIL` | Left Fingers |
| `UBW` | Lower Thoracic Spine | `FBL` | Left Foot Ball | `SBR` | Right Shoulder Plate |
| `OBW` | Upper Thoracic Spine | `OSR` | Right Thigh | `OAR` | Right Upper Arm |
| `UHW` | Lower Cervical Spine | `USR` | Right Lower Leg | `UAR` | Right Forearm |
| `BRK` | Chest | `FUR` | Right Foot | `HAR` | Right Hand |
| `OHW` | Upper Cervical Spine | `FBR` | Right Foot Ball | `FIR` | Right Fingers |
| `KO` | Head | `SBL` | Left Shoulder Plate | | |
| `SEH` | Viewing Ray | `OAL` | Left Upper Arm | | |

# CHAPTER 1

# Introduction

Observing and interpreting human activities is a constant topic of interest in *artificial intelligence* (AI) and *computer vision* (CV) research. The ability to understand human behavior and to act with respect to human actions or intentions is an ambitious goal. Once achieved, it will prove beneficial in a myriad of application areas ranging from *robotics* to *human computer interaction* (HCI) or even *medical research* such as *clinical gait analysis*. However, despite the broad progress made in the field in the last decades, the road ahead is still paved.

In this thesis we will present our contributions to this highly active field of research. We will present a markerless system for human fullbody motion tracking from three or more cameras that utilizes a realistic human model to estimate the observed motions at high accuracy. Our system is capable to extract this information for arbitrary types of motions in realistic environments. This includes manipulation activities where objects are being handled and where interactions with the environment or between humans take place. Furthermore, our system is self-adaptive in that it can improve its efficiency over time by learning environment specific motion patterns. Lastly, these motion patterns can be used to infer semantic labels for the observed activities, and thus to reason about human activities and intentions.

## 1.1 Aims and Motivation

The observation of human motion is an important step when reasoning about human activities. Depending on the requirements of the target application and on the activities one is interested in, human motion can be estimated at different levels of detail. For instance, when analyzing soccer games based on TV broadcasts [19, 22, 60], one will be interested in the 2D motions of all players on the field (Figure 1.1a). Based on this information and on knowledge of the soccer rules, it then becomes possible to analyze the current game play and to reason about important aspects of the game, such as the role of players or the overall strategies of

each team [24]. Surveillance scenarios [70] are another example where the observation of human motions is used to infer ongoing activities (Figure 1.1b). The main goal for related applications is to detect abnormal behavior and potential security threats. Once again, the 2D motion of humans on the ground plane can be a good indicator for their intentions, *e.g.* when a person trespasses into secured areas. In addition, other aspects of the human motion such as the viewing direction or the location of the hands might prove valuable when trying to detect suspicious behavior. However, detecting the hands or estimating the viewing direction increases the complexity of the computational problem. Finally, some applications require the detailed estimation of human fullbody postures that include the position of all relevant body joints (Figure 1.1c). This is one of the most comprehensive ways to describe human motion, and the corresponding estimation problem is among the most challenging and promising tasks in AI, CV and related disciplines. It is also the main topic addressed in this thesis. We believe that human fullbody motion data will eventually serve as the basis for many new applications in a wide range of application areas, some of which we will now briefly describe to motivate the work presented in this thesis.

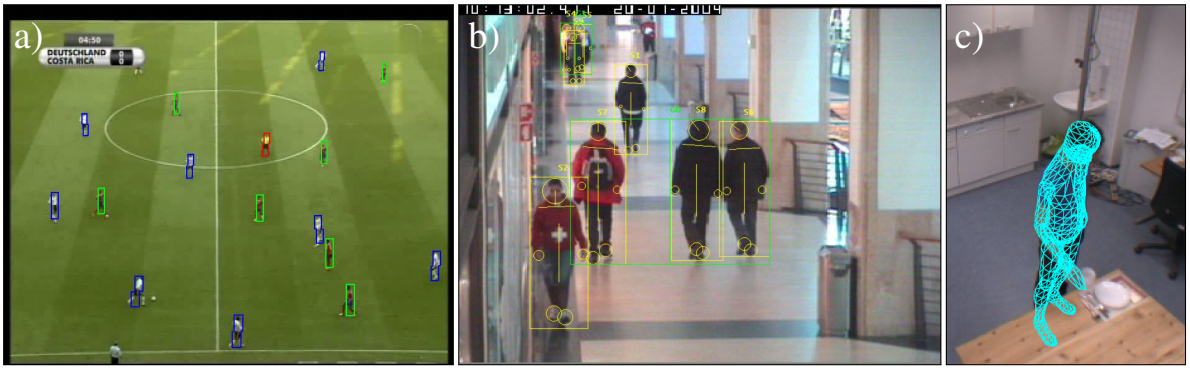


FIGURE 1.1 Different application scenarios for human observation: a) tracking of soccer players from TV broadcasts (image taken from our earlier work on sports analysis [19, 22, 60]) b) pedestrian detection for surveillance or safety applications (image with marked up ground-truth taken from the PETS 2004 CAVIAR data set [118]) c) fullbody tracking of human manipulation activities in living environments (as presented in this thesis). Notice the increasing level of body details that are being estimated from Figure a to c.

The unintrusive estimation of human fullbody motion as presented in this thesis has potential applications in many areas. Among the more obvious uses are novel input methods for HCI, where interactions between humans and their computers could be driven by gestures instead of traditional input methods such as keyboards or mice. The commercial potential of such applications is illustrated by the fact that MICROSOFT is currently developing a closely related system that should use human fullbody motions as a controller for their gaming consoles [104].

Another application area is in *computer graphics* (CG) or *computer animation*, where markerless human motion capture systems will one day replace the commonly used marker-based systems to transfer human motion from a performing actor to a virtual avatar. As markerless systems evolve, they will do away with the disadvantages of their marker-based counterparts, where (•) the actors have to be equipped with several markers in a cumbersome process, (•) the actors have to wear skin-tight clothing, (•) the actions are restricted to a small capture volume, and (•) the final motion capture data has to undergo extensive manual post-processing to account for errors produced by occluded and temporarily missed markers.

Many applications are interested in analyzing human motions. *Clinical gait analysis* tries to detect signs for specific medical conditions based on the detailed observation of human gait. As in human motion capturing for entertainment purposes, gait analysis is currently performed using marker-based systems. The aforementioned limitations that come with the use of markers are especially critical when dealing with potentially impaired patients. High-accuracy markerless systems could thus provide new opportunities for medical research. Motion analysis is also becoming increasingly popular in the area of high performance sports, where athletes and coaches are interested in optimizing their motion sequences. As many exercises feature complex and spatially extensive motions, equipping athletes with markers is often not an option. We provide some examples where we have used our markerless motion capture system to track athletic exercises in Section 4.6.3. Finally, motion analysis is of interest in the area of industrial design. Ergonomic studies that aim at analyzing comfort and user-friendliness of new products are mostly relying on the analysis of static postures. Providing precise motion sequence data that is grounded in realistic anthropometric human models such as the one presented in Chapter 3 could yield valuable data for such studies. The car-mockup sequence from our experiment in Section 5.3.1 is a case study from the ergonomics community where the comfort of getting into and out of a new car design has to be assessed.

The main focus when developing the methods presented in this thesis has been on the domain of *robotics* and *intelligent environments*. The ability to observe and interpret human activities is crucial when dealing with *intelligent autonomous systems* that should be able to interact and to assist humans in their everyday lives. We have developed our methods in the context of the ASSISTIVE KITCHEN at the TUM [23], where a mobile robotic platform is deployed in assistive tasks in a kitchen environment to aid humans and to improve their quality of life. Considering the problem of aging societies, such scenarios are becoming increasingly important to enable people with minor disabilities to live independent for a longer period of time. We believe that our system for markerless observation of humans is an important contribution towards this ambitious goal, and we see it as the basis for many applications in this

context that require the capabilities to perceive and understand human activities. We will now present three such applications that are already being developed on top of the work presented in this thesis.

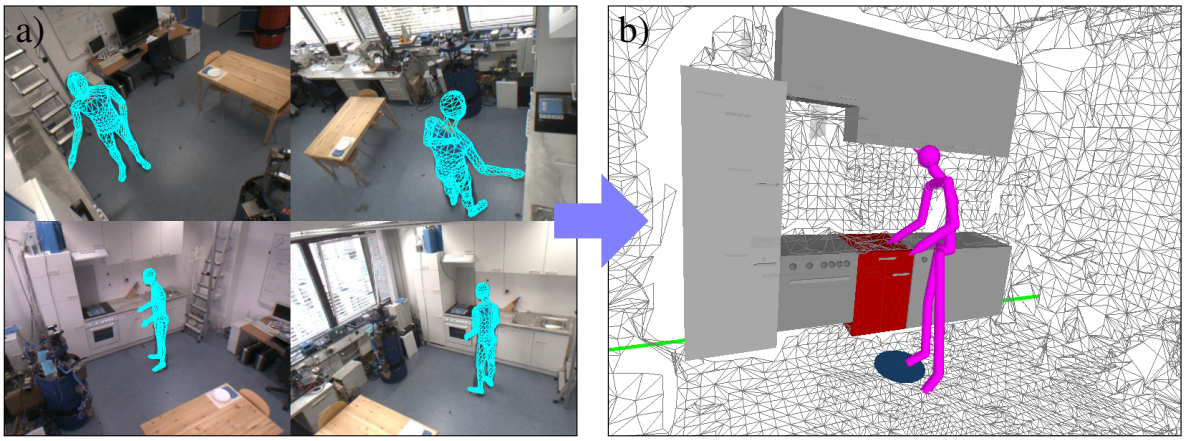


FIGURE 1.2 Integration of human motion capture data into a knowledge base: a) human motion capture data is retrieved using the multi-view motion tracking system proposed in this thesis; b) the data is integrated into a knowledge base where it is aligned with input from other perception modules (*e.g.* semantic environment maps or readings from the embedded sensor network) to enable higher-level reasoning about human activities [151]. The blue mark on the ground is the result of a query for action related places. The sequence shown is part of our publicly available TUM KITCHEN data set [150] (Section 5.3.3). Visualization from the knowledge base (Figure b) courtesy of Moritz Tenorth, TUM.

The first application is a knowledge base that serves as a source for practical knowledge processing in autonomous systems [151]. Input data from several perception modules is collected, synchronized and combined with encyclopedic knowledge to create a comprehensive database that can be queried based on first-order logics. For the ASSISTIVE KITCHEN environment, the perception input is provided by a mapping module that creates 3D semantic maps of the environment [127], by RFID and magnetic sensor readings from the embedded sensor network that provide the location of objects and the status of doors or drawers, and by semantically labeled human fullbody motion data retrieved by our system. Figure 1.2 shows the original videodata with our motion capture results on top and the corresponding visualization from the knowledge base. The knowledge base can be used to answer semantic queries such as *'give me the hand trajectories for picking up cups'* and to learn action-related models such as *'the place for picking up pieces of silverware'* from a set of observable features. Such a semantic description is important for analyzing and comparing human actions at different levels of abstraction, and provides autonomous systems with the means to learn important concepts based on observations and experience.

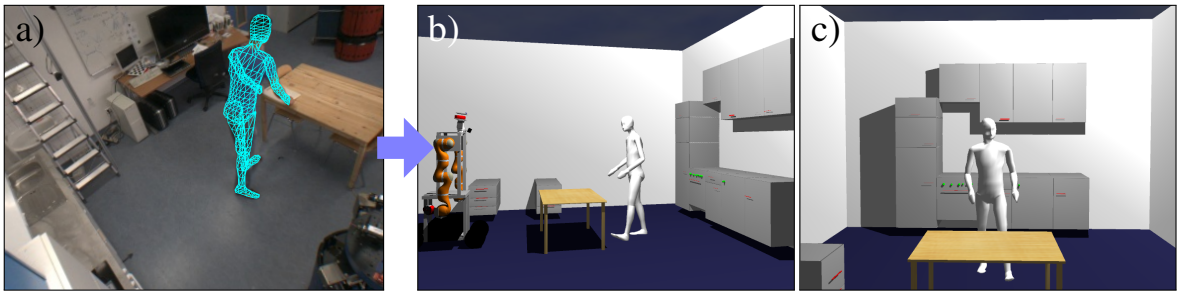


FIGURE 1.3 Integration of human motion capture data into the GAZEBO 3D robot simulator [113]: a) tracked human motion data b) playback of human motion data in a simulated scene with a mobile robot c) the same scene as perceived from the robot's perspective. The simulated human becomes a physical entity in GAZEBO that is perceived by the robot's sensing devices (*e.g.* laser scanners). Simulation images courtesy of Mihai Dolha, TUM.

Our second application example is the integration of human motion capture data into the GAZEBO 3D robot simulator [113]. Simulated environments are important tools in the development of autonomous systems that help to evaluate new algorithms without requiring access to real hardware, which can reduce costs and risks associated with the development. More important from a scientific point of view is the use of simulators to predict the effects of actions taken by the robot, to execute a selection of high-level plans to find the best sequence of actions to reach a goal, or to use the models of the environment for path-planning. Good simulators provide a realistic model of the environment and are capable to accurately model the robot's sensors and to predict the effects of its actuators based on the laws of physics. However, modeling humans as part of the simulation is a challenging task, given that human motion is complex and difficult to generate. In the application shown in Figure 1.3, we record human activities in the real environment, estimate the corresponding motion parameters, and then playback the motion in the simulated environment. The simulated human thus becomes a physical entity in the simulator that is perceived by the robot's sensing devices. Algorithms such as path-planning with dynamic obstacle avoidance can now be evaluated directly in the simulator with a good level of realism, as the human activities directly relate to the simulated environment.

The last application we want to use as a motivation for our work is the motion transfer of observed human motions to humanoid robots. By nature, human motions are highly complex and optimized depending on the current activity. Humanoid robots are an attempt to create robots that appear human-like, so that they are more likely to be socially accepted when acting in the proximity of humans. However, controlling the motion of humanoids remains a challenging problem. Leaving stability issues aside, the parameter space for performing spe-

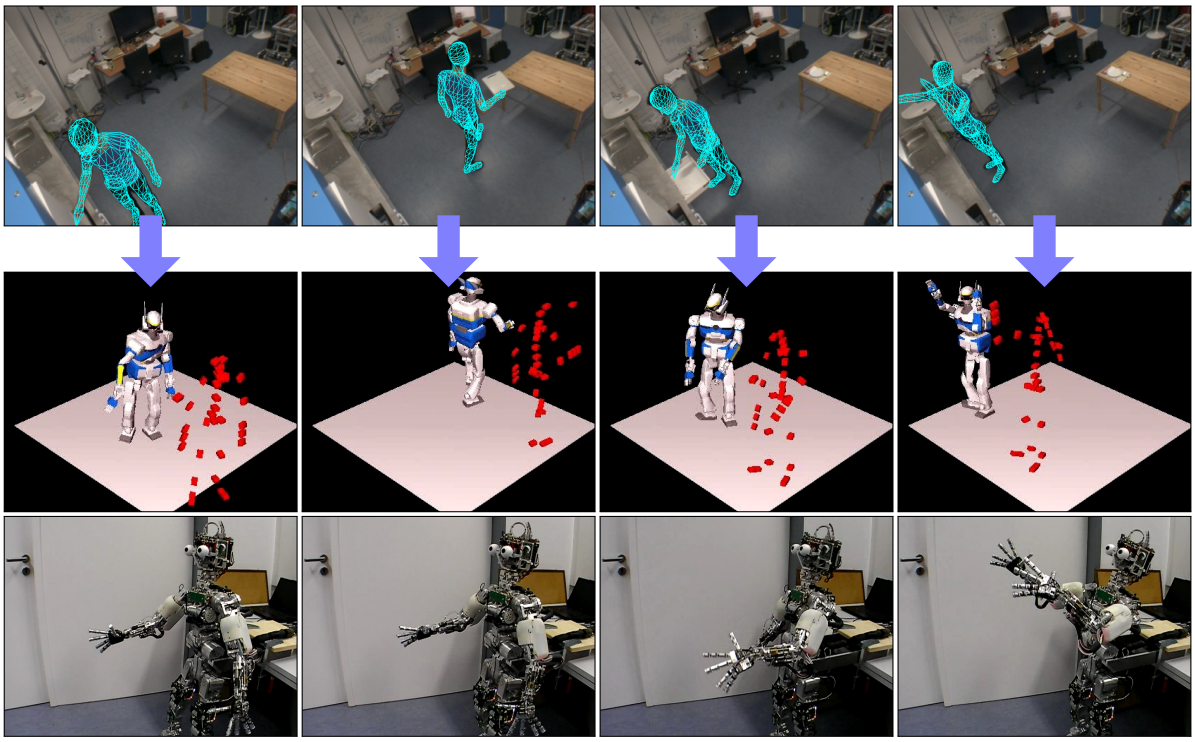

FIGURE 1.4 Transferring observed human motions to humanoid robotic platforms. The first row shows human motion capture data as retrieved by the markerless tracking system presented in this thesis. The second row shows simulation results for transferring the recovered motion to the HRP2 platform (images courtesy of Kei Okada and Masayuki Inaba, JSK Laboratory, University of Tokyo). The last row shows the same motion performed by the ICUB humanoid [155] (images courtesy of Federico Ruiz-Ugalde, TUM).

cific actions is highly redundant, *e.g.* an object can be grasped by the robot in many different ways. Yet not all of the possible motions will be human-like, and some will appear uncanny to a nearby observer. One possibility to avoid this phenomenon is to learn the motion control of humanoid robots by demonstration. Motion transfer from fullbody motion data retrieved by our system is one way to do this. Figure 1.4 shows two examples where we successfully transfered motion data captured in the ASSISTIVE KITCHEN to humanoid robotic platforms. In the first example, the motion data has been transfered to a simulated HRP2 platform in a cooperation with the JSK Laboratory at the University of Tokyo. The second example shows the transfer of the same motion to the ICUB humanoid [155] of our IAS group at the TUM.

We have presented a large variety of examples where the unintrusive observation of human fullbody motions provides the foundations for exciting new applications. The motions of a person can be used to actively control devices, be it the light in your house, your gaming console, or a humanoid robot. Motions reveal a lot about the intentions of a person, its mood,

or even its health status. They can tell us about the role of objects by putting them into context to the observed activities, an aspect that is often missed in knowledge processing. However, the bottleneck in all these applications remains the detailed perception of human motion. We hope to contribute with this thesis to remove some of the obstacles along the way.

## 1.2 Problem Description

As stated previously, knowledge about human activities can be of great value to many applications. However, the task of observing and interpreting human motions imposes many challenges. One of the biggest obstacles on the way is the perception of humans and their motions. Although humans have no difficulties to recognize and understand the motions of other humans, it is incredibly difficult to achieve similar perceptual capabilities in a machine. In this section we will describe the computational strategies and discuss some of the major challenges and pitfalls for automated observation and interpretation of human activities.

The computational problem can be split into two parts. The first part corresponds to the estimation of human motions. Because the motions of specific body parts are important indicators for ongoing activities, we need to be able to estimate the relative positions of body parts for the whole human body. Such a fullbody pose estimation is best performed using articulated human models. Once articulated pose data is available for each frame in a motion sequence, this data can be used as input to activity recognition. Here, the goal is to attach semantic labels to human motions, *i.e.* to classify the observed motion patterns. By associating motions with a *meaning* through semantic labels, subsequent applications are given the opportunity to reason about human behavior and intentions.

Let us now discuss the input data that is available to solve the problem. One of our prerequisites is to use only inexpensive and passive sensors. The obvious choice is to use CCD cameras, as they are among the richest passive sensors for perceiving the environment. They capture information about the environment in terms of a two-dimensional projection of the world on a discretized image plane, resulting in a 2D array of intensity values (*i.e.* an *image*). Color is represented using three intensity values, one for each color channel (usually red, green and blue). One disadvantage of cameras is that we need to infer 3D motion parameters from 2D sensors. Therefore, we make use of multiple cameras to account for this problem.

In a way, cameras resemble the human eye. However, most of the impressive capabilities of human perception are the result of visual processes taking place in the human brain. It is there that we reason about what we see, or in other words where we associate higher-level semantics to the low-level input data coming from the eye. All of these capabilities are missing

in available sensors such as cameras, and it is our task to infer meaning from the low-level data provided by the sensors.

The seemingly impossible problem of estimating detailed human motion parameters from a 2D array of independent intensity values and to reason about the corresponding human activities can be alleviated by modeling as many aspects of the problem as possible. This includes (•) the use of digital human models that accurately represent the shape and appearance of humans independent of the posture taken, (•) assumptions about the initial location and posture of each persons that should be observed, (•) prior assumptions about temporal limits of human motion, (•) knowledge about the characteristics of the sensors and a precise calibration of their setup in the world, (•) models of the environment that can be used to detect occlusions or to rule out places that are unreachable by humans, (•) the ability to extract representative features that can be compared with the human model, and (•) the availability of labeled exemplar data for all activities of interest.

A viable approach to estimate human motions is based on the *analysis-by-synthesis* principle. The idea is to predict the human pose for the current timestep and then use a digital model of the human to render the expected appearance for the hypothesized pose. The rendered model is then compared to the current image observations to assess the quality of the prediction. This is repeated many times, and the prediction that best coincides with the current observation is selected as the current pose estimate.

One of the challenges in this approach is the creation of a digital human model that is both accurate and efficient to compute. Many of today's approaches to human pose estimation use cylindrical models, where the body parts are modeled as cylinders attached to an underlying kinematic structure (see Figure 2.7b in the next section). Although these representations are efficient to compute, the realism of such models is low. At the same time, creating models that are accurate and realistic is a challenging modeling task in *computer graphics* as well as a tradeoff between accuracy and computational efficiency. Often, models that keep their realistic appearance for the majority of possible human postures require a larger number of free parameters, which adds to the complexity of the estimation problem. Other open problems with respect to the design of digital human models are whether the shape should be deformable (*e.g.* to model muscle contractions) or whether clothing should be modeled.

The comparison between the synthesized predictions and the image observations are usually not performed directly by comparing the individual intensity values (*i.e.* through *template matching*), but instead relevant features are extracted and then compared. The choice of good features has to be well-considered, as they should be suited to discriminate between good and bad matches. At the same time they should be robust with respect to noisy observation data

or changing lighting conditions, both of which are unavoidable in practical situations. Typical candidates for features are silhouette shapes, image contours or color features (see Figure 2.6 for more options). Other problems that frequently occur are shadows in the observation that can confuse the feature extraction, or wide clothing that might alter the appearance of the observed human subjects when compared to their digital models.

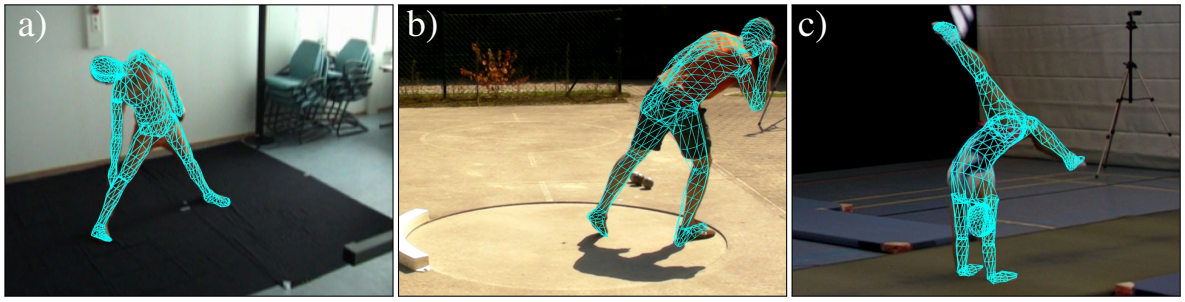


FIGURE 1.5 Challenging motions as tracked by our system: a) random exercise b) shotput exercise c) gymnastic exercise. Such motions are difficult to learn in advance.

One of the biggest challenges is the prediction of pose hypotheses. To keep the problem tractable, the usual assumption is that the human pose is known for the last timestep. Based on this knowledge, we want to predict the pose for the current timestep. Still, the space of possible poses is incredibly large, due to the relatively high dimensionality of the human model. To give an example, the challenging task of rigid body tracking in 3D has 6 d.o.f., whereas articulated structures such as the human body easily exceed 30 d.o.f. Therefore, many approaches learn priors from training data to better predict human motion. However, this strongly restricts the type of motions that can be detected, something that we want to avoid in our work by allowing arbitrary types of motions (Figure 1.5). The price to pay is a drastically increased state space that has to be searched efficiently to find the correct pose.

When estimating the current pose, one also has to take into account uncertainties that arise in almost all aspects of the problem. Sensor readings are noisy, our models of the human or its motion might be imprecise, and our observations correspond to a projection of the real world where relevant information might be lost. Thus, we believe that the estimation problem should be approached using probabilistic methods that are able to select the most probable solution based on the observations made so far. Still, most of the well-known probabilistic estimators are computationally intractable when faced with the high-dimensional problem of human pose estimation. In addition, nonlinear motion and observation models with many local maxima further complicate the task. A detailed discussion of these problems and suitable solutions are presented in Chapter 4.

The challenges mentioned so far apply to the tracking of a single person in an empty room.

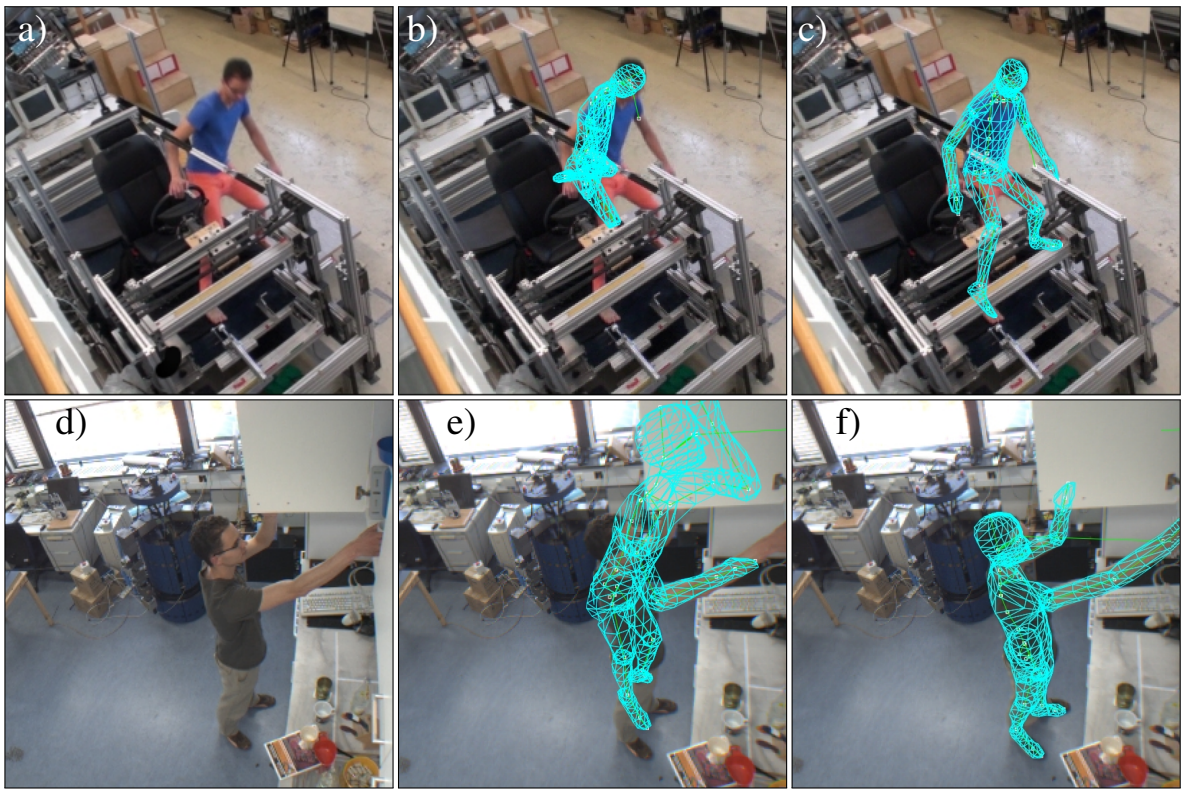


FIGURE 1.6 Challenges encountered in cluttered and dynamic environments. First row: A scenario where the tracking subject is occluded by a car-mockup (a); even sophisticated motion tracking methods will fail without a model of the environment (b). Second row: A scenario featuring a subject manipulating its environment (d); sophisticated tracking methods will be confused by the dynamic non-human parts of the scene (e). The last column (Figures c and f) shows the results when using methods proposed in this thesis.

When trying to track humans in natural environments during everyday activities, additional challenges arise. Humans can be occluded by parts of the environment, which might create discrepancies between expected features and the observations. Occlusions can be complex, *e.g.* a table might occlude the legs of the person while the arms are visible above the table. Furthermore, many human activities involve the manipulation of objects or parts of the environment. These activities might again create discrepancies between the observations and the predicted appearance due to occlusions from manipulated objects. This also complicates the feature extraction step due to dynamic objects that are present in the scene (*e.g.* when segmenting silhouette shapes). Finally, humans might not be alone in the scene, *i.e.* when several humans interact. This requires mechanisms to distinguish between features corresponding to each of the humans. Figure 1.6 indicates why it is important to consider these problems, which to date have been only insufficiently covered by human motion capture research.

All of the challenges mentioned so far have been related to the observation of human motions. The ultimate goal in *human-robot-interaction* (HRI) however is to understand human behavior. To achieve this, one must be able to associate semantic labels with the observed motions, so that robots will be enabled to reason about human activities and intentions. The problem of assigning these labels is a classical *machine learning* task referred to as *activity recognition*. Two research directions are typical for vision-based activity recognition. The model-free approach is to learn a direct mapping between image features and action classes. The main challenge here is to find a set of discriminative features that can be reliably detected under all conditions. The alternative model-based approach is to use pose parameters derived from human pose estimation as features for activity recognition. While pose parameters are potentially rich and representative features for activity recognition, the main bottleneck for this class of approaches remains their estimation.

## 1.3 Contributions

The work presented in this thesis contributes to a broad range of research and application areas within computer science, although our main research interests are in *computer vision* and *robotics*. The result and main contribution of our research is an integrated system for observation and interpretation of complex human activities. The system is able to accurately estimate human fullbody motions for a large spectrum of activities in realistic environments. This includes manipulation actions where objects are being handled and where interactions with the environment or between humans take place. On top of that, it can be used to recognize ongoing activities by inferring the semantic labels from the observed motion patterns.

The following theoretical and practical contributions have led to the development of this system and add to its high accuracy and robustness:

- We have adapted an anthropometric human model from the ergonomics community for use in markerless human motion tracking. We introduced the following optimizations to account for the specific requirements in high-dimensional pose tracking: (•) ergonomically sound interpolation of the spine to reduce the dimensionality of the pose parameters; (•) estimation of biomechanical inter-frame motion limits to reduce the state space during prediction; (•) caching of body-part dependant pose calculations to improve the computational efficiency of the model. Furthermore, we have learned a reduced set of shape parameters for the model that enables us to quickly adapt the shape and size of the model to match the tracking subject. In summary, the presented model

provides us with an efficient yet accurate biomechanical representation of human shape, posture and motion.

- We present a novel hierarchical sampling strategy for high-dimensional pose estimation in a probabilistic framework. Our algorithm combines hierarchical state space decomposition with a multi-layered search and parallel partitioning schemes to create a target-oriented yet highly diversified sampling strategy. It has been shown to be effective at finding the global maximum of a nonlinear weight function with lots of local maxima. This *branched iterative hierarchical sampling* algorithm outperforms state-of-the-art Bayesian methods for articulated pose tracking and is a key contributing factor to the high accuracy and constant reliability of our human pose tracking method.

- We introduce a simple yet effective color-based appearance model that can be used to enrich shape-based observation models with color information. This appearance model is also an integral part of our *layered environment models* that combine shape and appearance information to provide an implicit way to model the environment without the need for 3D models. This allows us to deal with cases of environmental occlusion that occur when (•) a subject manipulates objects or parts of the environment, (•) parts of the environment can both occlude and be occluded by humans (*e.g.* tables), or when (•) two interacting subjects are tracked at the same time.

- We present self-training motion models that learn environment- and task-specific motion patterns over time. This is used to improve the prediction and thus the computational efficiency of our tracking system. We use *spatio-temporal neighborhood graphs* that are closely related to ST-Isomap [74] as an efficient and incrementally extendable representation for learned pose manifolds.

- We additionally use *spatio-temporal neighborhood graphs* for activity recognition by providing semantically labeled training sequences. The semantic labels are retrieved during tracking, which facilitates reasoning about the currently observed activities.

The validity of all proposed approaches has been proven in extensive experimental evaluation on challenging benchmarks and video sequences. These include the HUMANEVA benchmarks for markerless human motion capture, sports sequences for shot-put and floor exercises, ergonomic case studies in a car mock-up, and human manipulation activities in a kitchen environment. In particular, we created the publicly available TUM KITCHEN Data Set [150] that currently consists of 21 sequences of everyday manipulation activities performed by 4 different subjects in our kitchen environment. We provide the original calibrated video sequences,

human motion capture data as retrieved by the system presented in this thesis, semantic activity labels for every frame, and complementary sensor readings from RFID readers and magnetic sensors embedded in the environment.

Our research on human pose tracking and activity recognition resulted in several peer-reviewed publications that are all based in parts on work presented in this thesis [16, 18, 17, 150, 20, 21, 23, 51, 145]. Furthermore, several research projects at the TUM IAS group build upon the work presented in this thesis.

## 1.4 Thesis Outline

The remainder of this thesis is organized as follows:

**Chapter 2. Overview:** We give a general overview on strategies for observing and interpreting human activities. This is split into a brief overview of the setup and algorithmic methodology used in our system (Section 2.1) and related work on both human motion capture and human activity recognition (Section 2.2). This chapter is recommended for readers that want a short but comprehensive introduction on the research topics addressed in this thesis and on related alternatives.

**Chapter 3. Anthropometric Human Model:** We present and discuss the anthropometric human model used in our system. This model contributes to the accuracy and realism of the estimated pose sequences due to its realistic appearance. We present both relevant specifications of the model as well as algorithmic improvements that are tailored towards the use of the model in tracking applications.

**Chapter 4. Human Pose Tracking:** We discuss human pose tracking and present our contributions. We start by giving an introduction to recursive Bayesian estimation which corresponds to the probabilistic framework used for tracking. We then discuss common Bayesian estimators such as the *Kalman filter*, the *extended Kalman filter* and *particle filters*, and motivate the use of a *particle filter* framework (Section 4.1). We present our choice for the *motion model* (Section 4.2) and the *observation model* (Section 4.3) that define how to predict the motion between timesteps and how to match our predictions to the current image observations. Due to the high-dimensionality of the state space when tracking articulated human models, standard *particle filtering* becomes computationally intractable. In Section 4.4 we discuss state-of-the-art variants of *particle filters* that utilize hierarchical sampling strategies to make pose tracking feasible. We also

point out weak spots and limitations of these methods. In Section 4.5 we then introduce *branched iterative hierarchical sampling* as a novel hierarchical sampling strategy that combines complementary strengths of the aforementioned approaches in a highly diversive sampling strategy that is accurate and robust to noisy observations. We consider this sampling strategy as one of the key contributions of this thesis. We justify our view by extensive evaluation of the methods presented in this chapter on simulated and real sequences (Section 4.6).

**Chapter 5. Appearance and Environment Modeling:** We show how to apply our human pose tracking methods to track complex manipulation activities in everyday environments. We first introduce a color-based appearance model for our digital human model and show how it can be used to augment the shape-based observation model presented in the previous chapter (Section 5.1). The appearance model is also used in the layered environment models presented in Section 5.2 that provide the means to implicitly model the environment. They are used to model occlusions and to filter dynamic objects (possibly being manipulated by the human) or dynamic parts of the environment (*e.g.* doors). We again present extensive experimental evaluation of our proposed methods on several challenging sequences, including a car-mockup scenario with occlusions and complex manipulation activities in a kitchen environment (Section 5.3). We also introduce the TUM KITCHEN data set where we provide 21 sequences of kitchen activities from 4 different subjects combined with complementary sensor data. Finally, we present a joint-action scenarios of two humans interacting in the kitchen.

**Chapter 6. Learning Models of Human Motion:** We present a method to learn environment-specific models of human motion that are used to improve the prediction of human poses and to recognize human activities. In Section 6.1.1 we introduce *spatio-temporal neighborhood graphs* that allow us to learn pose manifolds from training data. In Section 6.1.2 we show how motion models can be learned incrementally using the human motion tracker presented in Chapter 4, and how to use them to improve motion prediction and thus the efficiency of our tracker. Finally, we show in Section 6.1.3 how these models are used for activity recognition. We evaluate our contributions in Section 6.2 on several sequences from the HUMANEVAII and TUM KITCHEN data sets.

**Chapter 7. Conclusion:** We conclude with a summary of this thesis and a discussion of open research challenges and future work.

# CHAPTER 2

# Overview

## Strategies for Observing and Interpreting Human Activities

In this chapter we will give an informal overview on methodologies for observing and interpreting human activities. It is meant to introduce some of the underlying concepts and strategies that are commonly used in the related research areas, and to facilitate the understanding of the remaining chapters of this thesis. Readers that are already familiar with the concepts involved in human pose estimation and activity recognition may safely skip this section.

We will distinguish between the observation of human motions and the recognition of human activities. When talking about the observation of human motions, the goal is to provide a mathematical description of the observed motions. Usually, the motion is described as a sequence of postures and the mathematical description corresponds to the parameterization of an articulated digital human model. The task of estimating these parameters is generally known as *human pose estimation* (for single frames or motion sequences) or also as *human motion capture* (only for motion sequences). When the poses are estimated iteratively based on a known initial pose, the task can be referred to more specifically as *human pose tracking* or *human motion tracking*. The recognition of human activities on the other hand aims at providing semantic labels to coherent sequences of human motions. This is known as *human action recognition* or *human activity recognition*. Often, the mathematical description of the observed human motion sequence serves as an input for the activity recognition.

In the next section we will provide a general overview on our system for markerless human motion capture and activity recognition, which we will describe in greater detail in the remaining chapters of this thesis. We will describe the scope of our work, the underlying components, and their interplay. We will then present related work on marker-based and markerless human motion capture and on human activity recognition. This should help to provide the reader with broad knowledge on taxonomies and common approaches in these research areas, and also to classify the work presented in this thesis with respect to other approaches.

## 2.1 System Overview

We will now give a brief overview on the system for observing and interpreting human motions that we developed and that constitutes the foundation of this thesis.

The main component of our system is the observation of human fullbody motions from camera images. As opposed to marker-based tracking systems that are commercially available in a wide variety (see Section 2.2.1 for an introduction), we use an unintrusive setup where the human subjects need not be specifically prepared prior to observing their motions. This enables a variety of new application scenarios that would not be feasible using intrusive methods (*e.g.* long term observations of humans in their everyday environments). However, the increased flexibility brings along new algorithmic challenges, as it is no longer possible to use easily detectable markers to locate the positions of body joints.

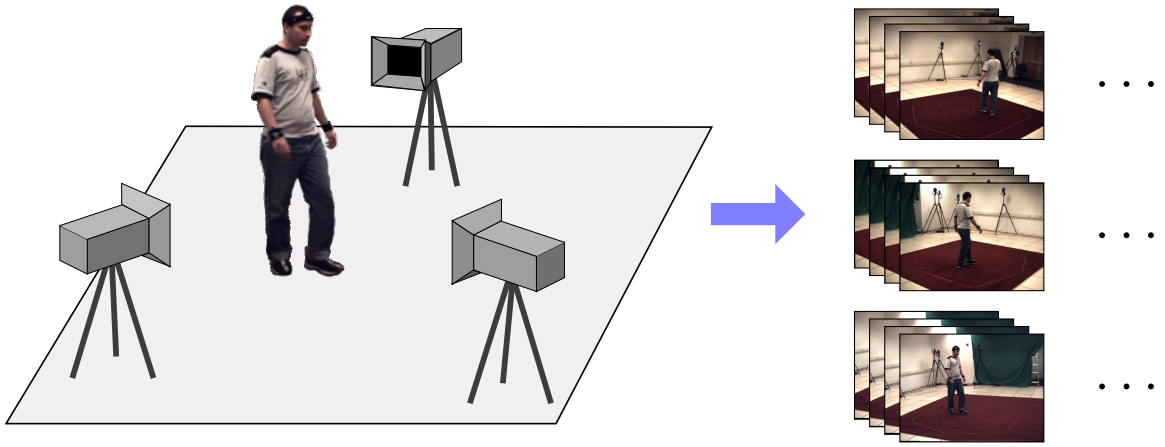

FIGURE 2.1 Camera setup for human motion capture. A minimum of three cameras records the subject from different viewpoints. The resulting image streams serve as input for our pose tracking algorithm. The subjects need not wear any markers, but their clothes should stand out from the background. Except for the cameras, the setup is completely unintrusive.

Figure 2.1 shows the principle setup used in our system. Multiple cameras are placed in the environment to observe the human subject from different viewing directions. This setup is comparable to that of optical marker-based systems, with the exceptions that it is unintrusive and that three cameras are sufficient for our task (assuming there are no occlusions). The required hardware is affordable and easy to obtain (standard CCD cameras at low resolution and a desktop computer to capture and process the video streams). The image streams recorded by the cameras then serve as the input to our pose tracking algorithms.

We use a model-based tracking approach to estimate fullbody poses of the observed human recursively based on the estimate from the previous timestep. This requires that an initial pose

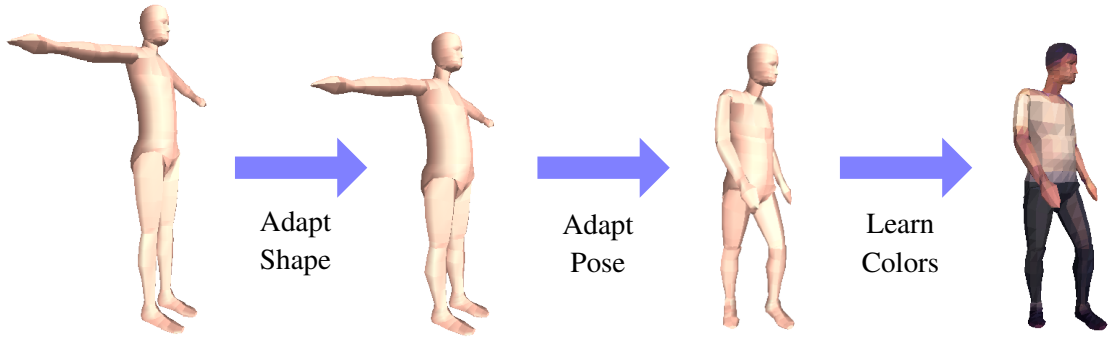


FIGURE 2.2 Model initialization before starting the pose tracking algorithm. A digital human model is first fit to the tracking subject by adapting its size and shape. Then, the pose of the model is adapted to match the first frame for tracking. Finally, the color appearance of the model is learned (this is actually an incremental process, see Section 5.1.2).

estimate is known before starting the tracking. Furthermore, the human model that is used for tracking must be adapted to the current subject to match its appearance. Figure 2.2 depicts the necessary steps for model initialization. In a first step, the default model is adapted to the shape and size of the human subject. This step is important, as a good estimate of the relative positions of the joints in the model helps to achieve good tracking accuracy. In the second step, the model is adapted to the pose of the subject in the first frame of the tracking sequence. Based on this frame, the tracking is then performed frame-by-frame. A third and optional initialization step is to learn the appearance of the tracking subject in terms of colors or textures of the model. This step is optional and only necessary when color or texture features are required during estimation (see Chapter 5). In practice, the colors of the model cannot be learned from the initial frame only, but have to be learned from several frames with known poses. Alternatively, they can be refined during tracking.

Throughout this thesis we will assume that initialized models for the tracking subjects are readily available, and that the initial pose is known. In our system, these steps are performed manually by the user with the help of a *graphical user interface* (GUI) that allows for fast adaptation of shape and pose parameters of our human model. Both shape and pose adaptation can be performed by an experienced user in just a few minutes. To simplify the adaptation of the model shape, we introduce a special reduced parameterization in Section 3.2. This parameterization has the potential to be used for automatic shape adaptation of the model, however the automatic initialization of shape and pose parameters is out of the scope of this thesis. The shape adaptation of the model is necessary only once per subject, so that practical applications could store initialized models based on the identity of the tracking subject. In contrast, the initialization of the pose is necessary for every continuous sequence to be tracked.

However, this needs to be done only approximately due to the large convergence radius of the pose tracking algorithms proposed in Chapter 4. Experiments have shown convergence up to a pelvis translation of about $0.5\ m$ when the initial posture roughly resembles the observed pose.

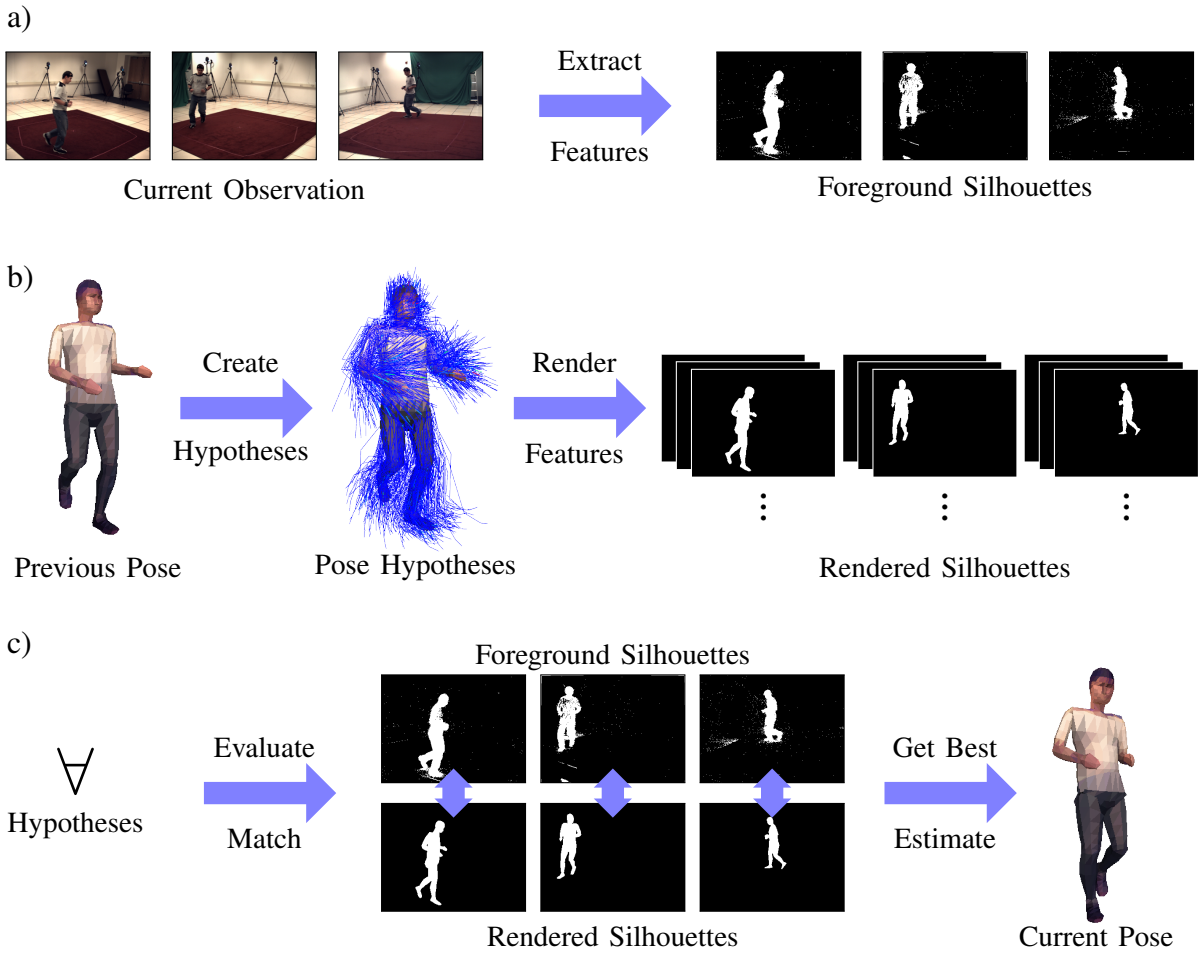


FIGURE 2.3 Schematic diagram for one timestep of human pose tracking: a) the current image observations are used to extract relevant features; b) the known previous pose is used to create hypotheses of the current pose, and the expected corresponding features are rendered for each hypothesis; c) the expected features for each hypothesis are matched against the observed features, and the best match is selected as the estimated pose. In this diagram we use human silhouettes as features, according to the shape-based observation model described in Section 4.3. These features can be augmented by color features as described in Section 5.1.2.

Once the initialization is done, our pose tracking algorithm processes the observed sequence frame-by-frame to estimate the corresponding sequence of poses (Figure 2.3). In each timestep, the input to the tracking algorithm consists of the current image observations and

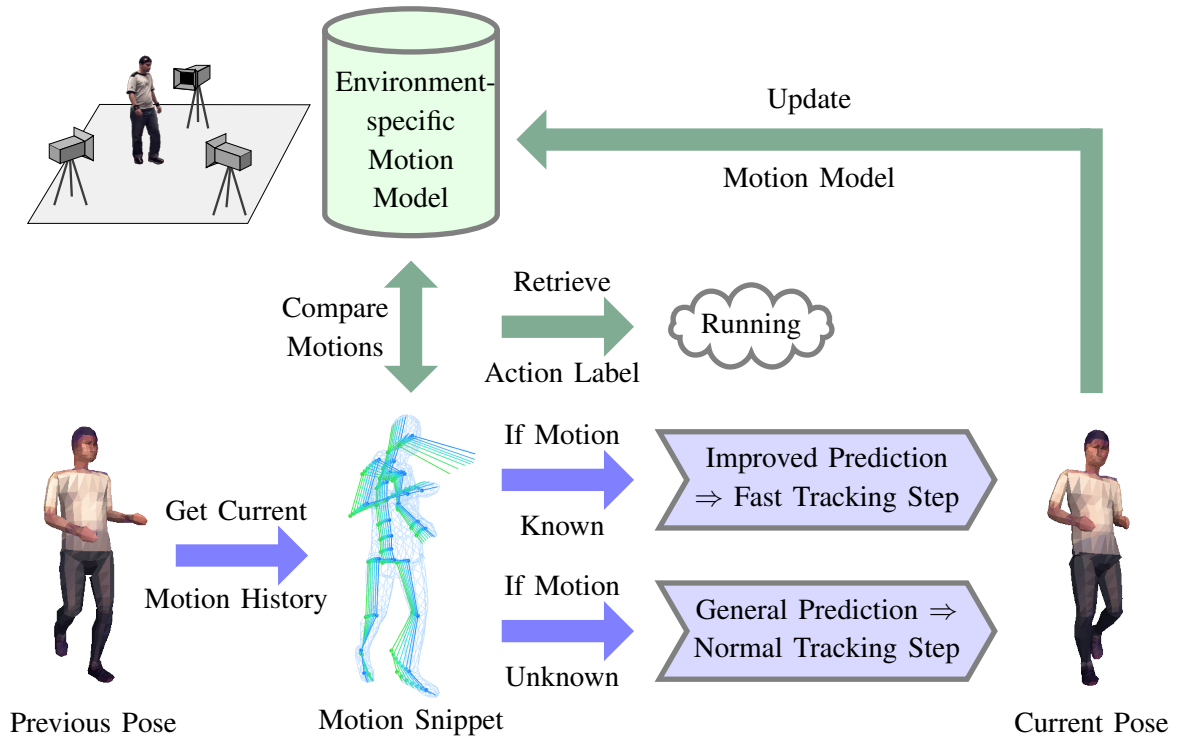


FIGURE 2.4 Using environment-specific motion models to improve the prediction for human motion tracking and to recognize human activities. The motion model is trained with previously observed motion patterns that are typical for a specific environment. During tracking it is used to compare trained patterns with motion snippets that encode the short-term motion history of the last estimated pose. Whenever correspondences are detected, they are used to predict the current pose with greater accuracy using the information stored in the motion model. This allows us to use a computationally more efficient scheme for tracking. If no correspondences were found, a normal tracking step with a general motion model is performed. The newly estimated (unknown) pose is then used to update the motion model. In addition to its use for motion prediction, the motion model can be used to recognize actions or activities whenever semantic labels have been provided with the training data.

the pose estimate from the last timestep. Tracking itself works in an *analysis-by-synthesis* manner. Based on the previous pose we create several hypotheses that predict the current pose (using the *motion model* in Section 4.2). These hypotheses are used to render (or synthesize) image features as they would be expected to be observed in the cameras (Figure 2.3b). The rendered image features are then compared to real features extracted from the current image observation (Figure 2.3a). This is done for all hypotheses, and the best matching hypothesis is selected as the current pose estimate (Figure 2.3c). The type of features and the procedure of comparison is defined by the *observation model*. We use a shape-based observation model that compares silhouettes extracted from the camera images by means of foreground-background

segmentation (Section 4.3). Furthermore, the features can be augmented by color features as described in Section 5.1.2. Note that the preceding description of our pose tracking algorithm is an illustrative simplification of the true underlying probabilistic framework presented in Chapter 4.

The space of possible human poses is very large, and human motion patterns can be complex and hard to predict. This has to be taken into account when generating pose hypotheses during tracking, *i.e*. when predicting the next pose. When no prior knowledge about the type of observed motion is available, we have to generate and evaluate a large number of hypotheses to end up with a good final pose estimate. One possibility to improve the prediction is to learn models of human motion from training data. However, good training data is difficult to come by and often requires the use of expensive marker-based motion capture systems. In the end, trained motions are often unrelated to motions that are specific to an environment, and thus of no practical use. In our system we learn environment-specific motion models over time and use these to improve our prediction for repeatedly observed activities. Figure 2.4 depicts this strategy. In each timestep, we use the current motion history to create so-called *motion snippets* that represent a short motion pattern. These motion snippets are matched against the learned motion model to check for corresponding motion patterns. Whenever no match is found, we use the tracking scheme as illustrated in Figure 2.3 with an uninformed (and computationally expensive) motion model to create the new pose hypotheses. The final pose estimate is then used to self-train the environment-specific motion model and therefore to account for future detections of similar motion patterns. Whenever known motion patterns are detected however, we use knowledge about likely successors that is derived from previous observations to create potentially much more accurate pose hypotheses. Due to the improved prediction, we can then switch to a more efficient tracking step (again according to Figure 2.3 but with less hypotheses required for evaluation).

The learned environment-specific motion models are not only used to improve the prediction of our tracking system, but also to perform activity recognition during tracking. Our straightforward approach is to train the motion model with semantically labeled pose sequences, so that every motion pattern is associated with an action or activity label. When looking up motion correspondences during each tracking step, we can then retrieve the best matching action labels to estimate the currently observed activity, as illustrated in Figure 2.3.

After tracking a sequence of human motion, we suggest a post-processing step to smooth the estimated pose sequence as described in Section 3.4. This then results in a realistic and steady motion sequence.

## 2.2 Related Work

A large body of related work is available both in the context of human motion capture and human activity recognition. We will discuss the main concepts and present corresponding references in the following sections. For human motion capture, we will present both marker-based and markerless approaches, but the main focus will be on markerless approaches in accordance with the scope of this thesis.

### 2.2.1 Marker-based Human Motion Capture

The vast majority of systems for human motion capture are marker-based. Such systems are intrusive and require the subject to be equipped with several passive or active markers. As these markers need to be attached close to the body, the subjects are further required to wear skin-tight clothing. Marker-based systems can be divided into optical or non-optical systems. Optical systems [147, 146, 37] use calibrated camera arrays to detect the markers in several images and to triangulate their 3D positions. To ease the detection of markers, one can use either active markers that periodically emit light to identify their position [146, 37], or passive markers that reflect light emitted by special lightsources near the cameras (*e.g.* infrared light) [147]. Amongst the non-optical systems are magnetic systems [40] where the 6 d.o.f. pose of the markers inside a magnetic field is measured based on the relative magnetic flux (Figure 2.5). These systems require less markers and are able to detect these even when they are visually occluded, but the capture volume is usually smaller when compared to optical systems, and metallic objects can confuse the measurements. Inertial systems [153] use small inertial sensors (*e.g.* gyroscopes) that are attached to the body and translate the measured motion directly onto a biomechanical model of the human body. These systems do not require external cameras or emitters which makes them more convenient to apply in many situations, but they can suffer from positional drift due to errors accumulating over time.

### 2.2.2 Markerless Human Motion Capture

Early approaches on markerless vision-based human motion capture mostly target monocular tracking and therefore often rely on two-dimensional human models. The used 2D models are often made up of image patches that correspond to individual body parts and that are connected at the body joints (Figure 2.7a). Such models have been presented by Cham and Rehg as *scaled prismatic models* [38, 165] or by Ju *et al.* as *cardboard models* [75, 67] and differ in the number of free parameters for each body part. While *scaled prismatic models* are parame-

FIGURE 2.5 Magnetic system for marker-based motion capture. In this example, the system is used to record grasping motions. The glove is equipped with several markers to detect the articulated pose of the hand. Image courtesy of Alexis Maldonado, TUM.

terized by an in-plane rotation and relative scaling, *cardboard models* are additionally able to model perspective deformations of body patches by using 3D motion parameters for each body part. Felzenszwalb and Huttenlocher introduced *pictorial structures* [52, 120] that are related graphical models where the spatial arrangement of parts as well as their appearance is parameterized by probability distributions. 2D models are still in widespread use today [165, 120], as they provide reasonable results for motion estimation based on single camera views. They are mostly used in *bottom-up* approaches where body parts are first detected independently and then combined to a final connected posture that corresponds to the most consistent pose given the available image observations [120]. However, the lack of 3D information restricts the aforementioned approaches to rough approximations of the true postures. The typical complexity of 2D models is around 22 d.o.f. [165].

Agarwal and Triggs [3] extract more detailed and precise 3D poses from monocular camera views by learning a mapping from observed shape features to the corresponding pose parameters of human models. The mapping is learned from humans silhouettes that have been synthetically generated using available human motion capture data (a similar approach using multiple cameras is taken by Grauman *et al.* [63]). Statistical inference is then used to recover unknown 3D pose parameters from the mapping and extracted foreground silhouettes. A related approach by Howe *et al.* [69] tries to compensate for missing 3D information by learning patterns of human motion from training data. Sigal and Black [137] first estimate 2D postures in a bottom-up approach before attempting to classify the corresponding 3D motion sequence.

As only small subsets of all possible human motions (*e.g.* walking) can be learned directly at reasonable expense, several methods use more than one camera view to obtain precise 3D measurements of unconstrained human motions. An intermediate step is taken by approaches that use stereo cameras to acquire the missing 3D information that is necessary for accurate pose estimation [112, 64, 11]. Such methods can achieve good results in many application areas, however they remain viewpoint-dependant when it comes to self-occlusions in the human body, due to the relatively small baselines between cameras in a stereo-rig. Therefore, a large amount of published work uses multiple cameras that differ substantially in their viewing directions. This can lead to improved pose estimation accuracy when used in combination with three-dimensional human models.

Approaches that use 3D models with multiple cameras are usually *top-down* approaches, *i.e.* the pose parameters of the model are first predicted and then evaluated based on the available image information. They can be further distinguished according to several criteria.

Probabilistic approaches try to infer the solution using recursive Bayesian estimation. As a closed-form solution is computationally intractable due to the nonlinear probability distributions in both the motion and observation models, the solution is usually approximated using sequential *Monte Carlo* methods (*particle filters*). Common approaches such as *sampling importance resampling* fail due to the high dimensionality of the problem. Deutscher and Reid [46] proposed the *annealed particle filter* that is related to the method of *simulated annealing* [81] to evolve the particle set towards the global maximum of the weight function. MacCormick and Isard [94] introduced *partitioned sampling* to estimate the parameters of articulated models by splitting the state space into hierarchical partitions that can be evaluated sequentially, thus reducing an initially high-dimensional estimation problem into several lower-dimensional problems. Mitchelson and Hilton [99] presented a variant of hierarchical sampling where partitions are evaluated in parallel to increase efficiency. Wang and Rehg [165] have evaluated non-hierarchical variants of particle filters in the context of monocular figure tracking.

Other work uses deterministic as opposed to stochastic methods. Here, the parameters are estimated based on nonlinear optimization [31, 79, 58, 83]. A good initial estimate is necessary to avoid getting stuck in local minima of the objective function. This can have a negative impact when tracking at low video sampling rates, where the motion between two timesteps is large. Rosenhahn *et al.* [121] use several randomly chosen starting points for the optimization to reduce the risk of getting stuck in local minima. Bray *et al.* [29] circumvent this problem by combining nonlinear optimization with *Monte Carlo* methods. They present results on a hand tracking sequence, a task comparable to human motion tracking due to

the similar kinematic structures of articulated hand models and human models. Ivekovič *et al*. [72] presented a closely related approach where they use *particle swarm optimization* to track upperbody movements. Several approaches use variants of the *iterative closest point* (ICP) algorithm to register the human model to a 3D point cloud [65, 109, 82].

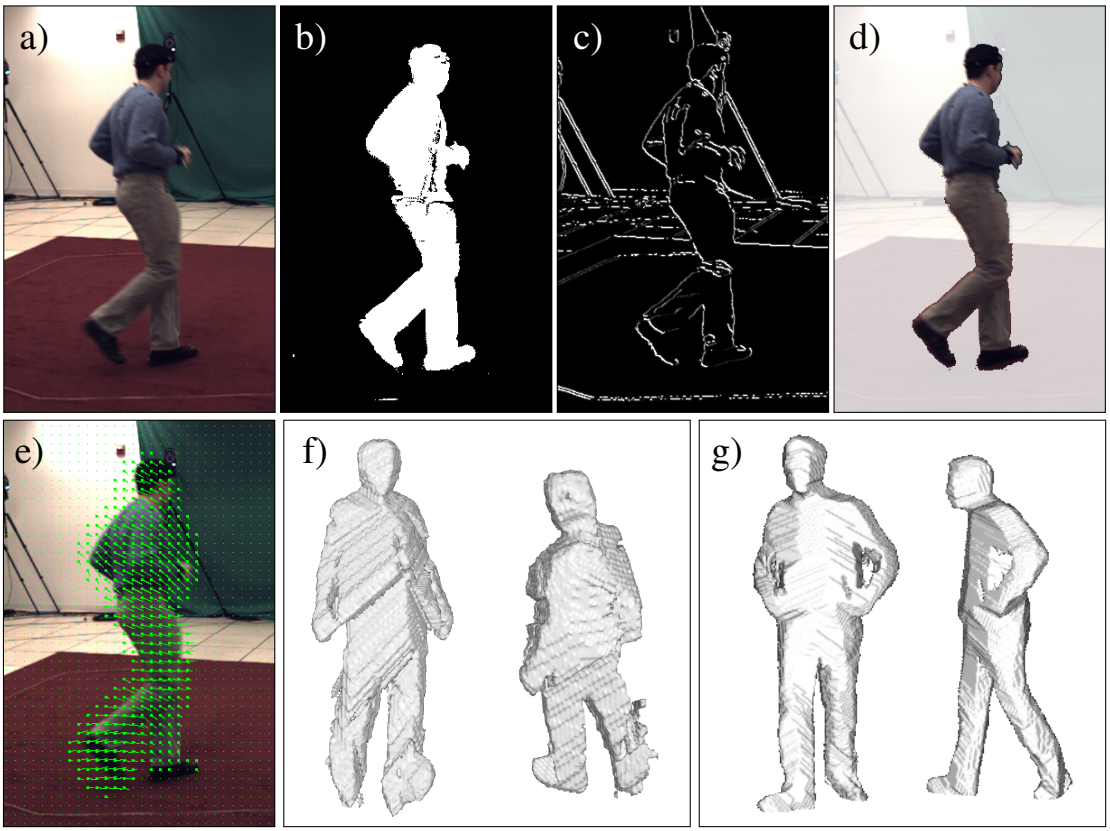


FIGURE 2.6 Features commonly used in human pose estimation: a) original image b) silhouette/contour shapes c) image edges d) color features e) optical flow f) visual hull extracted from 4 cameras g) visual hull extracted from 16 cameras. All visual hull images courtesy of Alexander Ladikos, TUM [89].

Another possibility to distinguish approaches for 3D human pose estimation is given by the different observation models that are used (Figure 2.6). Feature-based observation models often compare foreground silhouettes/contours that have been extracted from the current image with the projections of the human models for the estimated pose parameters [46, 121]. Extracted image edges are also commonly-used features [162, 141, 46], however they can lead to inaccuracies due to folds in the clothing of the subjects (Rosenhahn *et al*. [122] present a method that uses physics-based models of human clothes to track a woman wearing a skirt).Sminchisescu and Triggs [141] and Brox *et al*. [32] use *optical flow* as a complementing feature to the human contour. Some approaches use appearance models based on color or

texture information [168, 15]. Methods that build 3D reconstructions of the human surface from multiple cameras (so-called *visual hulls*) are becoming increasingly popular. They can be used to measure the deviation between the surface of the human model and the nearest reconstructed 3D points [97, 79, 68]. Other related approaches do not use explicit models at all and estimate the body joint positions directly from clusters of 3D points that share the same motion patterns [39, 6]. Methods that utilize *visual hulls* require a large density of cameras in the environment to be able to reconstruct the observed subject at good accuracy (usually a minimum of 8 cameras is needed, see Figures 2.6f and 2.6g).

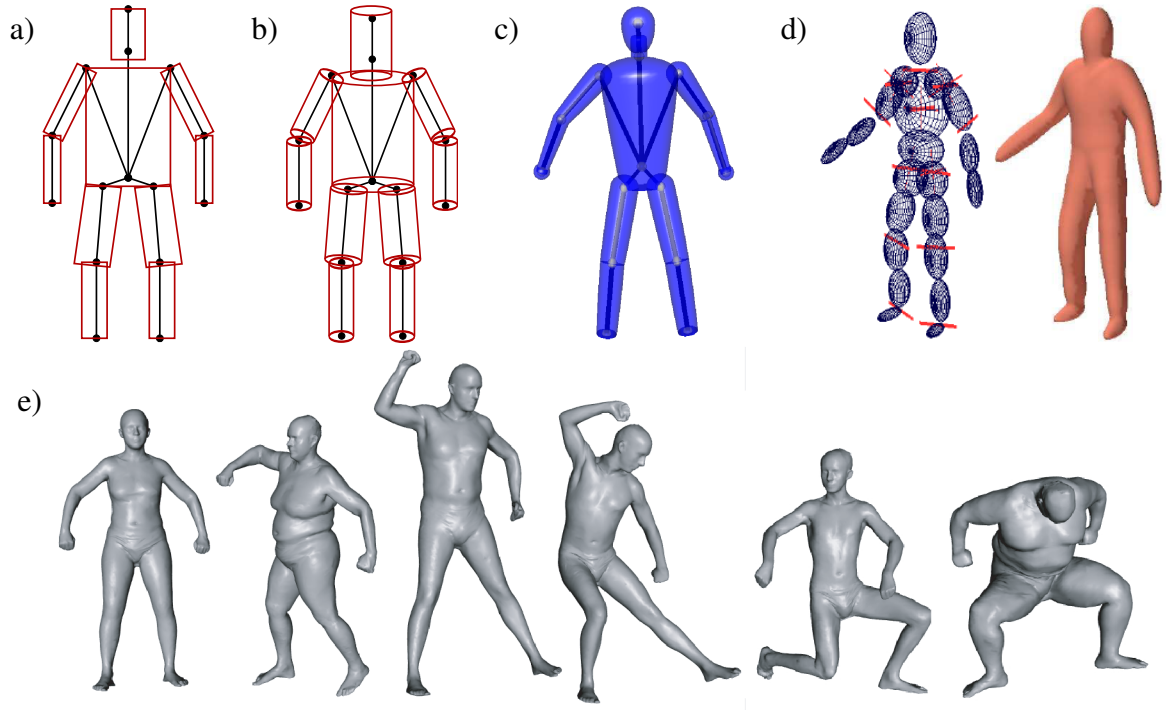


FIGURE 2.7 Different types of digital human models used in human pose estimation: a) 2D planar model as used for monocular tracking b) 3D cylindrical model c) 3D model based on superellipsoids (reprinted from Kehl and van Gool [79], © 2006 Elsevier) d) 3D model with implicit surface defined as a blending of 21 ellipsoids (reprinted from Horaud *et al.* [68], © 2008 IEEE) e) 3D SCAPE model with deformable surface mesh (reprinted from Anguelov [5], © 2005 Dragomir Anguelov, Stanford University).

The 3D human models that are being used can also help to categorize approaches. While almost all 3D models are based on an underlying kinematic structure that resembles the human skeleton (at different levels of detail), they differ in the way that the outer shape or flesh and skin of the human is modeled. Volumetric outer models try to approximate body parts with geometric primitives such as ellipsoids [168], cylindrical shapes [161, 82] (Figure 2.7b) or truncated cones [46], superquadrics or superellipsoids [79] (Figure 2.7c). In

contrast to these implicit surfaces, other models use explicit surfaces that are provided by polygonal meshes [121, 55]. Such surfaces can be derived from body scans of the subject (*e.g.* through laser scans) and often seem more realistic than their implicit counterparts. Plänkers and Fua [112] and Horaud *et al.* [68] use smooth implicit surfaces that provide a level of realism that is comparable to explicit surfaces through blending of several implicit geometric primitives (Figure 2.7d). Such models can additionally be augmented with explicit surfaces for fine details [112]. Anguelov *et al.* [7] have introduced the SCAPE model [102, 13, 66] (Figure 2.7e) whose surface parameters have been learned from 3D laser scans of several human subjects in varying poses. The resulting surface mesh is deformable with respect to body size and posture, and enables accurate display of human shapes up to the level of muscle contractions. SCAPE is not associated with an inner kinematic structure, although extensions have been presented to estimate the underlying skeleton [6]. The impressive realism of the model comes at increased computational expense, which makes it difficult to apply SCAPE-like models for tracking tasks. The 3D human models typically used in markerless tracking approaches range from around 24 d.o.f. [79] to 34 d.o.f. [46].

To cope with the high dimensionality of the state space for human poses and the resulting difficulties in estimating these parameters, many approaches make use of machine learning techniques to solve the task. In the simplest case one can learn strong motion priors for specific motions to limit the search space when tracking [14]. Some approaches project training motions from the high-dimensional state space to a low-dimensional manifold, using techniques such as *principal component analysis* [158] or *Gaussian process dynamical models* [157, 163]. The low-dimensional manifold can then be used in a generative manner to create predictions for the next states. Taylor *et al.* [148] learn *binary latent variable models* that can be used for the same purpose.

Another class of exemplar-based methods [63, 3, 156, 26, 28] learns a direct mapping between observed image features such as silhouettes or *SIFT* features [91] and the corresponding human poses, which makes them easy to implement and to apply. Such methods are constricted to the type of features they have been trained with and training has to be redone when transferring to environments with different camera perspectives. Elgammal and Lee [50] learn a mapping from extracted features to a low-dimensional pose manifold that is then used to interpolate the high-dimensional 3D pose.

An advantage of exemplar-based techniques is that pose estimation can often be implemented efficiently once the training set has been processed. When implemented with care, such methods can reliably detect poses that correspond to known motions. The downside is that they are not capable to generalize over motions that are missing in the training set. Al-

though recent advances have made it possible to train discriminative methods on larger training sets [156, 26], the generation of all possible exemplar postures necessary to detect arbitrary motions remains intractable. Bourdev and Malik [28] present an interesting approach that is able to generalize to poses outside of the training set by training a multitude of poselet classifiers for subparts of the body that can be stitched together to retrieve previously unobserved poses. The tradeoff for the increased generality is the high computational cost associated with evaluating all poselet classifiers.

Vondrak *et al*. [161] use physics-based motion priors to constrain the rigid-body motion of body parts based on forces exhibited by human motor control and by environmental contacts.

Fully integrated markerless and thus unintrusive systems are emerging only slowly in real world applications, due to the immense algorithmic and practical challenges associated with these methods. The ORGANIC MOTION STAGE [142] system is the only serious competitor on the market to date. Similar to optical marker-based systems it uses a large array of cameras to capture human motions inside a capture volume. It requires an empty room with an uncluttered and distinct background to compute the visual hull, from which the kinematic parameters are then extracted. Most markerless systems presented in the scientific community have been evaluated only on short sequences and lack the final prove of robustness and applicability in practical scenarios. The PFINDER system by Wren *et al*. [168] uses a statistical blob-based model to keep track of human body parts in real time. It has been applied in a variety of contexts including gesture recognition, telepresence applications and ubiquitous environments, however the level of accuracy of the system is insufficient for motion analysis or animation tasks. Knoop *et al*. [82] developed the VOODOO system that uses a cylindrical body model to track articulated human motion in realtime from a mobile robot. Ramanan *et al*. [120] have successfully applied their method for monocular human pose detection using planar 2D models on extended sequences including a full feature-length movie.

A more comprehensive summary of related work on markerless human motion capture is provided by the surveys from Gavrila [59], Moeslund *et al*. [100, 101] and Poppe [114].

### 2.2.3 Human Activity Recognition

Action recognition or interpretation of human motions aims at assigning higher-level semantic labels to human motion patterns. This facilitates the recognition of ongoing activities and intentions when combined with context information. Video-based action recognition can be roughly divided into two research directions. The first class of methods provides a direct mapping of image cues to action classes without the intermediate estimation of model parameters. An alternative class of methods is model-based, where parameters of a representative model

are determined in a first step, and then used to perform action classification in phase space.

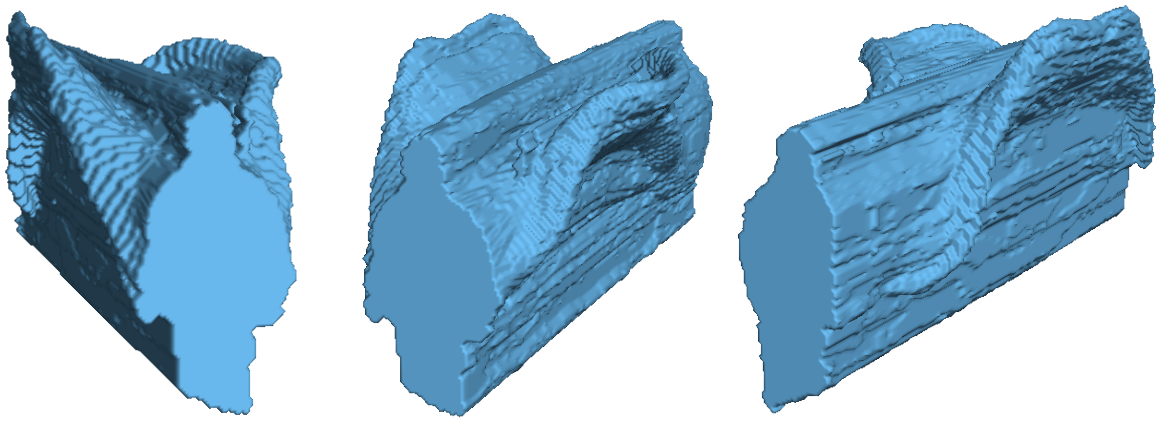

FIGURE 2.8 Space-time-shapes for action recognition. Human silhouette shapes extracted from video sequences are stacked in temporal direction to create a 3D shape. Geometry features extracted from this shape are then used as input for classification. Such holistic approaches to activity recognition do not extract pose parameters using human motion capture methods, but instead use simple features that are extracted directly from the image stream. However, this makes them viewpoint-dependant and subtle activities are difficult to distinguish. Images are taken from our own work on holistic activity recognition [125, 126].

Model-free methods can be divided into global/holistic approaches and patch-based approaches. Holistic approaches consider image information as a whole, *e.g.* by using silhouettes [27], edge images [164] or optical flow fields [48]. Bobick *et al.* [27] have introduced *motion-history-images* as a 2.5D representation of actions, where silhouettes are overlaid on a single image with brightness corresponding to the distance in time. Blank *et al.* [25, 62] create *space-time-shapes* from extracted silhouettes by using time as the third dimension (Figure 2.8). A similar technique was independently proposed by Yilmaz and Shah [170] as *spatiotemporal volumes*. Weinland *et al.* [166] create view-independent representations of human motions by fusing silhouettes from multiple cameras to obtain visual hulls, that are then augmented with temporal information to form 3.5D *motion-history-volumes* (extending *motion-history-images* [27] by another dimension). In contrast to holistic approaches, patch-based approaches extract only selected salient features and group them for classification, which helps to overcome problems in the case of partial occlusions [107].

In model-based approaches, the most commonly used parameters for action classification are joint angles or joint positions of articulated human models [133, 88]. Such parameters are invariant to translations, scale and rotations, and form a rich and comprehensive representation of human movements. They are well suited for recognition and provide high recognition rates especially for more subtle motions. However, the model parameters are difficult to extract by

unintrusive means, so most of the methods rely on commercial marker-based motion capture systems for retrieving the joint parameters. Markerless motion capture systems [46, 121, 79, 68] provide an alternative, but are yet computationally demanding and lack robustness and general applicability.

Besides the feature extraction step, classification plays an important role. Here, an action label or a probability distribution over multiple labels is associated with the observed features. *Nearest neighbor* classification is probably the simplest method for selecting class membership. Sometimes, *dynamic time warping* is used to achieve invariance with respect to the speed of the motion [160]. Better recognition rates and a better generalization can be achieved by using discriminative classifiers such as *support vector machines* (SVM) [128]. Another class of methods is modeling an action in state space as a set of states and transitions. *Hidden markov models* (HMM) *e.g.* can be used for recognition as well as for generation of actions [85, 92, 110, 36]. Closely related are probabilistic grammars [98]. Ogale *et al.* [106] use context-free grammars to describe actions as a sequence of keyframe silhouettes. Such generative models are usually learned separately for all action classes. Sminchisescu *et al.* [140] use *conditional random fields* (CRF) as discriminative graphical models to achieve better separation of action classes.

More detailed surveys on human action recognition are provided by Krüger *et al.* [86] and by Poppe [115].

# CHAPTER 3

# Anthropometric Human Model

## Biomechanical Representation of Human Shape, Posture and Motion

Digital human models play an important role when observing and interpreting the motions of human subjects. In particular, most approaches to human pose estimation are model-based. Several types of models have been proposed over the years to capture the kinematic structure of the human body (*i.e*. the skeleton) as well as its geometry and appearance (*i.e*. the flesh and skin). Such models are able to provide a mathematical abstraction of human postures and appearance as a compact set of parameters to the model. They are often used to hypothesize the visual appearance of a specific human subject given a predicted pose, and to compare the rendered (or synthesized) hypothesis with the current image observation (*analysis by synthesis*). Advanced models can also be used to restrict the space of valid pose parameters respectively the configuration of body parts to meaningful and physiologically realistic postures.

Good human models should be able to accurately represent the shape and appearance of human subjects based on the subjects posture or motion. Depending on the application, the representation should also be compact and if possible intuitive. Therefore, the design or choice of a suitable model is a tradeoff between accuracy and computational simplicity. It is furthermore determined by the task at hand, *e.g*. 2D models might be better suited for monocular tracking, but they will not reach the realism achievable by more sophisticated 3D models.

There is a large body of work on human models that can be classified into 2D and 3D models. 2D models typically consist of image patches for each body part that are connected at the joints (Figure 2.7a). Such simple models are commonly encountered in monocular pose estimation tasks as a consequence of the missing depth information. Body patches are parameterized by in-plane rotations and a scale factor to model the distance from the camera (such models are known as *scaled prismatic models* [38, 165]). *Cardboard models* [75, 67] are extensions with additional parameters for 3D motions to model perspective deformations of the body patches. *Pictorial structures* [52, 120] are related graphical models where the spatial

arrangement of parts as well as their appearance is parameterized by probability distributions. They are often used in a bottom-up manner, *i.e*. body parts are first localized independently, and the detections are then combined to form the most probable consistent posture [120].

3D models provide a more faithful and accurate reproduction of humans. They are usually composed as a tree-like kinematic structure (*i.e*. skeletons) with associated geometry (*i.e*. skin and flesh). Volumetric models use geometric primitives to implicitly describe the geometry of body parts using just a few parameters. Commonly used primitives are ellipsoids [168], cylinders (Figure 2.7b), truncated cones [46], or superellipsoids [79] (Figure 2.7c). A more accurate outer appearance can be achieved by blending of geometric primitives [112, 68] (Figure 2.7d) or by specifying explicit surfaces as polygonal meshes [121, 55]. The SCAPE model [7, 102, 13, 66] (Figure 2.7e) is a precise 3D model whose surface mesh is deformable both with respect to body size and posture. Its surface parameters have been learned from a multitude of 3D laser scans of human subjects in varying poses, enabling accurate geometric display of humans up to the level of muscle contractions. However, such precisely deformable models come at increased computational expense which makes them difficult to apply for tracking tasks.

In our work we take the new approach to integrate the digital human model RAMSIS [129, 33, 130] for tracking of human motions (Figure 3.1). RAMSIS is an advanced and industry-proven model from the ergonomics community, that is widely-used especially in the automotive community [34]. It was initially developed to ease CAD-based design of car interior and human workspaces, as well as for use in ergonomic studies. The following advantages come with the use of this model:

- The model is capable of capturing different body types according to anthropometric considerations, *i.e*. the different appearance of a wide range of humans. Its design has been guided by ergonomic considerations from leading experts in the field.
- The locations of the inner joints correspond precisely to the real human joint locations, making this model an ideal choice for motion analysis tasks *e.g*. in sport analytics or ergonomic studies.
- The model is able to capture most of the movements humans can perform while retaining a correct outer appearance. Absolute motion limits are integrated and help to reduce the search space when tracking. Motion limits can be queried for different percentiles of the population using anthropometric knowledge.

In the next sections, we will describe the digital human model RAMSIS in more detail. We will first describe the inner kinematic model and the corresponding outer surface repre-

sentation in Section 3.1. Both inner and outer model are adaptable to different body shapes, however body adaptation in the original model is a tedious task due to a complex parameterization of inner and outer appearance of the model. We have improved on this by learning a small relevant subset of statistical parameters from available model instantiations by means of *principal component analysis* (PCA). This process is detailed in Section 3.2. We introduce several optimizations to the original RAMSIS model that are tailored towards the special needs of motion tracking applications in Section 3.3. Among these are a reduction of the d.o.f. of the original model by providing an ergonomically sound interpolation of the spine, the incorporation of biomechanical inter-frame motion limits, and the caching of pose-specific calculations that allows for efficient repeated computations of locally restricted pose changes. Finally, in Section 3.4 we show how a temporal sequence of poses can be filtered to produce smooth natural motions with good accuracy.

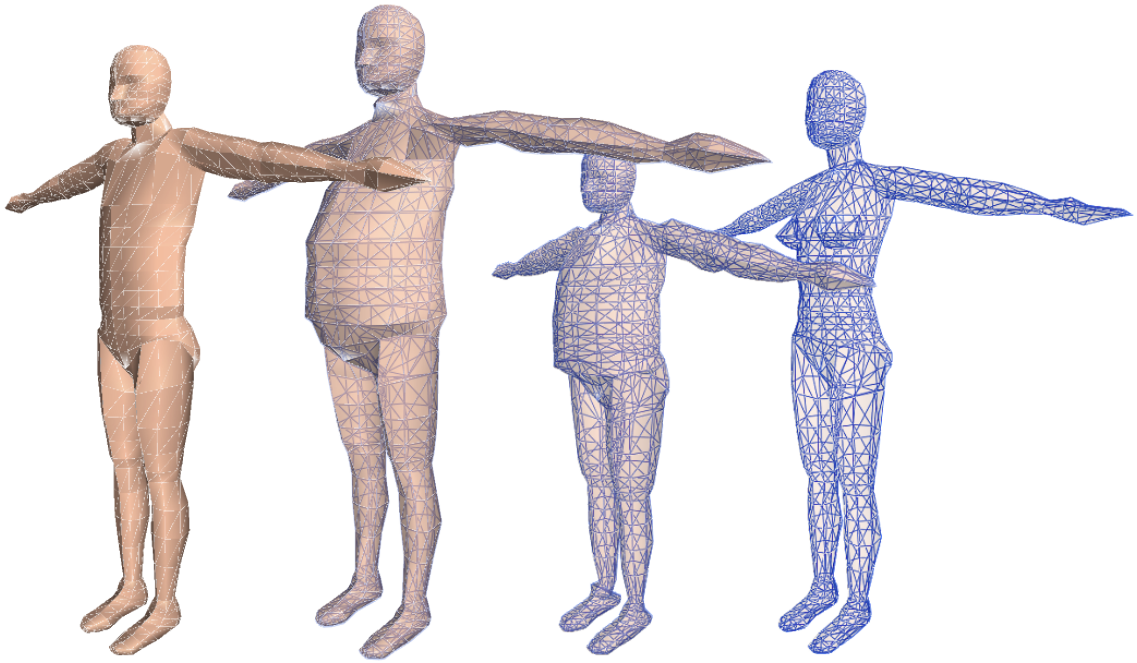

FIGURE 3.1 The digital human model RAMSIS. Shown are several examples of the outer model for different body shapes and gender. The surface is modeled as a triangle mesh that is connected to an underlying kinematic structure.

## 3.1 Model Specification

The digital human model RAMSIS has originally been designed by experts in the field of ergonomics [129] in order to provide a human model that can be used for analysis, synthesis and simulation of three-dimensional human postures. At the same time, it should be capable of

capturing the body characteristics of the largest part of the human population. To this end, the model design has been heavily influenced by anthropometric reference literature and human expertise. The model is in widespread use in the automotive and textile industry [61], where several (commercial) tools and extensions have been developed.

Figure 3.1 shows examples of different model instantiations in their default pose (the *null pose*). The main model definition comprises aspects such as the hierarchy of body parts, the relative positions of body joints and surface vertices in a corresponding local part coordinate system, and absolute motion limits for body parts. All corresponding parameters are fixed and have been specified during the initial development of the RAMSIS model. In addition, variable parameters describe the pose (dependant on the underlying kinematic structure) and the shape of the model (based on metric length values that provide an absolute scale for the relative positions in the model definition). We will now discuss the kinematic structure and the shape representation in more detail.

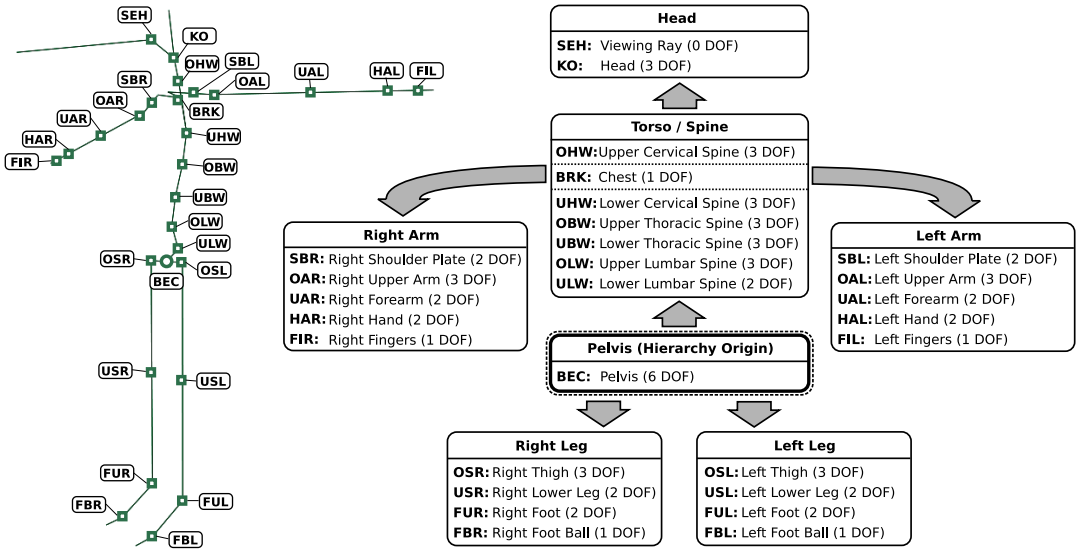


FIGURE 3.2 Inner model of the digital human model RAMSIS. The joint locations with abbreviations are shown on the left, the hierarchical structure of the model including the d.o.f. per body part are shown on the right. The hierarchical origin is the pelvis. The inner model is displayed in its *null pose* (see Figure 3.1 for corresponding outer models).

### 3.1.1 Kinematics

The inner model of RAMSIS is a kinematic tree structure that closely resembles the human skeleton (Figure 3.2). Deviations from the true skeleton are present due to the restriction that the tree nodes are ball joints with pure rotational transformations. The human shoulder *e.g.* is approximated by two ball joints (shoulder and shoulder blade), but the true structure

is more complex. Furthermore, only the most relevant joints are modeled, *e.g.* the spine is approximated by $6$ joints. The complete model is constructed from a total of $28$ joints.

Each body part $\mathtt{bp}$ is associated with a local part coordinate system (PCS) that can be fully described by a homogeneous $4{\times}4$ transformation matrix $\mathbf{H}_{\mathtt{bp}}$. The column vectors of the upper-left $3{\times}3$ rotation matrix embedded in $\mathbf{H}_{\mathtt{bp}}$ correspond to the axes of the local PCS while the first three components of the fourth column vector correspond to its origin in the world reference frame. Note that we use $\mathtt{bp}$ as a placeholder for any valid body part (see Figure 3.2 for all body parts and the corresponding abbreviations). Furthermore, we use the notations $\overleftarrow{\mathtt{bp}}$ to denote the hierarchical predecessor and $\overrightarrow{\mathtt{bp}}$ to denote the hierarchical successor of a body part $\mathtt{bp}$. The recursive formulation of the kinematic chain used to compute the local PCS is as follows:

$$\mathbf{H}_{\mathtt{BEC}} = \mathbf{T}_{\mathtt{BEC}} \cdot \mathbf{R}_{\mathtt{BEC}} \tag{3.1}$$

$$\mathbf{H}_{\mathtt{bp}} = \mathbf{H}_{\overleftarrow{\mathtt{bp}}} \cdot \mathbf{T}_{\mathtt{bp}} \cdot \mathbf{P}'_{\mathtt{bp}} \cdot \mathbf{R}_{\mathtt{bp}} \cdot \mathbf{P}''_{\mathtt{bp}} \tag{3.2}$$

Equation 3.1 corresponds to the initial $6$ d.o.f. pose of the pelvis (*i.e.* the hierarchical origin) in coordinates of the world reference frame. Based on this, all subsequent body part coordinate systems $\mathbf{H}_{\mathtt{bp}}$ are computed using forward kinematics. When reading the chain of transformations in Equation 3.2 from left to right, $\mathbf{H}_{\mathtt{bp}}$ is computed by first translating the preceding part coordinate system $\mathbf{H}_{\overleftarrow{\mathtt{bp}}}$ with the translation matrix $\mathbf{T}_{\mathtt{bp}}$ and then changing its orientation by applying the rotation matrix $\mathbf{R}_{\mathtt{bp}}$. The rotation is furthermore enclosed by two permutation matrices $\mathbf{P}'_{\mathtt{bp}}$ and $\mathbf{P}''_{\mathtt{bp}}$ that serve the purpose of switching coordinate axes while maintaining a right-handed coordinate system. The permutation matrices correspond to the identity matrix $\mathbf{I}$ in most cases, with the exceptions being denoted in Table 3.1 and Equations 3.11 and 3.12. The justification for these (mathematically unnecessary) permutations is that they allow the rotation parameters to conform with the notations prevalent in the anthropometrics and ergonomics literature.

By convention, 3D rotations are expressed using Euler angles, and the rotational axes are denoted as $t$ (tangential), $n$ (normal) and $b$ (binormal), with the following order of rotation:

$$\mathbf{R} = \mathbf{R}_b \cdot \mathbf{R}_n \cdot \mathbf{R}_t \tag{3.3}$$

While the $t$-axis is always oriented towards the next body part in the kinematic chain, the $n$-axis is facing towards the frontal surface of the current body part and the $b$-axis is the cross product of $t$ and $n$ (Figure 3.3). This conformity leads to a user-friendly parameterization that is easier to use in biomechanical or ergonomic applications such as gait or motion analysis.

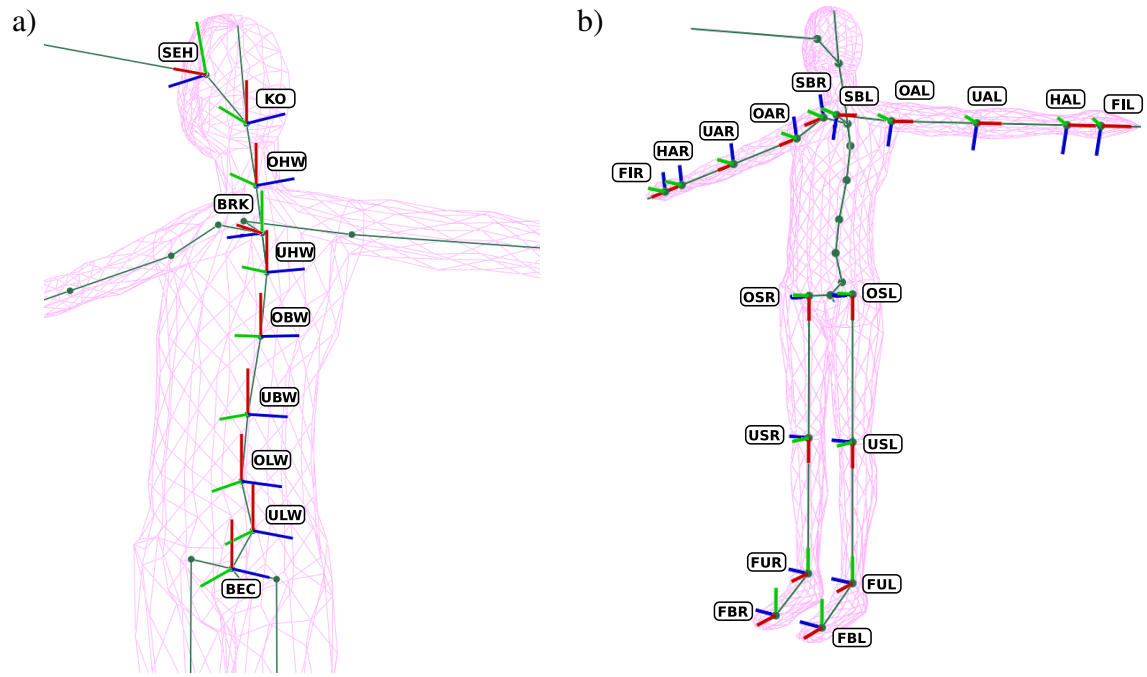


FIGURE 3.3 Orientations of the local part coordinate systems for the null pose of the human model RAMSIS: a) part coordinate systems of the spine b) part coordinate systems of the limbs. The $t$-axis (red) is always oriented towards the next body part in the kinematic chain. The $n$-axis (green) is facing towards the frontal surface of the body part. The $b$-axis (blue) is the cross product of $t$ and $n$.

The three rotation matrices $\mathbf{R}_t$, $\mathbf{R}_n$, $\mathbf{R}_b$ and the translation matrix $\mathbf{T}$ are computed from the Euler rotation vector $\mathbf{r} = (rt, rn, rb)^{\mathrm{T}}$ respectively the translation vector $\mathbf{t} = (tx, ty, tz)^{\mathrm{T}}$ as follows:

$$\mathbf{R}_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos rt & \sin rt & 0 \\ 0 & -\sin rt & \cos rt & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \qquad \mathbf{R}_n = \begin{pmatrix} \cos rn & 0 & -\sin rn & 0 \\ 0 & 1 & 0 & 0 \\ \sin rn & 0 & \cos rn & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.4}$$

$$\mathbf{R}_b = \begin{pmatrix} \cos rb & \sin rb & 0 & 0 \\ -\sin rb & \cos rb & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \qquad \mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.5}$$

The final rotation vector $\mathbf{r}_{\mathrm{bp}}$ used in the computation of the rotation matrices for each body

part $\mathtt{bp}$ is created by adding a base rotation vector $\tilde{\mathbf{r}}_{\mathtt{bp}}$ to the pose parameter vector $\mathbf{r}'_{\mathtt{bp}}$:

$$\mathbf{r}_{\mathtt{bp}} = \mathbf{r}'_{\mathtt{bp}} + \tilde{\mathbf{r}}_{\mathtt{bp}} \tag{3.6}$$

While being mathematically unnecessary, this is again due to the conformity with the anthropometric literature to provide a nice *null pose*, *i.e.* when the pose parameters for all body parts are zero ($\mathbf{r}'_{\mathtt{bp}} = (0, 0, 0)^{\mathrm{T}}$). Such a pose corresponds to standing upright with arms stretched to the side, as displayed in Figures 3.1 and 3.2. The basic orientations of all local part coordinate systems in the null pose under consideration of the base rotations $\tilde{\mathbf{r}}_{\mathtt{bp}}$ and the permutations $\mathbf{P}'_{\mathtt{bp}}$ and $\mathbf{P}''_{\mathtt{bp}}$ are depicted in Figure 3.3.

Note that by using homogeneous coordinates, the chain of transformations can be accumulated in a single homogeneous $4 \times 4$ transformation matrix as presented in Equations 3.1 and 3.2. A point in local part coordinates (*e.g.* a vertex on the outer surface mesh) is then easily transformed into world coordinates by right-multiplying its homogeneous vector representation $\mathbf{p}_{\mathtt{bp}} = (x_{\mathtt{bp}},\, y_{\mathtt{bp}},\, z_{\mathtt{bp}},\, 1)^{\mathrm{T}}$ with the transformation matrix $\mathbf{H}_{\mathtt{bp}}$:

$$\mathbf{p}_w = \mathbf{H}_{\mathtt{bp}} \cdot \mathbf{p}_{\mathtt{bp}} \tag{3.7}$$

This yields the corresponding homogeneous vector representation $\mathbf{p}_w = (x_w,\, y_w,\, z_w,\, 1)^{\mathrm{T}}$ of the point in world coordinates.

Based on the kinematic structure presented so far, the RAMSIS model can be set to a specific posture by providing the pose parameter vector $\boldsymbol{\psi}$. It consist of the 6 d.o.f. pose of the pelvis in the world reference frame (as specified in Equation 3.1, with translation vector $\mathbf{t}_{\mathtt{BEC}}$ and orientation vector $\mathbf{r}'_{\mathtt{BEC}}$) and the body part rotations $({\mathbf{r}'_{\mathtt{ULW}}}^{\mathrm{T}}, \ldots, {\mathbf{r}'_{\mathtt{FIR}}}^{\mathrm{T}})^{\mathrm{T}}$ of all remaining joints:

$$\boldsymbol{\psi} = \left({\mathbf{t}_{\mathtt{BEC}}}^{\mathrm{T}}, {\mathbf{r}'_{\mathtt{BEC}}}^{\mathrm{T}}, {\mathbf{r}'_{\mathtt{ULW}}}^{\mathrm{T}}, \ldots, {\mathbf{r}'_{\mathtt{FIR}}}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{3.8}$$

Note that the translation vectors $\mathbf{t}_{\mathtt{bp}}$ needed for Equation 3.2 are not part of the pose parameters $\boldsymbol{\psi}$ as they do not influence the posture explicitly (with the exception of $\mathbf{t}_{\mathtt{BEC}}$ which corresponds to the absolute pose in the world reference frame). However, there is an implicit dependency, as $\mathbf{t}_{\mathtt{bp}}$ is influenced by the inner shape parameters $\boldsymbol{\phi}_I$ (see Section 3.1.2 and Table 3.2), *i.e.* the location of body joints in a specific posture changes with the body shape. In practice, the shape parameters $\boldsymbol{\phi}$ and thus also $\mathbf{t}_{\mathtt{bp}}$ stay constant once they are initialized to match a specific subject.

Not all of the joints of a human body allow for unrestricted movement. To account for this, the possible range of motion has been carefully designed under consideration of physiological

limits, and some of the rotation vectors $\mathbf{r}'_{\mathrm{bp}}$ have been restricted to less than 3 d.o.f. Therefore, the final RAMSIS model has a total of 63 d.o.f.

In addition to removing some of the d.o.f. completely (*e.g.* elbow rotations around its $b$-axis), motion limits have been introduced to limit the possible range of motion for each d.o.f. These motion limits correspond to the minimal and maximal angular rotations $\mathbf{rmin}_{\mathrm{bp}}$ and $\mathbf{rmax}_{\mathrm{bp}}$ that the pose parameters $\mathbf{r}'_{\mathrm{bp}}$ can take for each body part $\mathtt{bp}$.

$$\mathbf{rmin}_{\mathrm{bp}} = (rtmin_{\mathrm{bp}},\, rnmin_{\mathrm{bp}},\, rbmin_{\mathrm{bp}})^{\mathrm{T}} \tag{3.9}$$

$$\mathbf{rmax}_{\mathrm{bp}} = (rtmax_{\mathrm{bp}},\, rnmax_{\mathrm{bp}},\, rbmax_{\mathrm{bp}})^{\mathrm{T}} \tag{3.10}$$

Table 3.1 summarizes the kinematic structure of the RAMSIS model. The six different types of permutation matrices used in the model specification are defined as follows:

$$\mathbf{P}_{yx\bar{z}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};\quad \mathbf{P}_{\bar{x}y\bar{z}} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};\quad \mathbf{P}_{y\bar{z}\bar{x}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.11}$$

$$\mathbf{P}_{yzx} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};\quad \mathbf{P}_{xz\bar{y}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};\quad \mathbf{P}_{x\bar{z}y} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.12}$$

Note that in an efficient implementation of the RAMSIS model, one might omit the use of these permutation matrices as in Equation 3.2 and instead transform the pose parameters $\boldsymbol{\psi}$ accordingly where needed.

Another thing to mention is that the use of Euler angles imposes certain limitations such as non-linearities and singularities (gimbal lock) that arise in the context of animation [134], and non-uniform sampling that arises in the context of pose estimation [87]. Quaternions might provide a better solution in these cases. On the other hand, Euler angles provide a user-friendly and descriptive representation that is consistent with the anthropometrics literature, and absolute motion limits can be more easily integrated than in alternative representations. In our work we stick to the Euler representation for the sake of compatibility with the original RAMSIS model.

| bp | $\overleftarrow{\text{bp}}$ | $\overrightarrow{\text{bp}}$ | $\tilde{\mathbf{r}}_{\text{bp}}$ (in deg) | $\mathbf{rmin}_{\text{bp}}$ (in deg) | $\mathbf{rmax}_{\text{bp}}$ (in deg) | $\mathbf{P}'_{\text{bp}}$ | $\mathbf{P}''_{\text{bp}}$ |
|---|---|---|---|---|---|---|---|
| BEC | | ULW, OSL, OSR | $(0, 0, 0)^{\mathrm{T}}$ | $(-180, -180, -180)^{\mathrm{T}}$ | $(180, 180, 180)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| ULW | BEC | OLW | $(0, 0, 0)^{\mathrm{T}}$ | $(0, -12, -25)^{\mathrm{T}}$ | $(0, 12, 11)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| OLW | ULW | UBW | $(0, 0, 0)^{\mathrm{T}}$ | $(-12, -18, -24)^{\mathrm{T}}$ | $(12, 18, 30)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| UBW | OLW | OBW | $(0, 0, 0)^{\mathrm{T}}$ | $(-12, -18, -12)^{\mathrm{T}}$ | $(12, 18, 20)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| OBW | UBW | UHW | $(0, 0, 0)^{\mathrm{T}}$ | $(-24, -18, -22)^{\mathrm{T}}$ | $(24, 18, 36)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| UHW | OBW | BRK, OHW | $(0, 0, 0)^{\mathrm{T}}$ | $(-22, -22, -32)^{\mathrm{T}}$ | $(22, 22, 28)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| BRK | UHW | SBL, SBR | $(0, 0, 0)^{\mathrm{T}}$ | $(0, 0, -1)^{\mathrm{T}}$ | $(0, 0, 1)^{\mathrm{T}}$ | $\mathbf{P}_{yx\overline{z}}$ | $\mathbf{I}$ |
| OHW | UHW | UBW | $(0, 0, 0)^{\mathrm{T}}$ | $(-30, -22, -36)^{\mathrm{T}}$ | $(30, 22, 32)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| KO | OHW | UBW | $(0, 0, 0)^{\mathrm{T}}$ | $(-42, -16, -22)^{\mathrm{T}}$ | $(42, 16, 25)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| SEH | KO | | $(0, 0, -17.5)^{\mathrm{T}}$ | $(0, 0, 0)^{\mathrm{T}}$ | $(0, 0, 0)^{\mathrm{T}}$ | $\mathbf{P}_{yx\overline{z}}$ | $\mathbf{I}$ |
| OSL | BEC | USL | $(0, 0, 0)^{\mathrm{T}}$ | $(-73, -50, -30)^{\mathrm{T}}$ | $(51, 55, 134)^{\mathrm{T}}$ | $\mathbf{P}_{\overline{x}y\overline{z}}$ | $\mathbf{I}$ |
| USL | OSL | FUL | $(0, 0, 0)^{\mathrm{T}}$ | $(-63, -10, 0)^{\mathrm{T}}$ | $(55, 160, 0)^{\mathrm{T}}$ | $\mathbf{P}_{xz\overline{y}}$ | $\mathbf{P}_{x\overline{z}y}$ |
| FUL | USL | FBL | $(0, 0, 90)^{\mathrm{T}}$ | $(-35, 0, -58)^{\mathrm{T}}$ | $(39, 0, 47)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| FBL | FUL | | $(0, 0, 0)^{\mathrm{T}}$ | $(0, 0, -45)^{\mathrm{T}}$ | $(0, 0, 80)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| OSR | BEC | USR | $(0, 0, 0)^{\mathrm{T}}$ | $(-51, -55, -30)^{\mathrm{T}}$ | $(73, 50, 134)^{\mathrm{T}}$ | $\mathbf{P}_{\overline{x}y\overline{z}}$ | $\mathbf{I}$ |
| USR | OSR | FUR | $(0, 0, 0)^{\mathrm{T}}$ | $(-55, -10, 0)^{\mathrm{T}}$ | $(63, 160, 0)^{\mathrm{T}}$ | $\mathbf{P}_{xz\overline{y}}$ | $\mathbf{P}_{x\overline{z}y}$ |
| FUR | USR | FBR | $(0, 0, 90)^{\mathrm{T}}$ | $(-39, 0, -58)^{\mathrm{T}}$ | $(35, 0, 47)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| FBR | FUR | | $(0, 0, 0)^{\mathrm{T}}$ | $(0, 0, -45)^{\mathrm{T}}$ | $(0, 0, 80)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| SBL | BRK | OAL | $(0, -10, 0)^{\mathrm{T}}$ | $(-29, -30, -18)^{\mathrm{T}}$ | $(14, 24, 41)^{\mathrm{T}}$ | $\mathbf{P}_{y\overline{z}\overline{x}}$ | $\mathbf{I}$ |
| OAL | SBL | UAL | $(0, 0, 0)^{\mathrm{T}}$ | $(-95, -89, -118)^{\mathrm{T}}$ | $(95, 89, 150)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| UAL | OAL | HAL | $(0, 0, 0)^{\mathrm{T}}$ | $(-149, -158, 0)^{\mathrm{T}}$ | $(116, 10, 0)^{\mathrm{T}}$ | $\mathbf{P}_{xz\overline{y}}$ | $\mathbf{P}_{x\overline{z}y}$ |
| HAL | UAL | FIL | $(0, 0, 0)^{\mathrm{T}}$ | $(0, -120, -59)^{\mathrm{T}}$ | $(0, 110, 42)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| FIL | HAL | | $(0, 0, 0)^{\mathrm{T}}$ | $(0, -102, 0)^{\mathrm{T}}$ | $(0, 12, 0)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| SBR | BRK | OAR | $(0, 10, 0)^{\mathrm{T}}$ | $(-14, -24, -18)^{\mathrm{T}}$ | $(29, 30, 41)^{\mathrm{T}}$ | $\mathbf{P}_{yzx}$ | $\mathbf{I}$ |
| OAR | SBR | UAR | $(0, 0, 0)^{\mathrm{T}}$ | $(-95, -89, -118)^{\mathrm{T}}$ | $(95, 89, 150)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| UAR | OAR | HAR | $(0, 0, 0)^{\mathrm{T}}$ | $(-116, -158, 0)^{\mathrm{T}}$ | $(149, 10, 0)^{\mathrm{T}}$ | $\mathbf{P}_{xz\overline{y}}$ | $\mathbf{P}_{x\overline{z}y}$ |
| HAR | UAR | FIR | $(0, 0, 0)^{\mathrm{T}}$ | $(0, -120, -59)^{\mathrm{T}}$ | $(0, 110, 42)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |
| FIR | HAR | | $(0, 0, 0)^{\mathrm{T}}$ | $(0, -102, 0)^{\mathrm{T}}$ | $(0, 12, 0)^{\mathrm{T}}$ | $\mathbf{I}$ | $\mathbf{I}$ |

TABLE 3.1 The kinematic structure of the RAMSIS model. Denoted are the kinematic predecessor $\overleftarrow{\text{bp}}$ and the kinematic successors $\overrightarrow{\text{bp}}$ for each body part bp. In addition, the base rotations $\tilde{\mathbf{r}}_{\text{bp}}$ (Equation 3.6), the minimal and maximal angular pose limits $\mathbf{rmin}_{\text{bp}}$ and $\mathbf{rmax}_{\text{bp}}$, and the permutation matrices $\mathbf{P}'_{\text{bp}}$ and $\mathbf{P}''_{\text{bp}}$ according to Equation 3.2 are given. The corresponding non-identity permutation matrices are listed in Equations 3.11 and 3.12.

### 3.1.2 Shape Representation

The pose parameters $\boldsymbol{\psi}$ presented in the last section are used to set the posture of the model by means of forward kinematics. Apart from that, another set of free parameters is used to adapt

the shape and size of the human model. We denote these as the shape parameters $\phi$:

$$\phi = \left(\phi_I{}^{\mathrm{T}}, \phi_O{}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{3.13}$$

$$\phi_I = \left(l_1, \ldots, l_{43}\right)^{\mathrm{T}} \tag{3.14}$$

$$\phi_O = \left(\mathbf{s}_1{}^{\mathrm{T}}, \ldots, \mathbf{s}_{100}{}^{\mathrm{T}}\right)^{\mathrm{T}}; \qquad \mathbf{s}_i = \left(t_i^+, t_i^-, n_i^+, n_i^-, b_i^+, b_i^-\right)^{\mathrm{T}} \tag{3.15}$$

While the pose parameters need to be updated in every frame during tracking or animation of the model, the shape parameters are usually initialized only once for the subject of interest. They are split into 43 inner shape parameters $\phi_I$ (Equation 3.14) and 600 outer shape parameters $\phi_O$ (Equation 3.15), summing up to a total of 643 parameters.

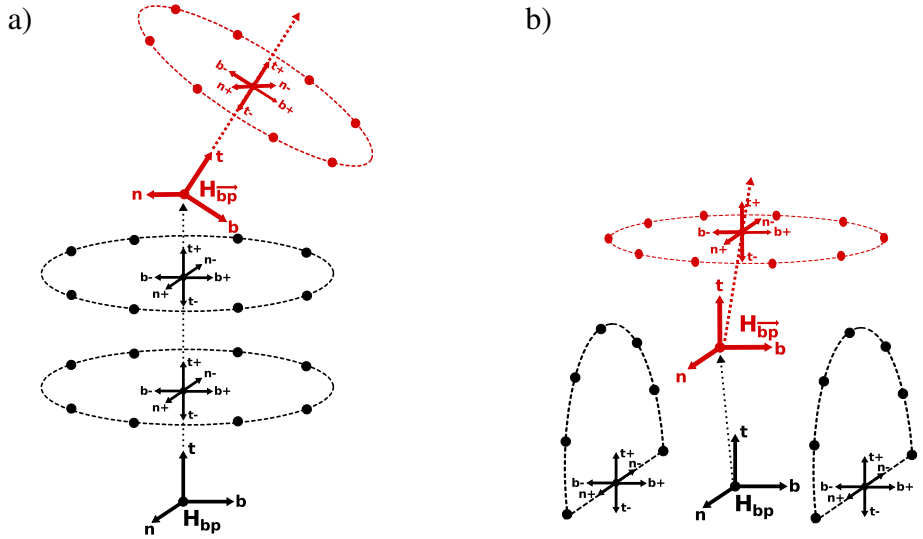


FIGURE 3.4 Local coordinate systems and shape representation in the RAMSIS model. Each body part bp is associated with a local PCS $\mathbf{H}_{\mathrm{bp}}$. The location of the next PCS $\mathbf{H}_{\overrightarrow{\mathrm{bp}}}$ (*i.e.* the next body joint) is dependant on the inner shape parameters $\phi_I$. The outer shape is represented by a connected set of vertices that are placed in slice coordinate systems (SCS) local to each PCS. The absolute positions of vertices inside each SCS are influenced by the outer shape parameters $\phi_O$, that provide a scale factor for each of the six direction $\pm t$, $\pm n$ and $\pm b$ of the SCS. Figure a) In most of the limb parts, the origins of the SCS and the subsequent PCS coincide with the $t$-axis, and the slices are planar and perpendicular to it. Figure b) In other body parts (*e.g.* around the neck), the origins are placed arbitrarily inside the current body part and the slices can take any (possibly non-planar) shape. Note that the triangulation of the outer vertices is not displayed in these figures.

In the RAMSIS model definition, the locations of important model points (*i.e.* joints for the inner model, surface vertices for the outer model) are specified in terms of relative coordinates in local coordinate systems. The shape parameters $\phi$ provide the absolute scaling to these relative coordinates, *i.e.* when calculating the absolute positions of points in their corresponding

coordinate system, the relative positions are multiplied by the corresponding lengths set in the shape parameters. The inner shape parameters $\boldsymbol{\phi}_I$ directly influence the distances between adjacent body joints and thus the proportions of the human skeleton. The outer shape parameters $\boldsymbol{\phi}_O$ influence the location of vertices on the outer surface and thus the shape and volume of the body parts.

Figure 3.4 depicts in greater detail the internal model structure. Each body part $\mathtt{bp}$ is associated with a local part coordinate system (PCS) $\mathbf{H}_{\mathtt{bp}}$ (Equation 3.2). The origin of the next PCS $\mathbf{H}_{\overrightarrow{\mathtt{bp}}}$ is defined by its corresponding translation vector $\mathbf{t}_{\overrightarrow{\mathtt{bp}}}$ (see Equations 3.2 and 3.5). All translation vectors and therefore the absolute proportions of the inner model (*i.e.* the bone lengths) are dependant on the inner shape parameters $\boldsymbol{\phi}_I$. These dependencies are denoted in Table 3.2. In addition, we have denoted the mean anthropometric length values for the inner shape parameters in Table 3.3. The outer shape is represented by a surface mesh that connects a set of vertices on the *human skin*. These vertices are placed on slices anchored in slice coordinate systems (SCS) that are local to each PCS. The origins of the SCS are directly influenced by the inner shape parameters $\boldsymbol{\phi}_I$, in the same way that the next PCS is. The absolute positions of the outer vertices inside each SCS are influenced by the outer shape parameters $\boldsymbol{\phi}_O$ (Equation 3.15). These consist of six scaling parameters $\mathbf{s}_i = (t_i^+, t_i^-, n_i^+, n_i^-, b_i^+, b_i^-)^{\mathrm{T}}$ for each slice $i$ that define the absolute scales of the corresponding slice along the negative and positive directions of the SCS axes.

The final surface mesh is provided by a fixed triangulation of the outer vertices. In total, the RAMSIS model definition comprises 36 inner joint positions (including body part endpoints), 962 outer vertices, and 1908 triangles on the surface mesh. When changing the pose of the model, the vertices move rigidly with the corresponding body part. The only exception is given by torsion deformations for the limbs and also parts of the spine (ULW, OLW, UBW, OBW, OSL, USL, OSR, USR, OAL, UAL, HAL, OAR, UAR, HAR). In these parts, the outer vertices all lie on slices that are perpendicular to the $t$-axis, *i.e.* the tangential direction towards the next body part (Figure 3.4a). To simulate the skin deformation of humans during torsions of the respective body parts, *i.e.* during rotations around the tangential axis, slices that are close to the next body part are rotated stronger around the tangential axis than slices that are close to the origin of the current body part coordinate system. This is achieved by scaling the tangential rotation component $rt_{\mathtt{bp}}$ of the rotation vector $\mathbf{r}_{\mathtt{bp}}$ (Equation 3.4) with a factor $\tau_i \in [0:1]$ depending on the relative location of the slice $i$ along the tangential axis (from 0 for the current part origin to 1 for the next part origin):

$$rt_{\mathtt{bp}i} = \tau_i \cdot rt_{\mathtt{bp}} \quad ; \qquad \tau_i \in [0{:}1] \tag{3.16}$$

| bp | $\overleftarrow{\text{bp}}$ | $\mathbf{t}_{\text{bp}}$ (in mm) |
|---|---|---|
| ULW | BEC | $(l_1,\ -l_2,\ 0.0)^{\mathrm{T}}$ |
| OLW | ULW | $(l_6,\ 0.37{\cdot}l_6,\ 0.0)^{\mathrm{T}}$ |
| UBW | OLW | $(l_7,\ -0.17{\cdot}l_7,\ 0.0)^{\mathrm{T}}$ |
| OBW | UBW | $(l_8,\ -0.3{\cdot}l_8,\ 0.0)^{\mathrm{T}}$ |
| UHW | OBW | $(l_9,\ -0.18{\cdot}l_9,\ 0.0)^{\mathrm{T}}$ |
| BRK | UHW | $(l_{11},\ 0.23{\cdot}l_{11},\ 0.0)^{\mathrm{T}}$ |
| OHW | UHW | $(l_{10},\ 0.23{\cdot}l_{10},\ 0.0)^{\mathrm{T}}$ |
| KO | OHW | $(l_{15},\ 0.29{\cdot}l_{15},\ 0.0)^{\mathrm{T}}$ |
| SEH | KO | $(0.24{\cdot}l_{16},\ l_{17},\ 0.0)^{\mathrm{T}}$ |
| OSL | BEC | $(0.0,\ 0.0,\ 0.5{\cdot}l_3)^{\mathrm{T}}$ |
| USL | OSL | $(l_{20},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| FUL | USL | $(l_{21},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| FBL | FUL | $(l_{22},\ -l_{23},\ 0.0)^{\mathrm{T}}$ |
| OSR | BEC | $(0.0,\ 0.0,\ -0.5{\cdot}l_3)^{\mathrm{T}}$ |
| USR | OSR | $(l_{25},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| FUR | USR | $(l_{26},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| FBR | FUR | $(l_{27},\ -l_{28},\ 0.0)^{\mathrm{T}}$ |
| SBL | BRK | $(l_{12},\ 0.0,\ -0.5{\cdot}l_{14})^{\mathrm{T}}$ |

| bp | $\overleftarrow{\text{bp}}$ | $\mathbf{t}_{\text{bp}}$ (in mm) |
|---|---|---|
| OAL | SBL | $(l_{30},\ -l_{31},\ l_{32})^{\mathrm{T}}$ |
| UAL | OAL | $(l_{33},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| HAL | UAL | $(l_{34},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| FIL | HAL | $(l_{35},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| SBR | BRK | $(l_{12},\ 0.0,\ 0.5{\cdot}l_{14})^{\mathrm{T}}$ |
| OAR | SBR | $(l_{37},\ -l_{38},\ -l_{39})^{\mathrm{T}}$ |
| UAR | OAR | $(l_{40},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| HAR | UAR | $(l_{41},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| FIR | HAR | $(l_{42},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| BEC′ | BEC | $(-l_4,\ -l_5,\ 0.0)^{\mathrm{T}}$ |
| KO′ | KO | $(l_{16},\ 0.23{\cdot}l_{17},\ 0.0)^{\mathrm{T}}$ |
| KO″ | KO | $(0.24{\cdot}l_{16},\ l_{17},\ 0.5{\cdot}l_{18})^{\mathrm{T}}$ |
| KO‴ | KO | $(0.24{\cdot}l_{16},\ l_{17},\ -0.5{\cdot}l_{18})^{\mathrm{T}}$ |
| SEH′ | SEH | $(l_{19},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| FBL′ | FBL | $(l_{24},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| FBR′ | FBR | $(l_{29},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| FIL′ | FIL | $(l_{36},\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| FIR′ | FIR | $(l_{43},\ 0.0,\ 0.0)^{\mathrm{T}}$ |

**TABLE 3.2** Relative joint locations $\mathbf{t}_{\text{bp}}$ of the body part bp in coordinates of the preceding body part $\overleftarrow{\text{bp}}$. To account for different human body sizes, the joint locations are dependant on the inner shape parameters $\boldsymbol{\phi}_I = (l_1, \ldots, l_{43})^{\mathrm{T}}$ of the RAMSIS model (see Table 3.3 for the anthropometric mean lengths). The joint locations in the lower right column correspond to endpoints that do not have a local coordinate system of their own: pelvis reference point (BEC′), top of skull (KO′), left eye (KO″), right eye (KO‴), focal direction endpoint (SEH′), left toes (FBL′), right toes (FBR′), left fingertips (FIL′), right fingertips (FIR′).

All vertices connected to the slice $i$ are then transformed by a transformation matrix $\mathbf{H}_{\text{bp}i}$ where $rt_{\text{bp}}$ has been replaced by $rt_{\text{bp}i}$ during its computation (Equation 3.2). This prevents the surface mesh from becoming twisted in such a strong way that intersections of the surface triangles occur.

The inner shape parameters $\boldsymbol{\phi}_I$ (Table 3.3) are most often directly related to values found in the anthropometrics literature (*e.g.* bone lengths, head height/breadth/depth, eye distance, *etc.*). Therefore, the inner model of RAMSIS can be described as an anthropometric model whose design has been entirely guided by the study of human anatomy and physical variation. This makes the RAMSIS model well-suited for biomechanical motion analysis and ergonomic

studies.

| | length (in mm) | | length (in mm) | | length (in mm) | | length (in mm) |
|---|---|---|---|---|---|---|---|
| $l_1$ | 42.00 | $l_{12}$ | 86.17 | $l_{23}$ | 72.00 | $l_{34}$ | 246.00 |
| $l_2$ | 58.00 | $l_{13}$ | 0.00 | $l_{24}$ | 70.00 | $l_{35}$ | 90.00 |
| $l_3$ | 162.00 | $l_{14}$ | 50.00 | $l_{25}$ | 429.00 | $l_{36}$ | 100.00 |
| $l_4$ | 19.90 | $l_{15}$ | 64.20 | $l_{26}$ | 422.00 | $l_{37}$ | 159.00 |
| $l_5$ | 18.80 | $l_{16}$ | 153.50 | $l_{27}$ | 142.00 | $l_{38}$ | 62.00 |
| $l_6$ | 83.00 | $l_{17}$ | 104.10 | $l_{28}$ | 72.00 | $l_{39}$ | 0.00 |
| $l_7$ | 100.00 | $l_{18}$ | 63.00 | $l_{29}$ | 70.00 | $l_{40}$ | 283.00 |
| $l_8$ | 123.00 | $l_{19}$ | 500.00 | $l_{30}$ | 159.00 | $l_{41}$ | 246.00 |
| $l_9$ | 108.00 | $l_{20}$ | 429.00 | $l_{31}$ | 62.00 | $l_{42}$ | 90.00 |
| $l_{10}$ | 129.79 | $l_{21}$ | 422.00 | $l_{32}$ | 0.00 | $l_{43}$ | 100.00 |
| $l_{11}$ | 61.00 | $l_{22}$ | 142.00 | $l_{33}$ | 283.00 | | |

TABLE 3.3 Anthropometric mean length values in millimeter for the inner shape parameters $\boldsymbol{\phi}_I = (l_1, \ldots, l_{43})^{\mathrm{T}}$ of the male RAMSIS model.

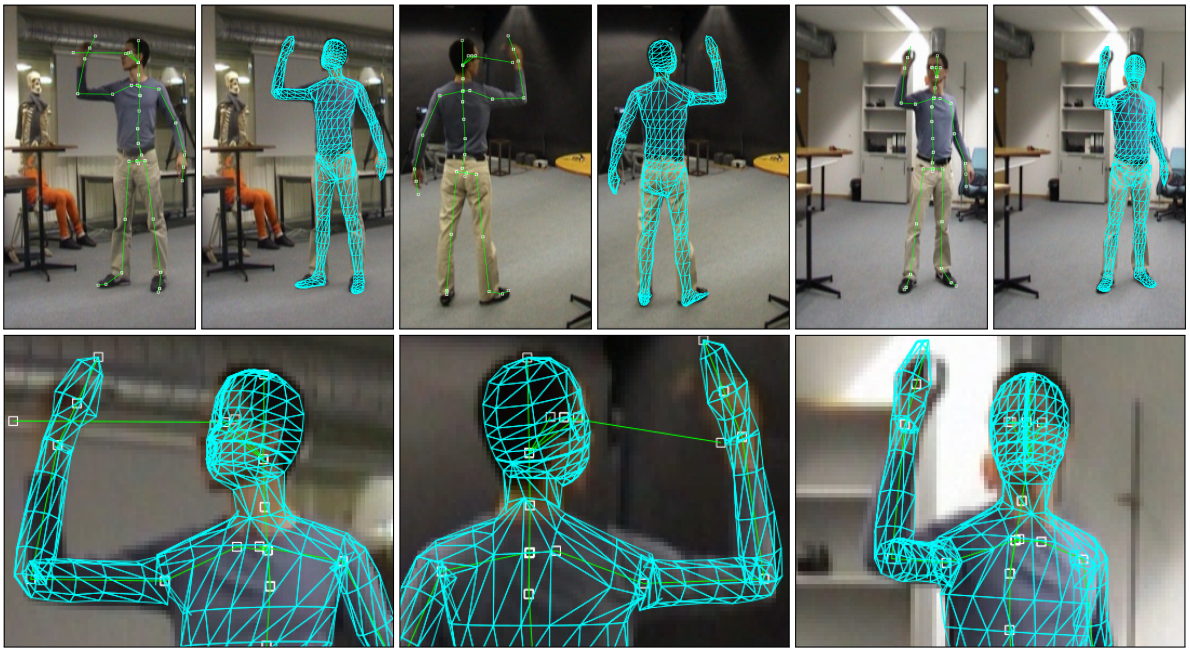

FIGURE 3.5 Single synchronized frame from three camera views to depict the accuracy of the human model RAMSIS. 2D image projections are shown for the inner model and the outer model (incl. zoomed image portion).

The design process of the outer model has been guided by manual expertise using anthropometric CAD models of humans. The main goal has been to provide a minimal representation

that is capable to capture the body shapes of a large variety of humans while retaining a realistic appearance throughout possible postures. Thus, the locations and connections of the outer vertices have been carefully selected. The model resolution used is a good compromise between accurate outer appearance and fast computations. Using a higher surface resolution would not improve the overall shape very much (except for better details), and using a lower resolution would result in unrealistic torsion deformations and problems with surface connections between neighboring body parts. Figure 3.5 shows an overlay of both the inner and the outer model on a still frame captured from three camera perspectives to illustrate the accuracy of the RAMSIS model.

The large number of shape parameters allows for a detailed adaptation of the model, but it is also overly complex and makes manual and automatic adaptation of the model tedious. In the next section we will present a reduced parameterization learned from training examples that improves usability and efficiency of the model shape initialization.

## 3.2 Learning Shape Parameters

In the original model specification of RAMSIS a total of $643$ shape parameters ($43$ inner and $600$ outer parameters) is provided. All of these parameters model very local changes to the shape, and especially the outer parameters $\phi_O$ influence only the placement of a few of the outer vertices. This over-parameterization makes it difficult to adapt the model shape to different humans in an intuitive manner. Furthermore, one would expect that shape variations among humans can be captured by far fewer parameters when the dependencies between the local shape parameters are considered.

To reduce the number of shape parameters, we have statistically analyzed the parameters by means of *principal component analysis* (PCA) [108]. We have taken $N = 139$ different (male) model adaptations for RAMSIS that cover a broad spectrum of human shapes (see Figure 3.1 for some examples). These have been retrieved in a manual modeling process using the GUI-tool PCMAN [130] from the TUM ergonomics department.

PCA is an orthogonal linear transformation that transforms the shape parameters of the known model adaptations into a new coordinate system such that the orthogonal basis of the new coordinate system is given by its *principal components*. These are the orthogonal directions that best capture the variance in the given shape parameters. In many applications, a subset of all principal components explains most of the variance in the data set, so that the remaining principal components can be ignored without loosing valuable information. This fact is often used to compress data by reducing the dimensionality of the data vectors. In our

application, we can use the principal components to provide an improved and possibly more intuitive parameterization of the human model shape, as dependencies between parameters are revealed by this process.

The principal components of the set of $N$ known shape parameters $\boldsymbol{\phi}^{(i)};\ i \in 1 \dots N$ of dimensionality $d$ are given by the *eigenvectors* of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$. The $d \times d$ covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ is computed as follows:

$$\boldsymbol{\Sigma}_{\boldsymbol{\phi}} = \frac{1}{N-1} \sum_{i=1}^{N} (\boldsymbol{\phi}^{(i)} - \bar{\boldsymbol{\phi}}) \cdot (\boldsymbol{\phi}^{(i)} - \bar{\boldsymbol{\phi}})^{\mathrm{T}} \tag{3.17}$$

Here, $\bar{\boldsymbol{\phi}}$ corresponds to the sample mean of all known shape parameters:

$$\bar{\boldsymbol{\phi}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\phi}^{(i)} \tag{3.18}$$

The eigenvectors can be extracted from the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ by means of *singular value decomposition* (SVD), which in the case of positive semi-definite symmetric matrices such as $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ results in the following decomposition:

$$\boldsymbol{\Sigma}_{\boldsymbol{\phi}} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^{\mathrm{T}} \tag{3.19}$$

The column vectors of $\mathbf{U} = (\mathbf{e}_1 \dots \mathbf{e}_d)$ correspond to the requested eigenvectors. Furthermore, $\boldsymbol{\Lambda}$ is a diagonal matrix with the corresponding eigenvalues $\lambda_i$ as diagonal entries.

The eigenvectors $\mathbf{e}_1 \dots \mathbf{e}_d$ correspond to the principal components and by convention are usually ordered in descending order with respect to the variance in that direction. When scaling all eigenvectors such that their length equals the squareroots $\sqrt{\lambda_i}$ of the corresponding eigenvalues, the endpoints of the scaled eigenvectors all lie on an isosurface with Mahalanobis distance $1$ of the Gaussian distribution induced by the known parameter sets. Thus, the eigenvectors with the largest corresponding eigenvalues are responsible for most of the body shape variation in the training set.

The orthogonal space spanned by the principal components provides an alternative parameterization of the body shape. By ignoring principle components with small corresponding eigenvectors, we can furthermore reduce the number of parameters by ignoring irrelevant ones. A common criterion is to set the number of parameters based on the percentage of variance in the training set that is explained by the first $x$ principal components. The variance in the direction of each principal component is given by the corresponding eigenvalue $\lambda_i$. Figure 3.6 plots the cumulative variance as a function of the first $x$ eigenvectors for our

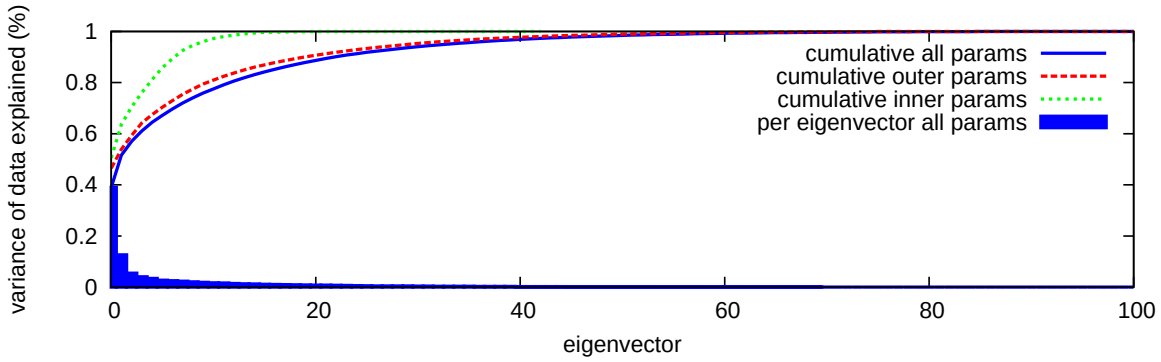


FIGURE 3.6 Plotting the variance accounted for by the principal components. The cumulative plots show the total percentage of the variance in the training set that is accounted for by the first $x$ eigenvectors when estimating all $643$ shape parameters, only the $600$ outer shape parameters, or only the $43$ inner shape parameters. Additionally, the variance for each eigenvector is plotted for the combination of inner and outer parameters.

application. When performing PCA on all $643$ parameters, the first $2$ eigenvectors already account for more than $50\%$, and the first $20$ eigenvectors for more than $88\%$ of the overall variance. These numbers improve further when trying to find a new parameterization for only the outer shape parameters (*e.g.* the body shape with constant bone lengths) or only the inner shape parameters (*e.g.* bone lengths). Learning new parameterizations for subsets of the shape parameters helps to find parameterizations that are more intuitive for a human user. When training only on the inner parameters corresponding to the human limbs (arms and legs), most of the variance will be explained by two principal components that correspond to limb length and relative ratio between upper and lower limb. These correspond exactly to the observations made in anthropometric literature. However, for applications such as automatic body shape initialization (*e.g.* by means of optimization), the intuitive meaning is irrelevant and the parameterization learned from all shape parameters should be used.

One thing to note is that the number of eigenvectors can never exceed the number of training samples. In our case, we have only $139$ training samples for $643$ dimensions, so that only the first $139$ eigenvectors will be non-zero. However, these eigenvectors still provide a suitable orthogonal basis for the parameter space, and as Figure $3.6$ shows, only a much smaller subset of the calculated eigenvectors is relevant anyway. In future work, we plan to repeat the statistical analysis with a much larger training set based on anthropometric databases.

Once a good new parameterization is found (*i.e.* the $x$ most relevant principal components have been selected), a vector $\phi$ of RAMSIS shape parameters can be projected to a lower-

dimensional vector $\boldsymbol{\varphi}$ in the new parameter space as follows:

$$\boldsymbol{\varphi} = \mathbf{W}^{\mathrm{T}} \cdot (\boldsymbol{\phi} - \bar{\boldsymbol{\phi}}) \tag{3.20}$$

Here, $\mathbf{W} = (\mathbf{e}_1 \dots \mathbf{e}_x)$ is a $d \times x$ matrix where the $x$ column vectors are exactly the principal components of the new parameter space, *i.e.* $\mathbf{W}$ corresponds to the eigenvector matrix $\mathbf{U}$ from Equation 3.19 with the less important eigenvectors $\mathbf{e}_{x+1} \dots \mathbf{e}_d$ cropped. Before transforming the parameter vector $\boldsymbol{\phi}$ it is mean-adjusted by subtracting the mean $\bar{\boldsymbol{\phi}}$ of the training set.

A more frequently needed transformation however is the transformation from the new parameter space back to the original parameter space of RAMSIS. To create a modification of the human shape, the components of the projection vector $\boldsymbol{\varphi}$ could *e.g.* be diffused along the corresponding principal components. The amount of diffusion added should be consistent with the variance along each principal component, *i.e.* its eigenvalue. The so-modified parameter vector $\boldsymbol{\varphi}'$ is then transformed back to calculate the parameter vector $\boldsymbol{\phi}'$ in the original parameter space:

$$\boldsymbol{\phi}' = (W\boldsymbol{\varphi}') + \bar{\boldsymbol{\phi}} \tag{3.21}$$

We have applied the concepts presented in this section to provide a GUI to aid with the manual initialization of the shape of the human model in an intuitive manner. Apart from a new parameterization of the male RAMSIS model, we have also computed parameters for the female RAMSIS model from 95 separate female adaptations. Splitting female and male parameterization is not only required by a slightly different model definition, but is also reasonable due to typical variations between the genders.

## 3.3 Model Optimizations for Tracking

The human model RAMSIS has been designed to measure human postures (and to mathematically describe them by the pose parameters $\boldsymbol{\psi}$) through manual adaptation. At the time of development and up to the start of this thesis, the applications were purely static in a sense that *e.g.* the posture taken by a human in a specific image should be described. If a sequence of postures was considered, all postures were adapted independently, without considering temporal dependencies. An exception is the work by Seitz [131], where tracking using the RAMSIS model was first applied in a restricted context, but without modifications to the original model. The typical requirements and limitations of tracking applications however made it necessary to introduce some modifications to the original RAMSIS model in order to successfully deal with the ambitious new objectives. We thus introduce the following modifications tailored

towards the tracking applications presented in Chapter 4:

1. Reduction of the d.o.f. of the initial model by modeling dependencies in the spine joints.

2. Incorporation of reasonable body-part dependant inter-frame motion variances to reduce the search space.

3. Caching of body part relative calculations for improved efficiency in tracking applications.

### 3.3.1 Coupling of Spinal Motion

As tracking of high-dimensional articulated models is a very challenging task, one of the first priorities in our work was to reduce the d.o.f. of the original model (Figure 3.2) as much as possible. While the limbs are already modeled with the minimal number of parameters necessary to express all possible poses, the largest potential for reduction is given in the model of the human spine. In RAMSIS, the spine is modeled by 7 independent joints (including the head) with 3 d.o.f. each. In the human anatomy however, the spine is a coupled structure that allows only for a restricted set of motions. It is *e.g.* impossible to bend the spine in several opposing directions, whereas this would be possible in the RAMSIS model. A solution to this problem is to couple dependant parameters from sets of neighboring joints to provide a common spinal motion. We have developed such a coupling in joint work with the TUM ergonomics department based on biomechanics literature [167, 78] and experimental studies.

To provide a physiologically sound coupling of the spine, we split it into the lower spine $\mathtt{LS} = \{\mathtt{ULW}, \mathtt{OLW}\}$, the upper spine $\mathtt{US} = \{\mathtt{UBW}, \mathtt{OBW}, \mathtt{UHW}\}$, and the head spine $\mathtt{HS} = \{\mathtt{OHW}, \mathtt{KO}\}$. Such a coupling is reasonable as most of the common motions are performed by deforming only the upper spine (*i.e.* the thoracic and the cervical spine), while the lower spine (*i.e.* the lumbar spine) is only used to support extreme bending motions once the deformation of the upper spine reaches its limits. Each of the coupled spine regions $\mathtt{xS}$ is associated with a 3 d.o.f. parameter vector $\mathbf{q}_{\mathtt{xS}} = (qt_{\mathtt{xS}}, qn_{\mathtt{xS}}, qb_{\mathtt{xS}})^{\mathrm{T}}$ with components in the range of $[0:1]$ that represent the amount of rotation around the $t$-, $n$- and $b$-axis. Rotations are distributed across the linked joints such that a value of $0$ corresponds to the minimum angular value $\mathbf{rmin}_{\mathtt{bp}}$ and $1$ corresponds to the maximum angular value $\mathbf{rmax}_{\mathtt{bp}}$ of each associated joint $\mathtt{bp}$, *i.e.* the coupled parameters jointly interpolate all linked angular parameters between their min/max-limits. In the null pose of the model, *i.e.* when all pose parameters are zero, the coupled spine parameters are all set to $0.5$. The angular pose parameters $({\mathbf{r}'_{\mathtt{ULW}}}^{\mathrm{T}}, \ldots, {\mathbf{r}'_{\mathtt{KO}}}^{\mathrm{T}})^{\mathrm{T}}$ for all of the

linked joints are calculated from the coupled parameters $(\mathbf{q}_{\text{LS}}{}^{\text{T}}, \mathbf{q}_{\text{US}}{}^{\text{T}}, \mathbf{q}_{\text{HS}}{}^{\text{T}})^{\text{T}}$ as follows:

$$\mathbf{r}'_{\text{ULW}} = \mathbf{q}_{\text{LS}} \cdot (\mathbf{rmax}_{\text{ULW}} - \mathbf{rmin}_{\text{ULW}}) + \mathbf{rmin}_{\text{ULW}} \quad (3.22)$$

$$\mathbf{r}'_{\text{OLW}} = \mathbf{q}_{\text{LS}} \cdot (\mathbf{rmax}_{\text{OLW}} - \mathbf{rmin}_{\text{OLW}}) + \mathbf{rmin}_{\text{OLW}} \quad (3.23)$$

$$\mathbf{r}'_{\text{UBW}} = \mathbf{q}_{\text{US}} \cdot (\mathbf{rmax}_{\text{UBW}} - \mathbf{rmin}_{\text{UBW}}) + \mathbf{rmin}_{\text{UBW}} \quad (3.24)$$

$$\mathbf{r}'_{\text{OBW}} = \mathbf{q}_{\text{US}} \cdot (\mathbf{rmax}_{\text{OBW}} - \mathbf{rmin}_{\text{OBW}}) + \mathbf{rmin}_{\text{OBW}} \quad (3.25)$$

$$\mathbf{r}'_{\text{UHW}} = \mathbf{q}_{\text{US}} \cdot (\mathbf{rmax}_{\text{UHW}} - \mathbf{rmin}_{\text{UHW}}) + \mathbf{rmin}_{\text{UHW}} \quad (3.26)$$

$$\mathbf{r}'_{\text{OHW}} = \mathbf{q}_{\text{HS}} \cdot (\mathbf{rmax}_{\text{OHW}} - \mathbf{rmin}_{\text{OHW}}) + \mathbf{rmin}_{\text{OHW}} \quad (3.27)$$

$$\mathbf{r}'_{\text{KO}} = \mathbf{q}_{\text{HS}} \cdot (\mathbf{rmax}_{\text{KO}} - \mathbf{rmin}_{\text{KO}}) + \mathbf{rmin}_{\text{KO}} \quad (3.28)$$

By using this parameterization we are able to reduce the dimensionality of the spine from 21 to 9 d.o.f. while keeping the physiological expressivity. Figure 3.7 shows some renderings of extreme spinal deformations that are possible with the RAMSIS model.

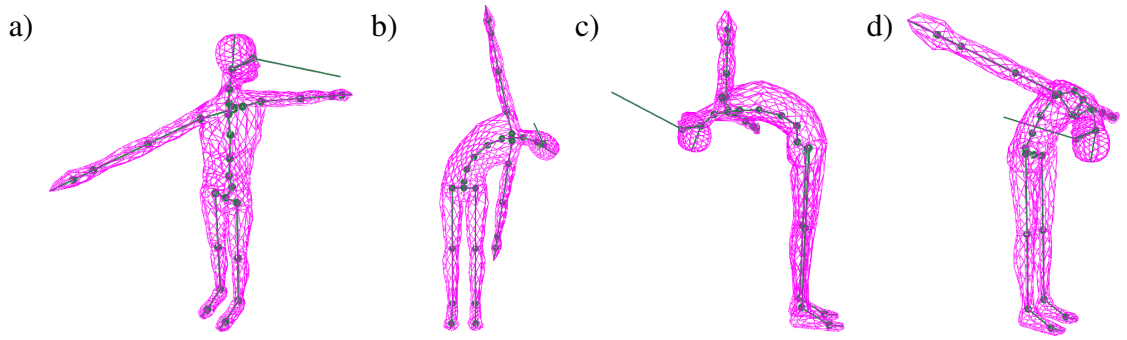


FIGURE 3.7 Motion limits of the combined spine when keeping the pelvis fixed: a) maximum rotation around the $t$-axis with $\mathbf{q}_{\text{LS}} = \mathbf{q}_{\text{US}} = \mathbf{q}_{\text{HS}} = (1.0, 0.5, 0.5)^{\text{T}}$ b) minimum rotation around the $n$-axis with $\mathbf{q}_{\text{LS}} = \mathbf{q}_{\text{US}} = \mathbf{q}_{\text{HS}} = (0.5, 0.0, 0.5)^{\text{T}}$ c) minimum rotation around the $b$-axis with $\mathbf{q}_{\text{LS}} = \mathbf{q}_{\text{US}} = \mathbf{q}_{\text{HS}} = (0.5, 0.5, 0.0)^{\text{T}}$ d) maximum rotation around the $b$-axis with $\mathbf{q}_{\text{LS}} = \mathbf{q}_{\text{US}} = \mathbf{q}_{\text{HS}} = (0.5, 0.5, 1.0)^{\text{T}}$. Bending can be further increased by rotating the pelvis.

When using the optimized model in tracking applications (see Chapter 4), the coupling of the spine considerably improves stability and reduces physiologically unfeasible postures (*e.g.* twisted and constricted deformations of the torso). Another step to simplify the tracking depending on the application is to ignore some of the d.o.f. in the hands and feet of the model, as these are often difficult to estimate from static wide-angle camera views. This further reduces the dimensionality of the model to 41. Figure 3.8 gives a graphical representation of the modified parameterizations as we use them for tracking.

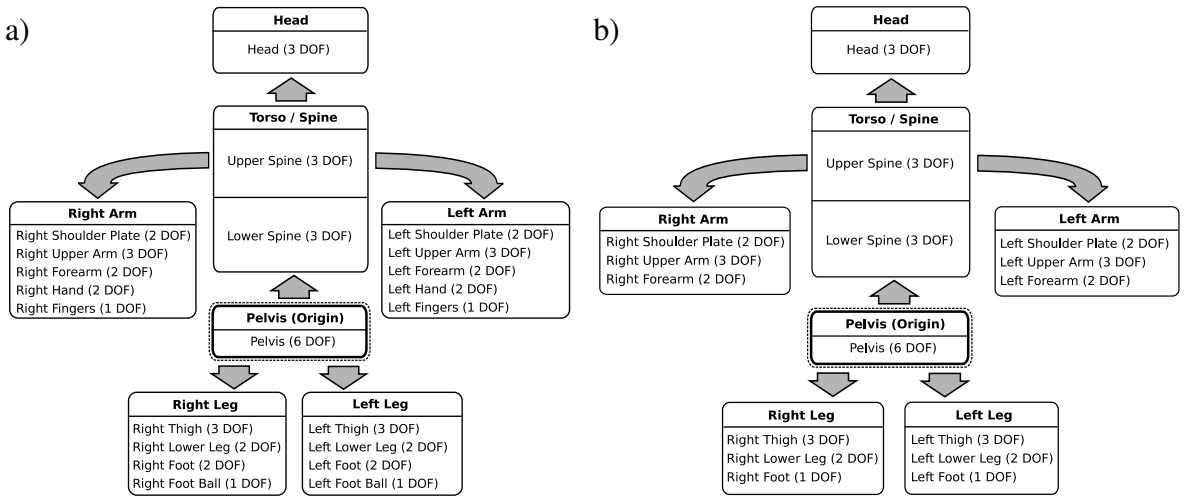


FIGURE 3.8 Modifications of the original RAMSIS model for tracking: a) after ergonomically sound interpolation of the spine (51 d.o.f.) b) after additionaly omitting some d.o.f. of the hands and feet that are hard to identify (41 d.o.f.). The latter model is the one we use for tracking throughout this thesis. The 3 d.o.f. of the lower spine are only relevant for extreme bending motions and can be omitted when constricting the tracker to upright motions (such as walking).

### 3.3.2 Biomechanical Inter-Frame Motion Limits

The original RAMSIS model comes with absolute motion limbs for each angular joint parameter (except for the unlimited rotation of the pelvis in 3D space). These motion limits specify the possible range of motion of the corresponding body part. However, there is no model for the temporal motion limits, *i.e*. the maximal angular velocities that can be reached during body movements. Good estimates of such values are important to constrain the high-dimensional search space when tracking.

Together with the TUM ergonomics department, we performed a biomechanical study to get estimates for the maximum angular velocities $\mathbf{vmax}_{\mathrm{bp}} = (vtmax_{\mathrm{bp}}, vnmax_{\mathrm{bp}}, vbmax_{\mathrm{bp}})^{\mathrm{T}}$ for each body part $\mathrm{bp}$ [51]. In this study, a human subject performed predefined isolated motions (*i.e*. each d.o.f. separately) at maximum speed while being recorded with two high-speed video cameras at 250 f.p.s. The pose parameters $\boldsymbol{\psi}$ for two selected keyframes of the sequence were manually measured with the GUI-tool PCMAN [130]. The keyframes were selected such that the angular difference between them covers about $40\%$ of the possible range of motion (between $\mathbf{rmin}_{\mathrm{bp}}$ and $\mathbf{rmax}_{\mathrm{bp}}$), preferably around the center of the range of motion, where the expected velocities are highest. The velocity estimates are calculated from the measured angular differences and the number of frames in between the two keyframes. In addition, we used marker-based motion capture data from sports science to refine the estimates. While the

resulting values are by no means statistically representative, they provide a good estimate on the magnitude of possible velocities. The resulting maximum angular velocities $\mathbf{vmax}_{\mathtt{bp}}$ are presented in Table 3.4.

| bp | $\mathbf{vmax}_{\mathtt{bp}}$ (in $\frac{deg}{sec}$) |
|---|---|
| BEC | $(125.0,\ 35.0,\ 125.0)^{\mathrm{T}}$ |
| ULW | $(0.0,\ 32.5,\ 17.5)^{\mathrm{T}}$ |
| OLW | $(40.0,\ 7.5,\ 12.5)^{\mathrm{T}}$ |
| UBW | $(42.5,\ 40.0,\ 25.0)^{\mathrm{T}}$ |
| OBW | $(47.5,\ 12.5,\ 42.5)^{\mathrm{T}}$ |
| UHW | $(7.5,\ 17.5,\ 5.0)^{\mathrm{T}}$ |
| BRK | $(0.0,\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| OHW | $(100.0,\ 70.0,\ 180.0)^{\mathrm{T}}$ |
| KO | $(312.5,\ 15.0,\ 22.5)^{\mathrm{T}}$ |
| SEH | $(0.0,\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| OSL | $(505.0,\ 145.0,\ 342.5)^{\mathrm{T}}$ |
| USL | $(290.0,\ 280.0,\ 0.0)^{\mathrm{T}}$ |
| FUL | $(440.0,\ 0.0,\ 375.0)^{\mathrm{T}}$ |
| FBL | $(0.0,\ 0.0,\ 0.0)^{\mathrm{T}}$ |

| bp | $\mathbf{vmax}_{\mathtt{bp}}$ (in $\frac{deg}{sec}$) |
|---|---|
| OSR | $(505.0,\ 145.0,\ 342.5)^{\mathrm{T}}$ |
| USR | $(290.0,\ 280.0,\ 0.0)^{\mathrm{T}}$ |
| FUR | $(440.0,\ 0.0,\ 375.0)^{\mathrm{T}}$ |
| FBR | $(0.0,\ 0.0,\ 0.0)^{\mathrm{T}}$ |
| SBL | $(0.0,\ 135.0,\ 77.5)^{\mathrm{T}}$ |
| OAL | $(277.5,\ 240.0,\ 190.0)^{\mathrm{T}}$ |
| UAL | $(945.0,\ 475.0,\ 0.0)^{\mathrm{T}}$ |
| HAL | $(0.0,\ 650.0,\ 400.0)^{\mathrm{T}}$ |
| FIL | $(0.0,\ 795.0,\ 0.0)^{\mathrm{T}}$ |
| SBR | $(0.0,\ 135.0,\ 77.5)^{\mathrm{T}}$ |
| OAR | $(277.5,\ 240.0,\ 190.0)^{\mathrm{T}}$ |
| UAR | $(945.0,\ 475.0,\ 0.0)^{\mathrm{T}}$ |
| HAR | $(0.0,\ 650.0,\ 400.0)^{\mathrm{T}}$ |
| FIR | $(0.0,\ 795.0,\ 0.0)^{\mathrm{T}}$ |

TABLE 3.4 Maximum angular velocities $\mathbf{vmax}_{\mathtt{bp}}$ for each body part bp in degrees per second.

For tracking applications, *e.g.* when doing stochastic search, the maximum angular velocities can be transformed into angular inter-frame standard deviations by considering the f.p.s. at which the videos were recorded. It might be advisable to provide upper limits to inter-frame motions when tracking at low frame rates, in order to keep the search space restricted. When tracking extremely fast human motions, *e.g.* in sports analysis, we recommend to use higher frame rates for recording to reduce the angular differences between neighboring frames (and also to avoid image motion blur by reducing exposure times).

### 3.3.3 Caching of Body-Part Dependant Pose Calculations

Another optimization that helps to improve computational efficiency is to cache body-part dependant pose calculations. Such a modification can be powerful when used in combination with hierarchical particle filters such as *partitioned sampling* (Section 4.4.2). These algorithms repeatedly modify only parts of the parameter space for large numbers of parallel evaluations, and caching helps in reducing the number of required computational operations (about a factor of 4 in practice).

Caching works by keeping all previous pose calculations in memory, including local part

coordinate systems $\mathbf{H}_{\mathrm{bp}}$ and both 3D world coordinates of surface vertices and their 2D image plane projections. Whenever a new pose is provided for model synthesis, all body parts whose parameters are changed when compared to the last calculated pose are invalidated. In addition, all dependencies of these body parts are also invalidated, *i.e.* all parts with invalidated kinematic predecessors. A recalculation of invalidated part coordinate systems is then performed using the recursive formulation given by Equations 3.1 and 3.2. The recalculated part coordinate systems are then used to update corresponding vertex coordinates.

## 3.4 Temporal Smoothing of Postures

In many applications the use of human models goes beyond the analysis of static postures, as the motions that humans perform over time are of particular interest. Usually, human motion is expressed as a temporal sequence of postures of the model sampled at discrete points in time. When trying to estimate human motions from cameras, *e.g.* by the methods presented in Chapter 4 of this thesis, the human poses are estimated frame by frame to create a temporal sequence. However, the resulting motion sequence often appears to be shaky and unsteady, as the temporal smoothness of motions has not been considered during pose estimation. A common solution is to post-process the motion sequence by providing a temporal smoothing, *e.g.* by using weighted mean filtering on a temporal window around the current frame. In this way, outliers are suppressed and the overall motion appears to be smoother and more human-like.

When implementing temporal smoothing in a straightforward manner, one would directly smooth the individual components of the pose parameters $\boldsymbol{\psi}$ (see Equation 3.8). However, as $\boldsymbol{\psi}$ consists purely of Euler angles that are part of a kinematic chain (except for the initial translation $\mathbf{t}_{\mathrm{BEC}}$), doing so results in undesired behavior. This is explained by the kinematic dependencies, *i.e.* whenever a joint angle is changed it affects all subsequent body joints in the kinematic chain. Thus, changes accumulate over several hierarchies, and the final motion results appear to be wobbly, *i.e.* the external limbs can deviate substantially from their expected position.

The solution is not to smooth the parameter space directly, but instead to smooth the locations of the body joints in the 3D Euclidean world space. This can be achieved by calculating the local mean part coordinate systems (PCS) from a temporal neighborhood, *i.e.* their mean positions and orientations. Once all mean PCS have been computed, the pose parameters $\boldsymbol{\psi}$ are estimated such that the model fits the new PCS as good as possible.

The mean PCS $\bar{\mathbf{H}}_{\mathrm{bp}}$ is calculated from the set of temporally adjacent local PCS $\mathbf{H}_{\mathrm{bp}}^{(j)}\,;\ j \in$

$W_t$ that have been selected from a temporal window $W_t$ around the current frame $t$ (in our work we use a window size of $5$ frames). Given that the local coordinate systems $\mathbf{H}_{\text{bp}}$ correspond to Euclidean transformations (*i.e.* we have only used translations, rotations and reflections/permutations in the composition of $\mathbf{H}_{\text{bp}}$, see Equations 3.1 and 3.2), they can be decomposed into a $3\times3$ rotation matrix $\mathbf{R}^{\star}_{\text{bp}}$ and a $3\times1$ translation vector $\mathbf{t}^{\star}_{\text{bp}}$:

$$\mathbf{H}_{\text{bp}} = \begin{pmatrix} \mathbf{R}^{\star}_{\text{bp}} & \mathbf{t}^{\star}_{\text{bp}} \\ 0 & 1 \end{pmatrix} \tag{3.29}$$

The column vectors of $\mathbf{R}^{\star}_{\text{bp}}$ correspond to the three axes $\mathbf{x}_{\text{bp}}$, $\mathbf{y}_{\text{bp}}$ and $\mathbf{z}_{\text{bp}}$ of the corresponding coordinate system, and $\mathbf{t}^{\star}_{\text{bp}}$ defines its position in world coordinates. Calculating the mean PCS $\bar{\mathbf{H}}_{\text{bp}}$ is thus equivalent to composing it from the mean translation vector $\bar{\mathbf{t}}^{\star}_{\text{bp}}$ and the three mean (orthonormal) axes $\bar{\mathbf{x}}_{\text{bp}}$, $\bar{\mathbf{y}}_{\text{bp}}$ and $\bar{\mathbf{z}}_{\text{bp}}$:

$$\bar{\mathbf{t}}^{\star}_{\text{bp}} = \sum_{j\in W_t} \omega_j \, \mathbf{t}^{\star(j)}_{\text{bp}} \tag{3.30}$$

$$\bar{\mathbf{x}}_{\text{bp}} = \sum_{j\in W_t} \omega_j \, \mathbf{x}^{(j)}_{\text{bp}} \; ; \qquad \bar{\mathbf{y}}_{\text{bp}} = \sum_{j\in W_t} \omega_j \, \mathbf{y}^{(j)}_{\text{bp}} \; ; \qquad \bar{\mathbf{z}}_{\text{bp}} = \sum_{j\in W_t} \omega_j \, \mathbf{z}^{(j)}_{\text{bp}} \tag{3.31}$$

Here, $\omega_j$ corresponds to weights associated with each frame $j$ in the temporal window $W_t$. To provide a correct weighted mean, all weights should sum up to $1$. In our work we compute the weights based on a Gaussian bell curve centered around the current frame, to decrease the influence of frames that are more distant in time. Note that the mean coordinate axes need to be normalized such that they can be combined into a valid rotation matrix.

After having calculated the temporally smoothed local PCS $\bar{\mathbf{H}}_{\text{bp}}$, the pose parameters $\boldsymbol{\psi}$ need to be chosen such that the model fits them as good as possible. Unfortunately, smoothing the local PCS changes the distances between adjacent PCS. As the inner model sizes are supposed to be constant during pose estimation, we can only come up with approximate solutions to the fitting. Finding optimal pose parameters that minimize the error between the corresponding local PCS and the temporally smoothed PCS is a difficult high-dimensional optimization problem (with $51$ d.o.f. in our case).

As an alternative, we provide a recursive analytical approximation to the model fitting. The goal is to calculate a new set of PCS $\hat{\mathbf{H}}_{\text{bp}}$ that is in conformance with the inner shape parameters $\boldsymbol{\phi}_I$ of the model (*i.e.* $\mathbf{T}_{\text{bp}}$, $\mathbf{P}'_{\text{bp}}$ and $\mathbf{P}''_{\text{bp}}$ are left untouched), yet whose local orientations are similar to the temporally smoothed PCS $\bar{\mathbf{H}}_{\text{bp}}$. The resulting error will thus be purely translational (and rather small in practice). We replace the local PCS calculations from

Equations 3.1 and 3.2 with:

$$\hat{\mathbf{H}}_{\text{BEC}} = \bar{\mathbf{H}}_{\text{BEC}} \tag{3.32}$$

$$\hat{\mathbf{H}}_{\text{bp}} = \hat{\mathbf{H}}_{\overleftarrow{\text{bp}}} \cdot \mathbf{T}_{\text{bp}} \cdot \mathbf{P}'_{\text{bp}} \cdot \hat{\mathbf{R}}_{\text{bp}} \cdot \mathbf{P}''_{\text{bp}} \tag{3.33}$$

First, the initial 6 d.o.f. pose $\mathbf{H}_{\text{BEC}}$ of the pelvis is replaced with the temporally smoothed pose $\bar{\mathbf{H}}_{\text{BEC}}$. In the following calculations, $\mathbf{T}_{\text{bp}}$, $\mathbf{P}'_{\text{bp}}$ and $\mathbf{P}''_{\text{bp}}$ are fixed, as they are defined by our current model instantiation. However, $\hat{\mathbf{R}}_{\text{bp}}$ is unknown and needs to be estimated. It should be a pure rotation matrix from which the temporally smoothed joint angle vector $\hat{\mathbf{r}}_{\text{bp}}$ will be extracted.

As already mentioned, we want the rotation defined by $\hat{\mathbf{R}}_{\text{bp}}$ to be equivalent with the rotation component of the temporally smoothed PCS $\bar{\mathbf{H}}_{\text{bp}}$. By substituting $\hat{\mathbf{H}}_{\text{bp}}$ in Equation 3.33 for $\bar{\mathbf{H}}_{\text{bp}}$ we can solve for the transformation $\mathbf{Q}$ that will evolve the kinematic chain computed up to now (under consideration of all model constraints) into the desired smoothed PCS:

$$\bar{\mathbf{H}}_{\text{bp}} = \hat{\mathbf{H}}_{\overleftarrow{\text{bp}}} \cdot \mathbf{T}_{\text{bp}} \cdot \mathbf{P}'_{\text{bp}} \cdot \mathbf{Q} \cdot \mathbf{P}''_{\text{bp}} \tag{3.34}$$

$$\mathbf{Q} = {\mathbf{P}'_{\text{bp}}}^{-1} \cdot {\mathbf{T}_{\text{bp}}}^{-1} \cdot \hat{\mathbf{H}}_{\overleftarrow{\text{bp}}}^{-1} \cdot \bar{\mathbf{H}}_{\text{bp}} \cdot {\mathbf{P}''_{\text{bp}}}^{-1} \tag{3.35}$$

$\mathbf{Q}$ will be an Euclidean transformation matrix of the same structure as the one presented in Equation 3.29. By simply dropping the translational part of $\mathbf{Q}$ (*i.e.* replacing its components with zero entries), we get the requested rotation matrix $\hat{\mathbf{R}}_{\text{bp}}$ which we can then use in Equation 3.33 to calculate the temporally smoothed PCS $\hat{\mathbf{H}}_{\text{bp}}$ that is in accordance with the inner shape parameters $\phi_I$. The dropped translation part corresponds to the approximation error for the origin locations of the body joints. In practice, the error is very small and levels itself over several hierarchies.

Finally, we are interested in the temporally smoothed estimates for the joint angle rotation vectors $\hat{\mathbf{r}}_{\text{bp}}$. These can be extracted from the upper 3×3 rotation matrix $\hat{\mathbf{R}}^{\star}_{\text{bp}}$ embedded in $\hat{\mathbf{R}}_{\text{bp}}$. Under consideration of the Euler rotation convention of the RAMSIS model in Equation 3.3 and the single-axis rotation matrices in Equations 3.4 and 3.5, the connection between the joint angle vector $\hat{\mathbf{r}} = (rt, rn, rb)^{\text{T}}$ and the rotation matrix $\hat{\mathbf{R}}^{\star}$ is the following (we omit the

bp subscripts for notational clarity):

$$\begin{aligned}\hat{\mathbf{R}}^{\star} &= \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \hat{\mathbf{R}}_b^{\star} \cdot \hat{\mathbf{R}}_n^{\star} \cdot \hat{\mathbf{R}}_t^{\star} \\ &= \begin{pmatrix} \cos rn \cos rb & \sin rt \sin rn \cos rb - \cos rt \sin rb & \cos rt \sin rn \cos rb + \sin rt \sin rb \\ \cos rn \sin rb & \sin rt \sin rn \sin rb + \cos rt \cos rb & \cos rt \sin rn \sin rb - \sin rt \cos rb \\ -\sin rn & \sin rt \cos rn & \cos rt \cos rn \end{pmatrix} \end{aligned} \tag{3.36}$$

In the case that $\cos rn \neq 0$, the three Euler angles are computed as follows (note that this is one of two possible solutions for this case):

$$rn = -\arcsin(r_{31}) \tag{3.37}$$

$$rt = \operatorname{atan2}\left(\frac{r_{32}}{\cos rn}, \frac{r_{33}}{\cos rn}\right) \tag{3.38}$$

$$rb = \operatorname{atan2}\left(\frac{r_{21}}{\cos rn}, \frac{r_{11}}{\cos rn}\right) \tag{3.39}$$

Equations 3.37 to 3.39 cannot be used when $\cos rn = 0$, *i.e.* when $r_{31} = \pm 1$. In that case there is an infinite number of solutions, as $rt$ and $rb$ are linked (*gimbal lock*). The solution here is to use the following alternative computation:

$$rb = 0.0 \tag{3.40}$$

$$rn = \begin{cases} -0.5\,\pi & \text{if } r_{31} \approx 1 \\ 0.5\,\pi & \text{if } r_{31} \approx -1 \end{cases} \tag{3.41}$$

$$rt = \operatorname{atan2}(r_{12},\, r_{13}) \tag{3.42}$$

So far we have estimated the joint angles for the temporally smoothed rotation vectors $\hat{\mathbf{r}}_{\mathrm{bp}}$. In the last step, the base rotations from Equation 3.6 need to be subtracted to retrieve the pose parameter vectors $\hat{\mathbf{r}}'_{\mathrm{bp}}$:

$$\hat{\mathbf{r}}'_{\mathrm{bp}} = \hat{\mathbf{r}}_{\mathrm{bp}} - \tilde{\mathbf{r}}_{\mathrm{bp}} \tag{3.43}$$

A more detailed derivation of the extraction of Euler angles from rotation matrices is given in [138]. The extracted smoothed rotational pose parameter vectors $\hat{\mathbf{r}}'_{\mathrm{bp}}$ correspond to the components of the final pose parameter vector $\hat{\boldsymbol{\psi}} = \left(\bar{\mathbf{t}}^{\star\,\mathrm{T}}_{\mathtt{BEC}}, \bar{\mathbf{r}}'^{\,\mathrm{T}}_{\mathtt{BEC}}, \hat{\mathbf{r}}'^{\,\mathrm{T}}_{\mathtt{ULW}}, \ldots, \hat{\mathbf{r}}'^{\,\mathrm{T}}_{\mathtt{FIR}}\right)^{\mathrm{T}}$

## 3.5 Summary

In this chapter we have presented an anthropometric human model from the ergonomics community and introduced it in the context of human pose estimation. The original model has been designed to capture the body characteristics with respect to shape and posture for a large part of the human population, and was used mainly for CAD-based design of car interior and human workspaces under ergonomic aspects. We have taken this model and optimized it with respect to the requirements of motion tracking applications, a task it was not initially designated for. In other words, we have taken a digital human model that was designed to be used in a static context and extended its applicability to dynamic scenarios.

Our first contribution has been to learn a new parameterization for the shape of the model by means of *principal component analysis*. This helps to simplify the initial model adaptation and to provide a reduced set of parameters that is better suited for automated shape estimation. Then, we have shown how to reduce the dimensionality of the pose parameters from initially $65$ d.o.f. to now $51$ d.o.f. without loss of expressivity by providing a biomechanically inspired interpolation of the spine parameters. This is an important contribution, as the high dimensionality of the parameter space is one of largest obstacles in human pose estimation. Furthermore, we have determined biomechanical inter-frame motion limits for each parameter in terms of maximal angular velocities. These motion parameters are important when using the model to estimate motion sequences as they help to limit the search space during tracking. The reduced dimensionality of the model and the reduced search space due to the motion limits is also crucial for computational efficiency. In addition, we have been able to improve computational efficiency further by caching body-part dependant pose calculations, a technique that is particularly effective when used in combination with hierarchical sampling strategies as will be presented in the next chapter. Finally, we have shown a filtering method to create smooth and realistic motion sequences from a series of pose estimates that can be solved analytically.

All of these optimizations make the modified RAMSIS model a powerful tool in the context of human pose tracking. The model is able to provide a link between a mathematical parameterization of human poses and the corresponding visual appearance of human subjects. It turns out that accurate representation of human shape, posture and motion is a key aspect for reliable human pose estimation, as will be shown in the following chapters.

# CHAPTER 4

# Human Pose Tracking

## Iterative Hierarchical Sampling for High-Dimensional State Estimation

In the last chapter we have presented a sophisticated human model that is able to map human pose parameters to 3D surfaces approximating the shape of the corresponding human posture. The pose parameters define the position and orientation of the model in Euclidean 3D space and the orientation of each body joint in the articulated human model. In this chapter we will show how to estimate the pose parameters from video sequences of human motion. This is particularly challenging due to the high dimensionality of the parameter space (up to $51$ d.o.f.).

To alleviate the complexity of the task at hand, we state human pose estimation as a tracking problem, *i.e.* as one of estimating the current pose given a pose estimate from the last timestep. By formalizing the problem in a Bayesian framework, we take advantage of the power and advanced methodology of probabilistic techniques.

We will give an introduction to *Recursive Bayesian Estimation* in the next section, and present a collection of estimators commonly used for Bayesian tracking. Out of these estimators, the *Particle Filter* was selected as a basis for the advanced tracking algorithms presented later in this chapter due to its generality. An important choice in particle filtering is the selection of a motion model and an observation model. We describe our choice for the motion model in Section 4.2, followed by a description of our observation model in Section 4.3. In Section 4.4 we discuss the shortcomings of standard particle filtering techniques when used for human pose estimation, and motivate the need for hierarchical sampling strategies. Two strategies that are better suited for human pose estimation have been proposed, which we describe in Section 4.4.1 (*Annealed Particle Filter*) and Section 4.4.2 (*Partitioned Sampling*) respectively. Both of these approaches have several issues in practice, and we propose an improved sampling strategy in Section 4.5. We finish this chapter with a detailed experimental evaluation of the presented approaches (Section 4.6), showing results on simulated as well as real data, including a common ground-truth benchmark.

## 4.1 Recursive Bayesian Estimation

We are interested in tracking human fullbody motion over time. Given image observations $\mathbf{y}_{0:T} \triangleq \{\mathbf{y}_t\,;\, 0 \leq t \leq T\}$ of a human motion sequence of length $T$ sampled at discrete points in time, we can model the task in a Bayesian manner as a state-space approach for dynamic systems. The state sequence $\mathbf{x}_{0:T} \triangleq \{\mathbf{x}_t\,;\, 0 \leq t \leq T\}$ then corresponds to the human pose parameters that contain all relevant information to describe the human motion in the observed sequence. To simplify matters, the state sequence $\mathbf{x}_{0:T}$ is assumed to be an unobserved (hidden) Markov process, *i.e.* the probability of a state $\mathbf{x}_t$ is only dependent on its direct predecessor $\mathbf{x}_{t\text{-}1}$ and conditionally independent of earlier states:

$$p\big(\mathbf{x}_t \,\big|\, \mathbf{x}_{0:t\text{-}1}\big) = p\big(\mathbf{x}_t \,\big|\, \mathbf{x}_{t\text{-}1}\big) \tag{4.1}$$

Likewise, the observation $\mathbf{y}_t$ at timestep $t$ is dependent only upon the current state $\mathbf{x}_t$ and conditionally independent of all other states:

$$p\big(\mathbf{y}_t \,\big|\, \mathbf{x}_{0:t}\big) = p\big(\mathbf{y}_t \,\big|\, \mathbf{x}_t\big) \tag{4.2}$$

This corresponds to a hidden Markov model (HMM) as displayed in Figure 4.1. The joint probability density function (pdf) over all states of such a HMM is given by:

$$p(\mathbf{x}_{0:t}, \mathbf{y}_{0:t}) = p(\mathbf{x}_0) \prod_{i=1}^{t} p\big(\mathbf{y}_i \,\big|\, \mathbf{x}_i\big)\, p\big(\mathbf{x}_i \,\big|\, \mathbf{x}_{i\text{-}1}\big) \tag{4.3}$$

Equation 4.3 embodies all available statistical information, and in principle, it can be used to derive an optimal estimate with respect to any criterion including a measure of accuracy. Unfortunately, inference is intractable, and it requires to store the complete data set and to reprocess existing data when new observations arrive.

However, we are not interested in the full joint pdf over all states and observations, but rather in the current pdf of state $\mathbf{x}_t$ conditioned on the observations up to timestep $t$. This corresponds

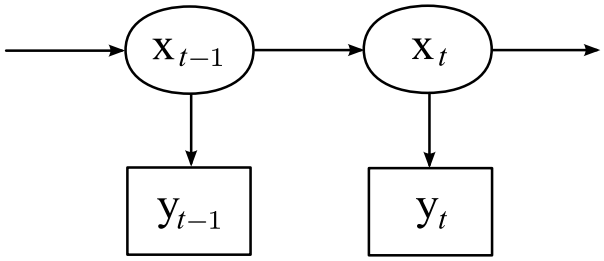


FIGURE 4.1 Bayesian Network of a Hidden Markov Model.

to *Bayesian filtering* [47], where the task is to estimate the *posterior* pdf $p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t})$. Under the assumption that the initial pdf $p(\mathbf{x}_0 \,|\, \mathbf{y}_0) \equiv p(\mathbf{x}_0)$ is known, the posterior pdf can be obtained recursively in a *prediction* and an *update* stage [47, 9].

The prediction stage uses the *motion model* $p(\mathbf{x}_t \,|\, \mathbf{x}_{t\text{-}1})$ to make a forward prediction of the posterior pdf $p(\mathbf{x}_{t\text{-}1} \,|\, \mathbf{y}_{0:t\text{-}1})$ from the last timestep to the current timestep, resulting in the *prior* pdf $p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t\text{-}1})$:

$$p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t\text{-}1}) = \int p(\mathbf{x}_t \,|\, \mathbf{x}_{t\text{-}1})\, p(\mathbf{x}_{t\text{-}1} \,|\, \mathbf{y}_{0:t\text{-}1})\, d\mathbf{x}_{t\text{-}1} \tag{4.4}$$

The motion model approximates the (often unknown) dynamics of the system, and must also account for unknown disturbances, *e.g.* by adding random noise. The prediction stage usually results in flattened modes of the pdf, which corresponds to an increase in uncertainty.

In the update stage, the latest observation $\mathbf{y}_t$ is incorporated into the prior pdf with the help of the observation model $p(\mathbf{y}_t \,|\, \mathbf{x}_t)$, resulting in the required *posterior* pdf $p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t})$:

$$p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t}) = \frac{p(\mathbf{y}_t \,|\, \mathbf{x}_t)\, p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t\text{-}1})}{p(\mathbf{y}_t \,|\, \mathbf{y}_{0:t\text{-}1})} = \eta\, p(\mathbf{y}_t \,|\, \mathbf{x}_t)\, p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t\text{-}1}) \tag{4.5}$$

$$= \eta\, p(\mathbf{y}_t \,|\, \mathbf{x}_t) \int p(\mathbf{x}_t \,|\, \mathbf{x}_{t\text{-}1})\, p(\mathbf{x}_{t\text{-}1} \,|\, \mathbf{y}_{0:t\text{-}1})\, d\mathbf{x}_{t\text{-}1} \tag{4.6}$$

Here, $\eta$ is a normalizing constant comprising the denominator

$$p(\mathbf{y}_t \,|\, \mathbf{y}_{0:t\text{-}1}) = \int p(\mathbf{y}_t \,|\, \mathbf{x}_t)\, p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t\text{-}1})\, d\mathbf{x}_t \tag{4.7}$$

and does not have to be explicitly evaluated in the context of tracking the most likely state.

The posterior pdf in Equation 4.5 is derived using Bayes' theorem, which can be seen as a mechanism for updating the knowledge about the current state in the light of additional information from incoming observation. It can be used to estimate the most likely state including its uncertainty. In general, however, it cannot be determined analytically. We will present some estimators that can be used for Bayesian filtering in the subsequent sections.

### 4.1.1 Kalman Filter

The *Kalman Filter* [77] provides an optimal solution to the Bayesian filtering problem under certain restrictive assumptions.

First, it assumes that the posterior pdf $p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t})$ at every timestep is Gaussian and hence

parameterized by a mean and a covariance matrix:

$$p\big(\mathbf{x}_t \,\big|\, \mathbf{y}_{0:t}\big) \sim \mathcal{N}(\boldsymbol{\mu}_t,\, \boldsymbol{\Sigma}_t) \tag{4.8}$$

Here, $\mathcal{N}(\boldsymbol{\mu}_t,\, \boldsymbol{\Sigma}_t)$ denotes a multivariate normal distribution with mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{\Sigma}_t$.

Second, it assumes that both state transitions and the observations are modeled linearly with added Gaussian noise (i.e. *linear Gaussian*):

$$\mathbf{x}_t = \mathbf{A}_t\, \mathbf{x}_{t\text{-}1} + \boldsymbol{\epsilon}_t \qquad ; \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0,\, \mathbf{Q}_t) \tag{4.9}$$

$$\mathbf{y}_t = \mathbf{C}_t\, \mathbf{x}_t + \boldsymbol{\delta}_t \qquad ; \quad \boldsymbol{\delta}_t \sim \mathcal{N}(0,\, \mathbf{R}_t) \tag{4.10}$$

Here, $\mathbf{A}_t$ is the state transition matrix that defines a linear transformation of the previous state and $\mathbf{C}_t$ is the observation matrix defining a linear relationship between states and measurements. The process noise $\boldsymbol{\epsilon}_t$ and the observation noise $\boldsymbol{\delta}_t$ correspond to added zero-mean Gaussian noise with covariance matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$ respectively.

The motion model and the observation model are thus given by a (possibly time-varying) normal distribution as follows:

$$p\big(\mathbf{x}_t \,\big|\, \mathbf{x}_{t\text{-}1}\big) \sim \mathcal{N}(\mathbf{A}_t\mathbf{x}_{t\text{-}1},\, \mathbf{Q}_t) \tag{4.11}$$

$$p\big(\mathbf{y}_t \,\big|\, \mathbf{x}_t\big) \sim \mathcal{N}(\mathbf{C}_t\mathbf{x}_t,\, \mathbf{R}_t) \tag{4.12}$$

Finally, the complete recursive formulation of the Kalman filter algorithm is shown in Algorithm 4.1 (see *e.g.* [152] for a complete derivation). It is distinguished between the *time update* step (predicting the state) and the *measurement* step (incorporating new measurement).

**Algorithm 4.1** Kalman Filter

**Require:** $\boldsymbol{\mu}_{t\text{-}1}, \boldsymbol{\Sigma}_{t\text{-}1}, \mathbf{y}_t$

/* time update: */

$\hat{\boldsymbol{\mu}}_t = \mathbf{A}_t\, \boldsymbol{\mu}_{t\text{-}1}$

$\hat{\boldsymbol{\Sigma}}_t = \mathbf{A}_t\, \boldsymbol{\Sigma}_{t\text{-}1}\, \mathbf{A}_t^{\mathrm{T}} + \mathbf{Q}_t$

/* measurement-update: */

$\mathbf{K}_t = \hat{\boldsymbol{\Sigma}}_t\, \mathbf{C}_t^{\mathrm{T}}\, (\mathbf{C}_t\, \hat{\boldsymbol{\Sigma}}_t\, \mathbf{C}_t^{\mathrm{T}} + \mathbf{R}_t)^{-1}$

$\boldsymbol{\mu}_t = \hat{\boldsymbol{\mu}}_t + \mathbf{K}_t\, (\mathbf{y}_t - \mathbf{C}_t\, \hat{\boldsymbol{\mu}}_t)$

$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\, \mathbf{C}_t)\, \hat{\boldsymbol{\Sigma}}_t$

**return** $\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t$

Given the restriction on linear motion and observation models, the Kalman filter has not been used successfully for articulated human pose estimation, as both the motion models and

the observation models are usually highly nonlinear [141]. Furthermore, the use of normally distributed posteriors poses the limitation of unimodal state estimates only. However, observation models for human pose estimation are often ambiguous, leading to multimodal posteriors. The behavior of the Kalman filter in this respect is undesirable, making it (almost) impossible to resolve ambiguities later on in the light of new observations.

### 4.1.2 Extended Kalman Filter

The *Extended Kalman Filter* (EKF) [73] relaxes some constraints of the linear Kalman filter, at the price of optimality. While still assuming that the resulting posterior pdf can be approximated by a mean and a covariance matrix, the motion model and the observation model no longer need to be linear functions of the state as long as they are differentiable (compare with Equations 4.9 and 4.10):

$$\mathbf{x}_t = g(\mathbf{x}_{t\text{-}1}) + \boldsymbol{\epsilon}_t \quad ; \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0,\, \mathbf{Q}_t) \tag{4.13}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \boldsymbol{\delta}_t \quad ; \quad \boldsymbol{\delta}_t \sim \mathcal{N}(0,\, \mathbf{R}_t) \tag{4.14}$$

The functions $g$ and $h$ are used to predict the new state or to compute the expected measurement from the predicted state. To be able to update the covariances, the Jacobians $\mathbf{G}$ of $g$ and $\mathbf{H}$ of $h$ (*i.e.* the matrices of partial derivatives) at the respective state estimates have to be computed. They can essentially be seen as locally linear approximations of the nonlinear functions $g$ and $h$. This results in a first-order Taylor expansion of the motion and observation models. The recursive estimation via the *time update* and *measurement update* in Algorithm 4.1 is then adapted as described in Algorithm 4.2.

**Algorithm 4.2** Extended Kalman Filter

**Require:** $\boldsymbol{\mu}_{t\text{-}1}, \boldsymbol{\Sigma}_{t\text{-}1}, \mathbf{y}_t, \mathbf{G}_t, \mathbf{H}_t$

/* time update: */

$\hat{\boldsymbol{\mu}}_t = g(\boldsymbol{\mu}_{t\text{-}1})$

$\hat{\boldsymbol{\Sigma}}_t = \mathbf{G}_t\, \boldsymbol{\Sigma}_{t\text{-}1}\, \mathbf{G}_t^{\mathrm{T}} + \mathbf{Q}_t$

/* measurement-update: */

$\mathbf{K}_t = \hat{\boldsymbol{\Sigma}}_t\, \mathbf{H}_t^{\mathrm{T}}\, (\mathbf{H}_t\, \hat{\boldsymbol{\Sigma}}_t\, \mathbf{H}_t^{\mathrm{T}} + \mathbf{R}_t)^{-1}$

$\boldsymbol{\mu}_t = \hat{\boldsymbol{\mu}}_t + \mathbf{K}_t\, (\mathbf{y}_t - h(\hat{\boldsymbol{\mu}}_t))$

$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\, \mathbf{H}_t)\, \hat{\boldsymbol{\Sigma}}_t$

**return** $\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t$

The EKF usually provides good results unless the nonlinear mappings $g$ and $h$ have high curvature near the state estimates and/or the estimates have a high uncertainty. In human

pose estimation, these requirements are often violated, partially because of the tendency of the observation model to "jump" between different ambiguous states (*e.g.* switching the legs all of a sudden). Furthermore, the restriction on unimodal posteriors still holds. Because of these limitations, the EKF is rarely used for human pose estimation. Wachter and Nagel [162] use an iterative EKF (IEKF) and a 10 d.o.f. model for monocular tracking of motions parallel to the image plane. The IEKF is also used by Mikic *et al.* [97] to track a 23 d.o.f. model using a 3D voxel reconstruction from 6 cameras.

An alternative to the EKF is the so-called *Unscented Kalman Filter* (UKF) [76], that uses a different (stochastic) method to linearize the nonlinear mapping $g$ and $h$ of a Gaussian. In short, the UKF transforms a set of so-called *sigma points* of a Gaussian state estimate (located at the mean and symmetrically along the main axes of the covariance defined by the eigenvectors) by passing them through $g$ and $h$. The resulting points are then used to calculate the linearized Gaussian (see *e.g.* [152] for a more detailed description). This method is more accurate (second-order approximation) than the linearization by first-order Taylor expansion as applied by the EKF. In many practical applications however, the difference between UKF and EKF is negligible, as the general restrictions for the EKF still apply. The UKF has nonetheless been successfully applied to the task of rigid 3D hand tracking [144], but to the best of our knowledge, no tracking of articulated models has been shown using the nonlinear extensions of Kalman filters presented in this section.

### 4.1.3 Particle Filters

All of the estimators for Bayesian filtering presented so far assume that the posterior pdf can be approximated by a Gaussian. *Particle Filters* are sequential Monte Carlo methods (*i.e.* stochastic methods based on random sampling) and provide the highest generality of all estimators for Bayesian filtering. They are nonparametric and thus able to approximate arbitrary pdfs, using a set of point-based estimates. Furthermore, neither motion nor observation models are restricted to linear mappings.

The posterior pdf at every timestep is approximated by a set $\mathcal{S}_t = \{\langle \mathbf{x}_t^{(i)}, w_t^{(i)} \rangle\}_{i=1}^{N}$ of $N$ weighted particles that consist of point-based state estimates $\mathbf{x}_t^{(i)}$ and associated normalized weights $w_t^{(i)}$ with $\sum_{i=1}^{N} w_t^{(i)} = 1$. Each particle can be seen as a hypothesis for the true state at time $t$.

Formally, the discrete approximation of the posterior can be expressed as follows using the *Dirac delta function* $\delta$:

$$p\big(\mathbf{x}_t \,\big|\, \mathbf{y}_{0:t}\big) \approx \sum_{i=1}^{N} w_t^{(i)}\, \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \tag{4.15}$$

In practice, the samples $\mathbf{x}_t^{(i)}$ cannot be drawn directly from the true posterior distribution, as it is unknown. Therefore, we have to sample our point estimates from a known and easy to sample *proposal distribution* $q\big(\mathbf{x}_t \,\big|\, \mathbf{y}_{0:t}\big)$. Ideally, the proposal distribution should resemble the *target distribution* $p\big(\mathbf{x}_t \,\big|\, \mathbf{y}_{0:t}\big)$ as close as possible. The only formal requirement however is that it must have non-zero probabilities in all regions where the target probability has non-zero probabilities. The importance weights $w_t^{(i)}$ in Equation 4.15 are needed to convert the unweighted sample set $\big\{\langle \mathbf{x}_t^{(i)},\, \frac{1}{N}\rangle\big\}_{i=1}^{N}$ that has been sampled from the proposal distribution into a weighted sample set $\mathcal{S}_t = \big\{\langle \mathbf{x}_t^{(i)},\, w_t^{(i)}\rangle\big\}_{i=1}^{N}$ that is distributed according to the target distribution. This is achieved when computing the weights as follows:

$$w_t^{(i)} = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(\mathbf{x}_t^{(i)} \mid \mathbf{y}_{0:t})}{q(\mathbf{x}_t^{(i)} \mid \mathbf{y}_{0:t})} \tag{4.16}$$

The aforementioned principle is known as *importance sampling* [47].

The particle set $\mathcal{S}_t$ including the importance weights $w_t^{(i)}$ can be computed recursively. We will assume that we start with a valid particle set $\mathcal{S}_{t\text{-}1}$ of unweighted particles that approximate the posterior pdf $p\big(\mathbf{x}_{t\text{-}1} \,\big|\, \mathbf{y}_{0:t\text{-}1}\big)$ from the last timestep. As proposal distribution, we will choose the *prior* pdf from Equation 4.4. We can easily sample from this distribution by applying the *motion model* $p\big(\mathbf{x}_t \,\big|\, \mathbf{x}_{t\text{-}1}\big)$ to all particles in $\mathcal{S}_{t\text{-}1}$ (see Section 4.2). The importance weights in Equation 4.16 are then calculated using substitutions from Equations 4.4 and 4.5:

$$w_t^{(i)} = \frac{\eta\, p(\mathbf{y}_t \mid \mathbf{x}_t^{(i)})\, p(\mathbf{x}_t^{(i)} \mid \mathbf{y}_{0:t\text{-}1})}{p(\mathbf{x}_t^{(i)} \mid \mathbf{y}_{0:t\text{-}1})} = \eta\, p\Big(\mathbf{y}_t \,\Big|\, \mathbf{x}_t^{(i)}\Big) \tag{4.17}$$

Here, $\eta$ is a normalizing constant that can be omitted when normalizing the importance weights in $\mathcal{S}_t$. As can be seen, the weights are easily computed by evaluating the *likelihood* $p\big(\mathbf{y}_t \,\big|\, \mathbf{x}_t\big)$ as defined by the *observation model* (see Section 4.3). Under the aforementioned assumptions, Equation 4.15 will converge towards the posterior pdf as the number of particles $N$ grows towards infinity.

This particle-based approach to recursive Bayesian estimation is known as *sampling importance resampling* (SIR) algorithm. In each timestep, the particle set $\mathcal{S}_{t\text{-}1}$ from the previous timestep is transformed into the particle set $\mathcal{S}_t$ in three sequential steps. Figure 4.2 and Algorithm 4.3 depict this strategy. The prediction step corresponds to drawing the particles from

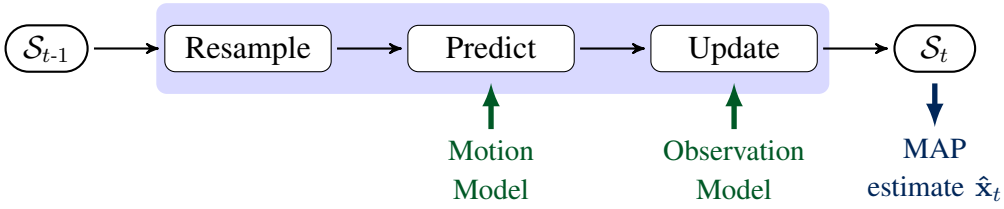


FIGURE 4.2 One timestep of Sampling Importance Resampling. The weighted particle set $\mathcal{S}_{t\text{-}1}$ from the previous timestep is resampled to create an unweighted set where particles have been drawn with probability proportional to their weights. The resampled particles are then used to predict new states according to the motion model. In the last step, particle weights are updated based on the current observation. This yields the new weighted particle set $\mathcal{S}_t$, from which a final state estimate $\hat{\mathbf{x}}_\mathbf{t}$ can be computed.

**Algorithm 4.3** Sampling importance resampling (SIR)

**Require:** $\mathcal{S}_{t\text{-}1} = \{\langle \mathbf{x}_{t\text{-}1}^{(i)}, w_{t\text{-}1}^{(i)} \rangle\}_{i=1}^{N}$ /* particle set from previous timestep */

1: $\mathcal{S}_t = \emptyset$
2: **for** $n = 1$ to $N$ **do**
3: draw $i$ with probability $\sim w_{t\text{-}1}^{(i)}$ /* resample */
4: sample $\mathbf{x}_t^{(n)} \sim p(\mathbf{x}_t \mid \mathbf{x}_{t\text{-}1}^{(i)})$ /* predict */
5: $w_t^{(n)} = p(\mathbf{y}_t \mid \mathbf{x}_t^{(n)})$ /* update */
6: $\mathcal{S}_t = \mathcal{S}_t \cup \{\langle \mathbf{x}_t^{(n)}, w_t^{(n)} \rangle\}$ /* add to particle set */
7: **end for**
8: **return** $\mathcal{S}_t = \{\langle \mathbf{x}_t^{(i)}, w_t^{(i)} \rangle\}_{i=1}^{N}$ /* particle set for current timestep */

the proposal distribution (*i.e.* the prior pdf), while the update step computes the importance weights that change the underlying distribution to the target distribution (*i.e.* the posterior pdf). Afterwards, the importance weights are normalized such that they sum up to 1. An additional resampling step is added before the prediction to transform the weighted particle set into an unweighted particle set without altering the underlying distribution. This is achieved by drawing particles from $\mathcal{S}_{t\text{-}1}$ with probability proportional to their weights. As a result, particles with high likelihood are expected to multiply, while particles with low likelihood might be removed from the set, which helps to prevent the particle set from becoming *degenerated* [47]. The resampling step is sometimes performed after the update step, which is otherwise equivalent. The only difference is that the particle set $\mathcal{S}_t$ will constitute an unweighted representation of the posterior pdf in this case.

In tracking, one is usually interested in a *maximum a posteriori* (MAP) estimate $\hat{\mathbf{x}}_\mathbf{t}$ of the

posterior, *i.e.* in its global mode. We can approximate this estimate by selecting the particle $\mathbf{x}_t^{(j)}$ from $\mathcal{S}_t$ with the highest associated weight $w_t^{(j)}$. Alternatively, we can use the particle set to approximate expectations of interest (*e.g.* the mean) from the set of available particles:

$$E[f(\mathbf{x}_t)] \approx \sum_{i=1}^{N} f(\mathbf{x}_t^{(i)})\, w_t^{(i)} \tag{4.18}$$

Note that calculating the mean of a distribution and using it as the final estimate for the tracked state is only recommended for unimodal distributions.

Although particle filters provide a general and elegant methodology for recursive state estimation, we need to bear in mind some of the drawbacks of SIR as presented in this section. First of all, the number of particles needed to approximate a pdf grows exponentially with the dimensionality $d$ of the state space $\mathcal{X}$. Therefore, SIR quickly becomes intractable when used in conjunction with high-dimensional models. We will present some options how to modify SIR to help in overcoming this limitation later in this chapter (Sections 4.4 and 4.5). A common point of criticism is the choice of the proposal distribution as in Equation 4.17. By using the prior pdf as the proposal, the current observation $\mathbf{y}_t$ is not involved when drawing particles from the proposal. Other variants such as the *auxiliary particle filter* (AuxPF) [111] or the *unscented particle filter* (UPF) [159] have been proposed that incorporate the latest observation to improve the efficiency of the particle set by a more informed selection of particles. Although this facilitates tracking with fewer particles, the computational costs for additional particle evaluations reduce the potential efficiency gain. Therefore, parametric approximations are often used to provide approximations for these evaluations. It yet remains to be evaluated whether suitable approximations can be found in the case of human pose tracking.

## 4.2 Motion Model

As discussed in the last section, probabilistic tracking can be formulated recursively in two alternating steps: prediction and update. While the general principles and algorithmic details of common Bayesian estimators are well-defined, the challenges in creating a successful tracking application lie in (•) the choice of a suitable Bayesian estimator (•) the choice of a representative motion model $p(\mathbf{x}_t \,|\, \mathbf{x}_{t\text{-}1})$ to estimate the prior pdf according to Equation 4.4 (•) the choice of a powerful observation model $p(\mathbf{y}_t \,|\, \mathbf{x}_t)$ that allows for a correct and accurate representation of the posterior pdf according to Equation 4.5. We have already shown that particle filters provide a suitable framework for the estimation, considering the multi-modal distributions and non-linear motion and observation models that are typical in human pose

estimation. We will now describe our choice for the motion model.

The motion model is relevant for describing the temporal evolution of a state (*i.e*. a human pose) during tracking. Ideally, precise knowledge of the human dynamics would enable us to directly predict a future state from the current state. Sometimes, the state space can be augmented with parameters describing the current dynamics of the system, *e.g*. velocity or acceleration. These parameters can then be used to predict the new pose based on first-order (constant velocity) or second-order (constant acceleration) dynamics. In practice, both constant velocity and constant acceleration models tend to produce unreliable predictions. While constant velocity simply is not a good model to describe human motions, constant acceleration is a reasonably well approximation of the angular motions in human body joints as long as no physical interaction with the environment takes place (*e.g*. ground plane contact). A severe problem with constant acceleration models is the difficulty to get an accurate estimate of the acceleration, as the framerates of the recording cameras are usually too low and produce an undersampling of the continuous velocity profile.

More complex motion models are often learned from exemplar motions using techniques such as *Hidden Markov Models* (HMM) [30, 119]. Some approaches try to estimate a low-dimensional manifold inside the high-dimensional pose space, *e.g*. by using *Principal Component Analysis* (PCA) [135, 158] or *Gaussian Process Dynamical Models* (GPDM) [163, 157]. While this enables efficient and accurate predictions of known motions, exemplar-based methods are unable to predict arbitrary, yet unobserved motions. Another problem is that exemplar motions are difficult to come by, so either marker-based motion capture systems need to be employed, or the motions need to be modeled manually by experts.

Considering the limitations of the aforementioned motion models, we decided to use the simplest and most general motion model for tracking. Thus, we use a static pose assumption and add a Gaussian white noise term $\boldsymbol{\delta}$ to account for the uncertainty introduced by the unknown motion. Our motion model can be written formally as follows:

$$\mathbf{x}_t = \mathbf{x}_{t\text{-}1} + \boldsymbol{\delta} \quad ; \qquad \boldsymbol{\delta} \sim \mathcal{N}(0,\, \boldsymbol{\Sigma}) \tag{4.19}$$

Here, $\boldsymbol{\Sigma}$ is a diagonal matrix where the diagonal entries correspond to the variance $\sigma_j^2$ of the $j$-th component of $\mathbf{x}$. The variance influences the amount of diffusion for each pose parameter or joint angle in the state vector $\mathbf{x}$. It is specified only once before tracking, and is dependant on the frames per second (f.p.s.) of the recordings. We have estimated biomechanical maximum angular velocities $\mathbf{vmax}_{\mathtt{bp}}$ for each body part $\mathtt{bp}$ based on isolated motion capture experiments and data provided by sport scientists (see Section 3.3.2). These can be transformed into body-part dependant inter-frame standard deviations $\boldsymbol{\sigma}_{\mathtt{bp}}$ by considering the f.p.s.

used for tracking. For sequences captured at $25$ f.p.s., standard deviations range from $0.2°$ for some of the joints in the spine up to $37.8°$ for the torsion of the forearms (the values for $\boldsymbol{\sigma}_{\mathrm{bp}}$ were estimated from Table 3.4 by dividing each component with the f.p.s.). During our experiments, we have limited all components of $\boldsymbol{\sigma}_{\mathrm{bp}}$ to a maximum of $12.5°$, or else the tracking would become both inaccurate and inefficient. Note that the absolute angular difference between two frames can actually be much higher than suggested by these values, as we use iterative annealing strategies in our tracker that effectively extend the reach (we will introduce these strategies later in this chapter). Equation 4.19 implements line 4 in Algorithm 4.3.

This simple motion model is capable of tracking arbitrary human motions. However, the uninformed prediction comes at a computational expense, as we have to use more particles for successful tracking. In Chapter 6 we will present informed motion models that are learned from exemplar data that has been retrieved using this simple and unconstrained motion model. These can be updated incrementally over time, enabling the human motion tracker to adapt itself towards environment-specific activities.

## 4.3 Observation Model

After presenting the motion model in the last section, we will now describe the observation model we use for human pose estimation. During development of the observation model, our primary objective was to provide an unintrusive setup for automated observation of human motions with passive sensors only. Furthermore, secondary objectives have been (•) the use of affordable and common sensor hardware (•) the development of generally applicable models that run in different scenarios without the need for parameter tuning (•) the development of highly accurate yet computationally feasible models.

The difficulty in human motion tracking, besides the high dimensionality of the problem, is the fact that one tries to estimate a 3D articulated human pose from 2D sensors. We have decided to use a setup with multiple CCD cameras from different viewpoints to tackle the problem (Figure 2.1). Single view setups suffer from view ambiguities and self-occlusion in human postures. Even when using sensors that output 3D data, such as *stereo cameras* [112, 64, 11] or *time-of-flight cameras* [82], recognition is still restricted to frontal or at best side views of human postures, unless learned models of previously observed human motions are employed [137].

Our observation model is based on an analysis-by-synthesis approach, where rendered silhouette shape projections of predicted particle states are compared to foreground masks extracted once per timestep by means of a foreground-background segmentation of the current

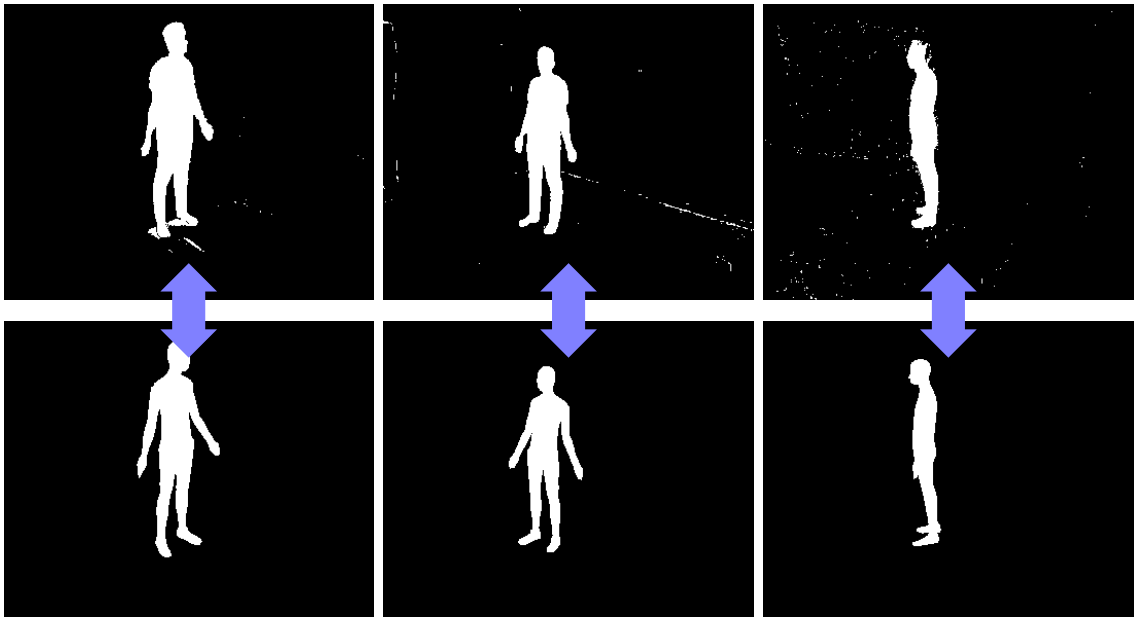

FIGURE 4.3 Binary region masks used in observation model. The foreground region masks $I_F$ (top row) are extracted using foreground-background segmentation, and compared to the rendered projection masks $I_P^{(n)}$ (bottom row) of the human model and a predicted state estimate $\mathbf{x}_t^{(n)}$. Masks are shown for three different camera views and a matching prediction.

images (Figure 4.3). A good overlap indicates a good pose estimate and should result in a high particle weight. This strategy is quite common in human pose estimation [2, 46, 121], as silhouette shapes provide rich and almost unambiguous information about a human pose, provided that enough cameras are used (see a discussion in Section 4.6.1). This holds despite the lack of depth or luminance information. Another advantage is that silhouette shapes are easy to extract by means of standard foreground-background segmentation techniques [143, 49, 80].

We will assume a calibrated setup of multiple cameras (a minimum of three is required but also sufficient in most cases) that are placed such that their viewing directions towards the common area of interest differ by at least 30°. Additionally, no two cameras should be placed opposite of each other, as this will result in mirrored silhouettes that contain little additional information. Internal and external calibration of the cameras is performed using common techniques [154, 173]. Furthermore, we will assume that a technique for the extraction of silhouette shapes is readily available. In our approach we use the codebook foreground-background segmentation method by Kim *et al.* [80], that provides sufficient results in most cases. However, the method could be easily replaced by more sophisticated techniques if necessary [124, 172]. Note that the computational expenses for the foreground-background segmentation are negligible when compared to the particle evaluations, as they have to be

computed only once per timestep.

Both the foreground mask $I_F$ extracted from the current image by means of foreground-background segmentation, and the projection masks $I_P^{(n)};\ n = 1 \dots N$ for the $N$ states predicted by the current particle set are binary region masks, where $I[x, y] \in \{0, 1\}$ corresponds to the binary value of the region mask $I$ at pixel $[x, y]$. Thus, we can define the following pixelwise logical operators on binary region masks that will be used throughout this thesis:

$$\begin{aligned}
\mathrm{AND}(I_a, I_b)[x, y] &= I_a[x, y] \ \wedge \ I_b[x, y] && (4.20)\\
\mathrm{XOR}(I_a, I_b)[x, y] &= I_a[x, y] \ \vee \ I_b[x, y] && (4.21)\\
\mathrm{DIFF}(I_a, I_b)[x, y] &= I_a[x, y] \ \ominus \ I_b[x, y] && (4.22)\\
\mathrm{XOR}(I_a, I_b)[x, y] &= I_a[x, y] \ \triangle \ I_b[x, y] && (4.23)\\
\mathrm{NOT}(I)[x, y] &= \neg I[x, y] && (4.24)\\
\mathrm{COUNT}(I) &= \sum_{x,y} I[x, y]\,; \qquad I[x, y] \in \{0, 1\} && (4.25)
\end{aligned}$$

The four binary operators AND, OR, DIFF and XOR correspond to pixelwise intersection ($\wedge$), union ($\vee$), difference ($\ominus$) and symmetric difference ($\triangle$) operations on the two image masks $I_a$ and $I_b$. NOT specifies inversion ($\neg$). To simplify matters, we assume that these operations work on the image masks of all cameras in parallel (*i.e.* one can assume that region masks from different cameras are stitched together to one big mask). The COUNT operation is used to sum up the non-zero pixels of all cameras.

The absolute shape error $e_s^{(n)}$ corresponding to the mismatch between a predicted mask $I_P^{(n)}$ and the observed foreground mask $I_F$ is then calculated as follows:

$$e_s^{(n)} = \mathrm{COUNT}(\,\mathrm{XOR}(I_P^{(n)},\, I_F)\,) \tag{4.26}$$

In contrast to many other approaches (*e.g.* [46, 14]), we do not use a *sum-of-squared-distances* (SSD) error measure between selected points on the model and the foreground mask (this would correspond to $\mathrm{COUNT}(\,\mathrm{DIFF}(I_P^{(n)}, I_F)\,)$ in our notation). While these approaches are able to explain a projection by the observed foreground mask, the XOR error measure also takes into account that the foreground mask should be covered as good as possible by the projection. When ignoring this, arms tend to stick to the torso unless other visual cues are incorporated, *e.g.* edges (that however tend to be unreliable with clothing). Other approaches that account for this problem are *e.g.* based on *Chamfer distances* [139]. While being elegant solutions, the calculation of distance transforms for each projection mask quickly becomes prohibitive as the number of particle evaluations increases.

| | 380×288 IMG | 380×288 RLE | 640×480 IMG | 640×480 RLE | 1920×1080 IMG | 1920×1080 RLE |
|---|---|---|---|---|---|---|
| Render Mask | 6.20 $s$ | 3.02 $s$ | 9.13 $s$ | 4.00 $s$ | 14.75 $s$ | 7.14 $s$ |
| Calculate Error | 1.07 $s$ | 0.11 $s$ | 3.08 $s$ | 0.14 $s$ | 29.75 $s$ | 0.32 $s$ |
| Overall | 7.27 $s$ | 3.13 $s$ | 12.21 $s$ | 4.14 $s$ | 44.50 $s$ | 7.46 $s$ |

**TABLE 4.1** Processing times for $10^4$ particle evaluations of the weight function when using image based region masks (IMG) or run-length-encoded region masks (RLE). Processing times are listed for rendering the projection mask $I_P$, for calculating the absolute shape error $e_s$ according to Equation 4.26, and overall. As can be seen, error calculations can become prohibitive for large image sizes unless RLE is used. When using RLE, the processing time is dominated by the rendering of the silhouette masks, and increases only sublinearly with respect to the image size.

It should be noted that using the symmetric difference (XOR) requires the whole image mask to be processed, which can also be computationally expensive if implemented trivially. We alleviate the increased computational expense by representing region masks using *run-length-encoding*. Here, the binary masks are represented as a list of intervals (*chords*) defined by their *row* and their start and end *column* image coordinates. Such a representation is very efficient both for morphological preprocessing steps as well as for the bitwise logical operations presented in Equations 4.20 to 4.25. The complexity of these operations is effectively reduced from $\mathcal{O}(xy)$ to $\mathcal{O}(cy)$, with $x$ and $y$ being the width and height of the image and $c$ being a constant for which $c \ll x$ holds. Table 4.1 present a comparison of the processing time (on a consumer laptop) for a repeated computation of $10^4$ update steps for a single particle (including the rendering of the projection masks and the error measure according to Equation 4.26). As these operations are by far the most expensive ones in our tracking algorithm, the impact of the run-length-coded region masks becomes evident, especially when processing high resolution images from *Full HD* videos.

A remaining problem is the calculation of particle weights $w^{(n)}$ from the absolute error measure $e_s^{(n)}$. In particular, we are missing a point of reference that would enable us to make a qualitative statement about $e_s^{(n)}$. Because the maximum region area for the model projection varies with the state estimate $\mathbf{x}^{(n)}$, it is not suited to serve this purpose. Furthermore, when normalizing the errors with respect to the absolute projection area, it becomes difficult to distinguish between subtle differences in absolute errors, such as when a lower arm is only slightly off. Instead, we propose to use the following mapping from absolute errors to particle weights:

$$w^{(n)} = 1 - \frac{e_s^{(n)} - \min\limits_n e_s^{(n)}}{\max\limits_n e_s^{(n)} - \min\limits_n e_s^{(n)}} \tag{4.27}$$

Here, the errors from Equation 4.26 are scaled relatively to all other errors that have been calculated for the predicted particle states $\mathbf{x}^{(n)}$. Thus, the particle with the lowest error gets a weight of 1, while the particle with the highest error gets a weight of 0. Scaling the error in such a way not only manages to separate good from bad particles without having to interpret absolute error values, it is also a good possibility to influence the survival diagnostic $\mathcal{D}$:

$$\mathcal{D} = \frac{1}{\sum_{n=1}^{N} {w^{(n)}}^2} \tag{4.28}$$

The survival diagnostic was introduced by MacCormick and Isard [94], and gives an estimate of the number of particles that will survive a resampling step (*i.e.* the effective sample size [9]). We found that it can also serve as a tool for controlling the particle spread in a non-parametric way. In our experiments, the best tracking results were achieved when only about a third of the particles survive each resampling step, which protects the particle set from the degeneracy problem [9]. Because particle weights computed by Equation 4.27 are (usually equally) distributed across the interval $[0:1]$, we can use one of the following equations to fine-tune the survival diagnostic:

$$w^{(n)} = {w^{(n)}}^{\beta} \tag{4.29}$$

$$w^{(n)} = 1 - \left(1 - {w^{(n)}}^{a}\right)^{b} \tag{4.30}$$

Figure 4.4 plots these functions for various parameterizations. While Equation 4.29 is more intuitive ($\beta < 1 \Rightarrow$ larger $\mathcal{D}$, $\beta > 1 \Rightarrow$ smaller $\mathcal{D}$), Equation 4.30 provides fine-grained control over how many of the top-weighted particles should be kept. It is usually sufficient to set these parameters only once when designing the tracking algorithm, which effectively makes the observation model presented in this section a non-parametric one. In all our experiments we have set $\beta = 8$ or $a = 16\ ;\ b = 8$ respectively.

Some related model-based approaches do not employ silhouette shapes directly, but rather use them to carve a convex 3D shape out of Euclidean world space $\mathbb{R}^3$. These shapes are often referred to as *visual hulls*, and usually require at least 6-8 viewpoints for an accurate approximation of human shapes. Kehl and Van Gool [79] use nearest-neighbor distances and Horaud *et al.* [68] also surface normals to register a human model to the surface of the visual

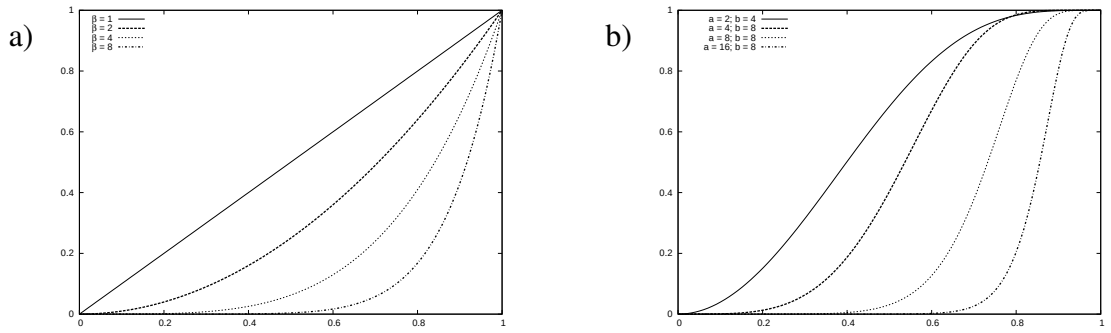


FIGURE 4.4 Weight scale functions for controlling the survival diagnostic $\mathcal{D}$. a) Using a single parameter $\beta$ according to Equation 4.29. The weight function is smoothed for $\beta < 1$ (*i.e.* more particles survive), or sharpened for $\beta > 1$ (*i.e.* fewer particles survive). b) Using two parameters $a$ and $b$ according to Equation 4.30, enabling more fine-grained control.

hull, however it is unclear whether this provides an improvement over evaluating multiple 2D silhouettes directly, considering that the underlying information is the same. A more common use of visual hulls is in approaches that do not rely on a prior shape model of humans but instead estimate the articulated structure directly from the temporal sequence of visual hulls [39, 6]. When compared to visual hulls, observation models based on 2D silhouette shapes can be beneficial when it comes to handling occlusions and dynamic objects, as will be shown in Section 5.2.

The observation model presented so far solely relies on shape cues (although color information is used for the foreground-background segmentation). In Section 5.1.2 we will show how the weighting function can be augmented with learned color-based appearance models of the humans.

## 4.4 Hierarchical Sampling Strategies

Particle filters try to solve the Bayesian filtering problem by approximating the posterior pdf through discrete samples. They are very effective in doing so as long as the state space is reasonably small. However, with increasing dimensionality, the number of particles necessary for a sufficiently good approximation grows exponentially, as the whole state space needs to be sampled. The required quantities of particles thus quickly become intractable. To get a better understanding of this problem, one can calculate the effective particle count per dimension $c_d$ given the number of particles $N$ and the dimensionality of the state space $d$ as follows:

$$c_d = \sqrt[d]{N} \tag{4.31}$$

The effective particle count $c_d$ corresponds to the number of particles available for each dimension when distributing all particles according to an uniformly spaced grid across the state space. Thus, to have an effective particle count of $10$ per dimension, one needs $1000$ particles when the dimensionality of the state space is $3$, but one would need $10^{30}$ particles for an estimation problem with 30 d.o.f. Note that $c_d$ should be considered as a demonstrative quantity for the general problem of exponential growth in particles with dimensionality. In practice, particle filters are often successful with fewer effective particles per dimension, as the probability mass is not equally distributed across the state space.

The variants of particle filtering presented so far (Section 4.1.3) are therefore not suited for tracking high-dimensional articulated models such as the ones necessary for human pose estimation. Although successful tracking has been demonstrated for problems with $< 10$ dimensions, *e.g.* robot localization [152] or contour based tracking of hand models [71], no mentionable results have been shown for model-based human pose estimation.

To deal with the curse of dimensionality, two different strategies seem promising and have been proposed in the literature. First, instead of approximating the full posterior pdf, one can try to concentrate particles around the modes of the distribution, as these are relevant areas of the pdf when estimating the most likely pose for a timestep. Second, the state space can be divided into smaller partitions that are evaluated independently. We will discuss these two directions in the following sections.

### 4.4.1 Annealed Particle Filter

Traditional SIR particle filtering aims at approximating the posterior pdf, whereas in high dimensional tracking ($\geq 10$ d.o.f.), the number of particles is clearly insufficient for such an approximation. However, in tracking applications one is usually interested in computing the *maximum a posteriori* (MAP) estimate $\hat{\mathbf{x}}_t$ for each timestep $t$, *i.e.* the global mode of the posterior pdf:

$$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} p\big(\mathbf{x}_t \,\big|\, \mathbf{y}_{0:t}\big) \tag{4.32}$$

Thus, a strategy to deal with the problem of high dimensionality in particle filtering is to focus particles around the modes of the pdf. This helps to increase particle efficiency and thus to reduce the number of particles necessary for successful tracking, facilitating its application to higher dimensional tracking problems.

Deutscher and Reid [46] have proposed the *annealed particle filter* (APF) in the context of human pose estimation as a variant of particle filtering that complies with the aforementioned strategy (they show results on a short sequence with 34 d.o.f.). It can be seen as a combina-

tion of particle filtering with the concept of *simulated annealing* [81]. Their method shares some ideas with *annealed importance sampling* (AIS), which was introduced by Neal [105] as an importance sampler for high-dimensional problems with isolated modes in the context of *Markov Chain Monte Carlo* (MCMC) methods. Gall *et al*. [56] provide an in-depth analysis of both methods, which they term *interacting simulated annealing* (generalization of APF) and *interacting annealed sampling* (variant of AIS). They show that APF is superior in high-dimensional motion tracking tasks, as it converges towards the global modes (they show an application with 18 d.o.f.). AIS on the other hand provides a better approximation of the posterior density, but is not as powerful in retrieving the final MAP estimate $\hat{\mathbf{x}}_t$ for high-dimensional tracking tasks. We will thus focus our description on the APF algorithm for the remainder of this chapter.

Simulated annealing has been introduced in the context of *Markov Chain Monte Carlo* (MCMC) methods as an efficient means to estimate the global mode of a pdf. In MCMC methods [4], a pdf is approximated by simulating a Markov Chain (*e.g*. by using the *Metropolis-Hastings* algorithm [96]) whose invariant distribution will converge towards the pdf (*i.e*. the samples drawn will represent the pdf). In simulated annealing, samples are no longer drawn according to $p(\mathbf{x})$, but from a distribution $p_i(\mathbf{x}) \propto p(\mathbf{x})^{1/T_i}$, where $i$ is the index of the current iteration, and $T_i$ corresponds to the *temperature* as defined by a decreasing cooling schedule (thus the term *annealing*). As the *temperature* decreases (with $\lim_{i\to\infty} T_i = 0$), the invariant distribution will become equal to $p(\mathbf{x})^{\infty}$, which under weak regularity assumptions on $p(\mathbf{x})$ is a pdf whose probability mass is concentrated on the global modes of $p(\mathbf{x})$. Thus, simulated annealing is a method to perform stochastic optimization on a pdf $p(\mathbf{x})$.

The concept of simulated annealing can be translated from MCMC methods to particle filtering. Instead of using a single particle to sample from the posterior distribution, the principle of annealing is applied to a whole set of particles in the APF. This turns out to be crucial when trying to estimate the global modes of a high-dimensional pdf where the probability mass is sparsely distributed across the state space, as is the case in human pose estimation. The details of the APF are depicted in Algorithm 4.4, which is additionally visualized in Figure 4.5. Note that in accordance with [46], the layer iterations will be labelled in descending order from $m = M$ to $m = 1$, and $\mathcal{S}_{t,m}$ will correspond to the particle set at timestep $t$ and iteration $m$.

In each timestep, a multi-layered search (starting from layer $m = M$ to layer 1) is conducted to gradually move the particle set towards the global maximum. Each iteration corresponds to a slightly modified version of SIR particle filtering with two key modifications.

First, the amount of diffusion added during the *prediction* step is gradually reduced, so that the initially broad search region becomes fine-grained towards the last iterations. This strat-

**Algorithm 4.4** Annealed particle filter (APF)

**Require:** $\mathcal{S}_{t\text{-}1} = \{\langle \mathbf{x}_{t\text{-}1}^{(i)}, w_{t\text{-}1}^{(i)} \rangle\}_{i=1}^{N}$ /* particle set from previous timestep */
$\langle \alpha_M, \ldots, \alpha_1 \rangle$ /* diffusion variance scheme */
$\langle \beta_M, \ldots, \beta_1 \rangle$ /* annealing scheme */

$\mathcal{S}_{t,M+1} = \mathcal{S}_{t\text{-}1}$
**for** $m = M$ to $1$ **do** /* iterate through layers */
  $\mathcal{S}_{t,m} = \emptyset$
  **for** $n = 1$ to $N$ **do**
    draw $i$ with probability $\sim w_{t,m+1}^{(i)}$ /* resample */
    $\mathbf{x}_{t,m}^{(n)} = \mathbf{x}_{t,m+1}^{(i)} + \boldsymbol{\delta}$ ; $\boldsymbol{\delta} \sim \mathcal{N}(0, \alpha_m \boldsymbol{\Sigma})$ /* predict with scaled diffusion */
    $w_{t,m}^{(n)} = p(\mathbf{y}_t \mid \mathbf{x}_{t,m}^{(n)})^{\beta_m}$ /* update using annealed weight functions */
    $\mathcal{S}_{t,m} = \mathcal{S}_{t,m} \cup \{\langle \mathbf{x}_{t,m}^{(n)}, w_{t,m}^{(n)} \rangle\}$ /* add to current layer particle set */
  **end for**
**end for**
$\mathcal{S}_t = \mathcal{S}_{t,1}$
**return** $\mathcal{S}_t = \{\langle \mathbf{x}_t^{(i)}, w_t^{(i)} \rangle\}_{i=1}^{N}$ /* particle set for current timestep */

egy is a natural way to account for the decreased uncertainty as the search progresses, and increases the estimation accuracy when approaching the global maximum (see Figure 4.6 for an illustration). The exact amount of diffusion in each iteration is controlled by the diffusion variance scheme $\langle \alpha_M, \alpha_{M-1}, \ldots, \alpha_1 \rangle$, where $\alpha_m \leq 1$ defines the scaling factor for the diffusion variance in iteration $m$ (a common choice for this schedule is to halve the variance in every iteration, *i.e.* $\alpha_M = \alpha_{M-1} = \ldots = \alpha_1 = 0.5$).

The second modification for the SIR steps is the annealing of the weight function in the update steps by exponentiating the particle weights as in Equation 4.29 with $\beta_m$ dependant on the current iteration $m$. By increasing $\beta_m$ with each iteration according to a predefined annealing scheme $\beta_M < \beta_{M-1} < \ldots < \beta_1$, particle survival will be increasingly coupled to state space regions with high observation weights. In other words, particles are able to move around the state space more freely during early iterations, thus having the potential to escape local maxima. In the last iterations however, particles will concentrate in regions with high probability, and eventually they should converge towards the global modes. When viewed from a different perspective, annealing the weight function over several iterations has

the effect of gradually reducing the survival diagnostic $\mathcal{D}$ as introduced in Equation 4.28, until only a few (*i.e.* the *fittest*) particles survive each resampling step.

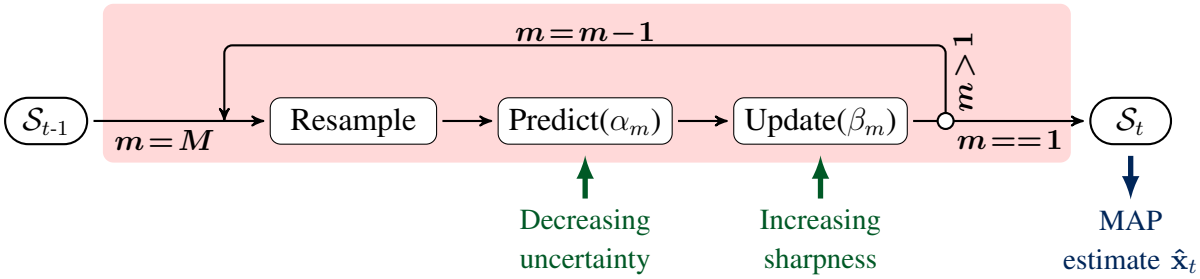


FIGURE 4.5 One timestep of Annealed Particle Filtering. The final particle set is estimated in $M$ iterations . Each iteration is comparable to a SIR step, where the amount of diffusion (*i.e.* uncertainty) added during the prediction step is reduced with each iteration by a factor $\alpha_m$, and where the weighting function in the update step is annealed according to the scheme $\langle \beta_M, \beta_{M-1}, \ldots, \beta_1 \rangle$ . This annealing has the effect of gradually sharpening the weight function to slowly push particles towards the global maximum.

Figure 4.6 visualizes the iterations of the APF during one timestep when used for human pose estimation, *i.e.* when the states of the particles correspond to a specific posture of a human model. It illustrates the effect of gradual convergence towards the global maximum that is due to the scaling of the diffusion variances and the annealing of the weight function.

It should be mentioned that focusing particles only around the global modes of the posterior violates the formulation of the SIR particle filter as a general solution to the recursive Bayesian estimation as presented in section 4.1.3. The ability to represent multi-modal distributions and thus to overcome tracking ambiguities in the light of new data is partially lost. In practice however, recovery from temporary tracking failures is often possible due to the multi-layer search mechanism of the APF that results in a larger area of convergence.

#### 4.4.1.1 Covariance Scaled Diffusion

APF works best when the number of layers is large, *i.e.* when the particle set is given time to slowly converge towards the global maximum. This implies that there is a trade-off between accuracy and efficiency, and often accurate tracking results in high-dimensional spaces come at the expense of the latter. The intuitive reason behind this is that APF searches the whole state space at once, repeatedly over several layers. However, knowledge about the state space from previous iterations is not taken into consideration. Effectively this means that parts of the state space with low probability mass are repeatedly revisited, and that particle aggregations in regions with high probability mass are reshuffled in every iteration. Another shuffling

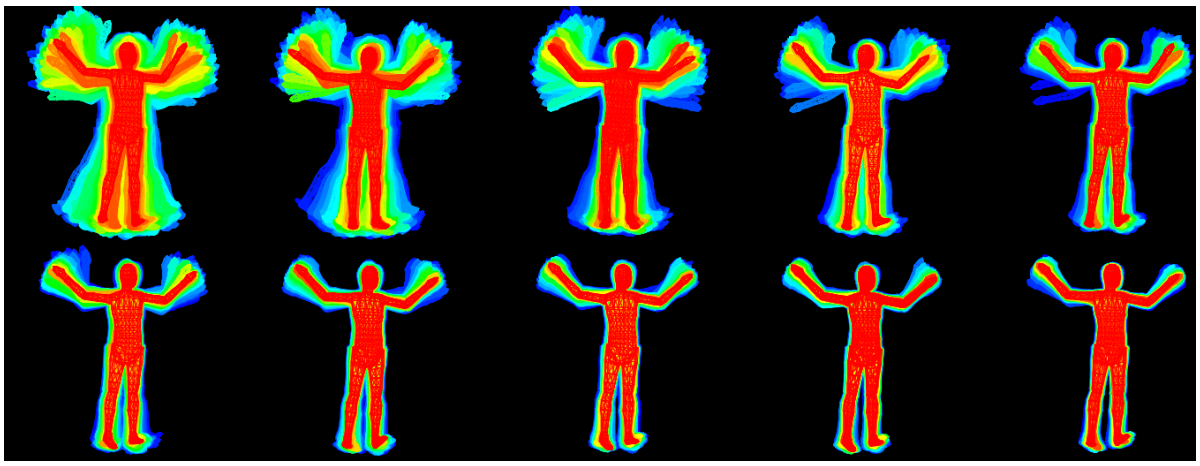

FIGURE 4.6 Visualization of the particle set during the iterations of the annealed particle filter (for 10 layers). Particles are rendered with their corresponding pose estimate. Higher color temperature corresponds to higher weights. With each iteration, diffusion is reduced and particles eventually cluster around the global maximum.

effect becomes evident when using articulated models, due to their kinematic dependencies. Consider *e.g.* joint angle parameters that are far away from the origin of the kinematic chain (such as wrists, elbows). These parameters are continuously estimated in every iteration even when the parameters of their kinematic predecessors are still highly uncertain. This can result in bad joint angle estimates being favored over good ones to correct observation errors introduced by incorrect estimates of the preceding parameters (according to Equation 4.26). After several iterations this effectively shuffles the inner and especially outer limb parameters, and prior knowledge from the last timestep is lost.

The aforementioned difficulties can be attributed to the lack of knowledge about kinematic dependencies of the state parameters in APF estimation. Instead, all parameters are assumed to be independent, which is not the case in human pose estimation. Deutscher *et al.* [46] have proposed *adaptive diffusion* as a means of soft hierarchical partitioning to alleviate the associated problems. Here, diffusion is adaptively guided based on how well a parameter has already been estimated, and the search becomes focused in regions where the optimal parameters could not yet be determined. A similar idea was presented by Sminchisescu and Triggs [141] as *covariance scaled sampling* in the context of monocular tracking to favor particle diffusion along the directions of the unobservable d.o.f. To be neutral, we will refer to this method as *covariance scaled diffusion*.

In covariance scaled diffusion, the random diffusion vector $\boldsymbol{\delta}$ that is added to each state parameter during prediction (line 6 of Algorithm 4.4) is sampled based on a Gaussian approximation of the state space density that is derived from the particle set $\mathcal{S}_{t,m+1}$ of the previous

iteration $m+1$ (except when $m = M$). This improves the effectivity of the search strategy by incorporating prior knowledge about the state space, which results in diffusion that is adaptive to the directions of high uncertainty. Mathematically, the diagonal matrix $\mathbf{\Sigma}$ from line 6 of Algorithm 4.4 is replaced by the sample covariance matrix $\mathbf{\Sigma}_{t,m+1}$ of the particle set $\mathcal{S}_{t,m+1}$ from the previous iteration (except when $m = M$, in which case the diagonal matrix $\mathbf{\Sigma}$ will still be used):

$$\mathbf{\Sigma}_{t,m+1} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_{t,m+1}^{(i)} - \bar{\mathbf{x}}_{t,m+1}) \cdot (\mathbf{x}_{t,m+1}^{(i)} - \bar{\mathbf{x}}_{t,m+1})^{\mathrm{T}} \tag{4.33}$$

where $\bar{\mathbf{x}}_{t,m+1}$ is the sample mean from that iteration:

$$\bar{\mathbf{x}}_{t,m+1} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{t,m+1}^{(i)} \tag{4.34}$$

The random diffusion term $\boldsymbol{\delta}$ in Equation 4.19 is now sampled from the multi-variate normal distribution $\mathcal{N}(0,\ \mathbf{\Sigma}_{t,m+1})$ calculated from the particle set $\mathcal{S}_{t,m+1}$.

Sampling from the multi-variate normal distribution $\mathcal{N}(0,\ \mathbf{\Sigma}_{t,m+1})$ is done by means of *eigendecomposition* of the covariance matrix $\mathbf{\Sigma}_{t,m+1}$. For numerical stability (especially since $\mathbf{\Sigma}_{t,m+1}$ might be singular in practice), it is best to use *singular value decomposition* (SVD) to find the *eigenvectors* and corresponding *eigenvalues*:

$$\mathbf{\Sigma}_{t,m+1} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}} \tag{4.35}$$

As $\mathbf{\Sigma}_{t,m+1}$ is a positive semi-definite symmetric $d \times d$ matrix, the column vectors of $\mathbf{U} = (\mathbf{e}_1 \ldots \mathbf{e}_d)$ correspond to its eigenvectors. Furthermore, $\mathbf{\Lambda}$ is a diagonal matrix with the corresponding eigenvalues $\lambda_i$ as diagonal entries.

Now let $\mathbf{\Lambda}^{1/2}$ be a diagonal matrix with the squareroots of the eigenvalues, and $\mathbf{z} = (z_1, \ldots, z_d)^{\mathrm{T}}$ a vector whose components $z_i \sim \mathcal{N}(0,\ 1)$ are $d$ independent standard normal variates (such variates can be computed using the Box-Muller method [117]). To account for the variance along the direction of the corresponding eigenvector $\mathbf{e}_i$, we need to scale the random components $z_i$ by the squareroot of the eigenvalues $\lambda_i$:

$$\mathbf{z}' = \mathbf{\Lambda}^{1/2}\, \mathbf{z} \tag{4.36}$$

The so-generated random noise vector $\mathbf{z}'$ is transformed from the space spanned by the eigen-

vectors into the state space by multiplying it with the eigenvector matrix $\mathbf{U}$:

$$\boldsymbol{\delta} = \mathbf{U} \cdot \mathbf{z}' \tag{4.37}$$

The resulting vector $\boldsymbol{\delta}$ is the requested sample from the multi-variate normal distribution $\mathcal{N}(0,\, \boldsymbol{\Sigma}_{t,m+1})$ and is used to diffuse the prediction as in line 6 of Algorithm 4.4, now in accordance to the covariance of the particle set in the last iteration of the APF.

Note that although particles are used more effectively when employing covariance scaled diffusion, the aforementioned problem of shuffling joint angle parameters that have many kinematic predecessors remains.

### 4.4.2 Partitioned Sampling

Although covariance scaled diffusion as discussed in the previous section provides a soft partitioning, a disadvantage of APF is that all state parameters are still effectively estimated in parallel. This is problematic when used in conjunction with articulated human body models that consist of hierarchically dependant parameters (*e.g.* the elbow joint parameters depend on the shoulder joint parameters). In such a case, a parallel estimation is inefficient and often leads to bad estimates, as we will show in Section 4.6.

An alternative approach to high dimensional state estimation is to subdivide the state space $\mathcal{X}$ into several partitions $\mathcal{X} = \mathcal{P}_1 \times \mathcal{P}_2 \times \ldots \times \mathcal{P}_K$ that can be estimated sequentially, following the hierarchical structure of the human body (*e.g.* torso, left upper arm, left lower arm, *etc.*). This strategy has been introduced by MacCormick and Blake [93] as *partitioned sampling* (PS) [93] in the context of multiple target tracking, and was later applied in the context of articulated pose tracking by MacCormick and Isard [94]. It can be seen as the statistical analogue to a hierarchical search.

Figure 4.7 illustrates PS for one timestep. The partitions are estimated sequentially using the same resampling, prediction and update pattern that is known from the standard SIR particle filter (Figure 4.2). However, the motion and observation models are exchanged with local versions, *i.e.* only the parts of the state space corresponding to the current partition $\mathcal{P}_k$ are estimated. Note that the original PS strategy as proposed by MacCormick and Blake [93] differs from Figure 4.7 in that they introduced a special *weighted resampling* operator for mathematical elegance. However, when exchanging this operator by an update step that is followed by *importance resampling* (as shown in Figure 4.7), PS can in fact be seen as a sequentially coupled series of local SIR particle filters. Both variants are equivalent, but we believe that describing PS in terms of the latter is beneficial due to the resulting simplification,

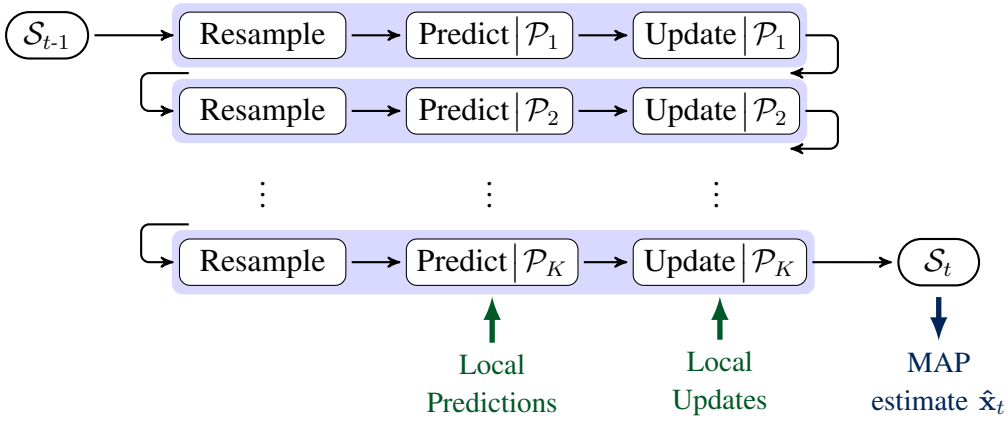


FIGURE 4.7 One timestep of Partitioned Sampling. Each partition is estimated using the resampling, prediction and update pattern known from the SIR particle filter (Figure 4.2). However, predictions and updates for each partition are local, *i.e.* only parts of the state space corresponding to the current partition $\mathcal{P}_k$ are estimated. The particle set $\mathcal{S}_{t\text{-}1}$ evolves into the new particle set $\mathcal{S}_t$ by traversing all partitions sequentially. The order of partitions is subject to kinematic precedence relations.

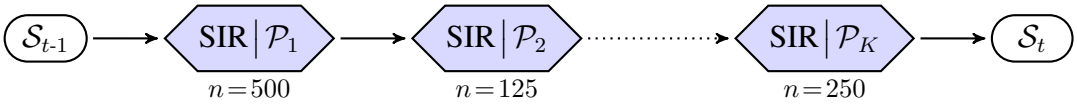


FIGURE 4.8 Simplified Representation of one timestep of Partitioned Sampling as sequentially coupled SIR filters, each working on parts of the state space only. The number $n$ below each SIR node represents the number of particles used for estimating the respective partition. As shown, the number of particles can vary and is adaptable to the complexity of each partition.

as illustrated in Figure 4.8.

We will now discuss the necessary prerequisites that have to be fulfilled in order to apply PS. When the following conditions hold, PS is a probabilistically correct and more efficient alternative to the simpler SIR particle filter:

1. The state space $\mathcal{X}$ can be partitioned as a Cartesian product of disjunct subspaces $\mathcal{P}_k$:

$$\mathcal{X} = \mathcal{P}_1 \times \mathcal{P}_2 \times \ldots \times \mathcal{P}_K \quad ; \qquad \mathcal{X} = \mathbb{R}^d \tag{4.38}$$

   This trivial equation holds whenever the dimensionality $d$ of the state space is larger or equal to the number of partitions $K$.

2. The dynamics $\mathbf{x}_t = g\left(\mathbf{x}_{t\text{-}1}\right)$ as defined by the motion model $p\left(\mathbf{x}_t \,\middle|\, \mathbf{x}_{t\text{-}1}\right)$ can be decom-

posed as follows:

$$\mathbf{x}_t = g\left(\mathbf{x}_{t\text{-}1}\right) = g_K\Bigg(\ldots\Big(g_2\Big(g_1\big(\mathbf{x}_{t\text{-}1}\big)\Big)\Big)\Bigg) \tag{4.39}$$

Here, $g_k$ corresponds to the dynamics that act on partition $\mathcal{P}_k$ to predict the associated parameters. A necessary condition for $g_k$ is that it does not modify the parameters belonging to the preceding partitions $\mathcal{P}_i$ where $i<k$.

3. The observation model $p\big(\mathbf{y}_t \,\big|\, \mathbf{x}_t\big)$ can be defined locally to evaluate the predictions induced by the dynamics $g_k$ that are local to the partition $\mathcal{P}_k$. In other words, it must be possible to define a weight function $h_k$ that is peaked in the same regions as the *posterior* pdf projected to $\mathcal{P}_k$.

Fortunately, these prerequisites are fulfilled in the context of articulated human pose tracking: the pose parameter space can be partitioned as a Cartesian product of joint angles, the dynamics of joint angles do not influence the dynamics of hierarchically preceding joint angles, and the weight function can be evaluated locally for each body part. The latter is achieved by rendering a partial version of the human model that consists of all body parts predicted so far. The modified projection mask $I_P$ is then plugged into Equation 4.26 to evaluate the error for the currently predicted pose parameters only (recall that the error is defined as mismatch between projection and segmented foreground, *i.e.* it will be smaller when the partial projection explains more of the foreground).

For articulated human pose estimation, meaningful partitions are given by rigidly moving body parts, such as upper or lower arms. Note however that partitions must not necessarily correspond to the body part definitions in our human model as shown in Figure 3.2. Often it makes sense to combine several body parts into a single partition (*e.g.* the shoulder plate `SBL` and the upper arm `OAL`). Sometimes it is also advisable to separate parameters that belong to a single body joint into two partitions. As an example, we associate the torsion parameter (rotation around the $t$-axis) of the upper arm `OAL` with the partition of the lower arm, because the torsion is best observed by its influence on the flection direction of the lower arm. Figure 4.9 depicts the PS strategy and the corresponding partitioning of our human model.

The main advantage of PS is the hard partitioning of the state space $\mathcal{X}$. By splitting the search space into manageable chunks, the particle growth can be kept within linear bounds, as opposed to traditional SIR particle filtering. A single high dimensional estimation problem is thus transformed into several lower dimensional problems that are easier to solve. Another

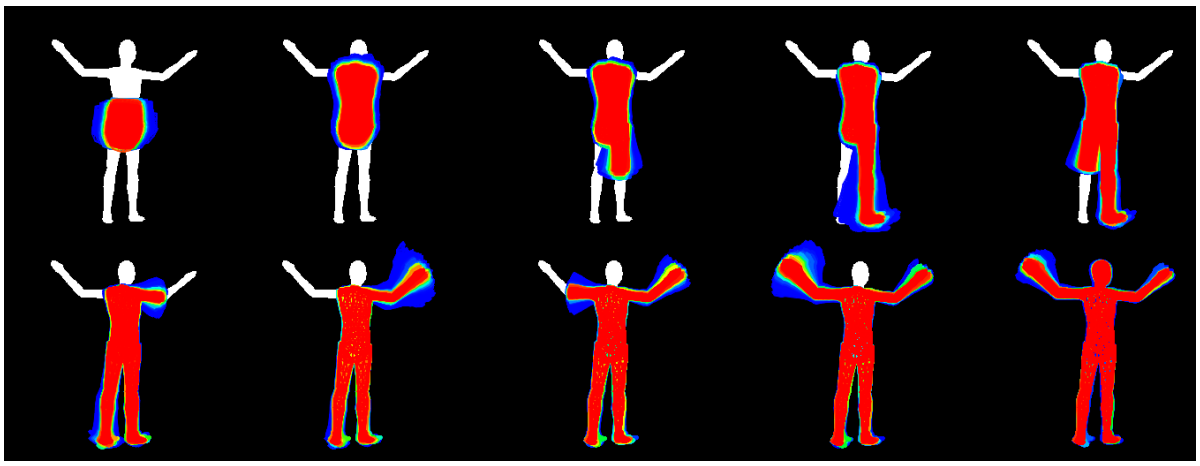

FIGURE 4.9 Visualization of the particle set during the progression of partitioned sampling with 11 hierarchical divisions (the 6th partition is omitted). Particles are rendered with their corresponding pose estimate, and higher color temperature corresponds to higher weights. Only the parts of the state space estimated so far are rendered. The order of partitions is based on the kinematic precedence in the model, *i.e.* the torso is estimated first before traversing the limbs and the head sequentially.

favorable characteristic of PS is that the number of particles can be varied in each partition (Figure 4.8). This makes it possible to distribute particle evaluations in an efficient way by allocating more particles to larger and more uncertain partitions.

A downside to PS is that the individual partitions must be reasonably small to be manageable by SIR filters, which can be problematic when *e.g.* the torso needs to be split into several partitions. This often leads to bad estimates due to the resulting ambiguities in the local weight functions (such cases are in fact violations of the aforementioned condition 3 for PS, as the local weight functions do not peak in the same regions as the posterior). What makes this worse is the fact that PS is unable to recover from bad estimates, which get propagated to subsequent partitions without chance of recovery, even in the light of new information.

## 4.5 Branched Iterative Hierarchical Sampling

As has been shown in practice, the sampling methods discussed in Section 4.4 suffer from several short-comings when applied in the context of human pose estimation. We will present proof of these limitations in our empirical evaluation later on in Section 4.6. Let us now discuss the main problems of the sampling strategies presented so far to motivate the novel hierarchical sampling strategy that we propose in this section:

1. The SIR algorithm as presented in Section 4.1.3 is intractable when the dimensionality

of the problem exceeds $\sim 10$ $d.o.f.$, due to the exponential growth in the number of particles needed with growing dimensionality $d$.

2. The APF algorithm as presented in Section 4.4.1 can be used to soften this constraint by focusing particles near the modes of the distribution. However, the necessary number of particles to is still subject to exponential growth with $d$, as the whole state space is explored at once. Covariance scaled diffusion (Section 4.4.1.1) alleviates this fact by focusing computation on areas with high uncertainty, which can be thought of as a mechanism for soft partitioning. In practice, human pose estimation with APF and covariance scaled diffusion works reasonably well up to about $25$ to $30$ d.o.f., but quickly becomes intractable as the dimensionality is further increased.

3. Another problem with the APF algorithm that we regularly observed during our experiments is that the estimates for the limb parameters tend to be incorrect. The cause of this is the inability of APF to model kinematic dependencies. As all parameters are continuously being estimated at once, high uncertainties associated with kinematically preceding parameters lead to pointless estimates for the dependant parameters. Over the course of several iterations, this effectively leads to the loss of all prior information from the last timestep (in other words, the parameters are randomized). An intuitive example is the flection parameter of an elbow that cannot be estimated reasonably without knowledge about the position and orientation of the upper arms.

4. The PS strategy as presented in Section 4.4.2 is able to confine the required particle numbers within linear bounds as the dimensionality $d$ increases. It is therefore possible to use it for tracking tasks where the dimensionality exceeds the capabilities of the SIR or APF algorithms. PS builds upon the fact that pose estimation problems based on articulated structures (*e.g.* human skeletons) are well suited for being split into smaller subproblems that are bound to local partitions of the pose space. The downside of PS is that it is prone to errors in early partitions, from which it is unable to recover even in the light of new evidence from subsequent partitions. It requires accurate and unambiguous local observation models for each partition, that are often difficult to come by. PS for human pose estimation works best when the human model accurately resembles the tracked subject, but it is highly susceptible to noisy observations.

5. As PS corresponds to a series of sequentially coupled SIR filters, the individual partitions have an upper limit in dimensionality that is tractable. The torso of the human model presented in Section 3 has already $16$ d.o.f. (including shoulder blades), so that it

needs to be split into two separate partitions. However, accurate local evaluation of the lower or the upper torso is difficult, as their shape is ambiguous.

Based on these insights, we propose a novel probabilistic sampling strategy which we will term *branched iterative hierarchical sampling* (BIHS). It is a combination of a *multi-layered search* strategy (*i.e*. APF) with *hierarchical partitioning* of the search space (*i.e*. PS). In addition, it splits particle evaluations into *parallel partitioning schemes* to overcome distracting local maxima of the weight function when the observation model is noisy. The result is a reliable and accurate sampling strategy for articulated human pose estimation that is capable of retrieving the global maximum of the observation weight function despite the high dimensionality of the state space. We also show how to automatically set the particle and iteration count for each partition using *adaptive sample sets*. Besides simplifying algorithmic development, adaptive sample sets can increase the computational efficiency without loss of accuracy by adapting these parameters based on the uncertainty of the current state estimate in each iteration. In the next sections we will discuss the individual concepts of BIHS in greater detail.

### 4.5.1 Multi-Layered Search and Hierarchical Partitioning

The core idea behind our novel sampling strategy is to combine APF and PS in a way that their individual strengths complement each other. As we have shown in Section 4.4.2, PS can be described as a sequential coupling of standard local SIR particle filters. By replacing the SIR filters (Section 4.1.3) with annealing particle filters (Section 4.4.1), the dimensionality of the local partitions can be increased. As discussed, APF works well up to about $25$ to $30$ d.o.f., which enables us to increase the maximum dimensionality for individual partitions by a factor of $\sim 3$, considering that SIR filters require that the maximum dimensionality is $< 10$. Increasing the size of local partitions is crucial for human pose estimation, as small partitions especially in the torso area are difficult to evaluate unambiguously, and resulting inaccuracies get propagated to successive partitions. Figure 4.10 sketches the basic principle behind this combination strategy, that we will term as *iterative hierarchical sampling* (IHS).

IHS outperforms both APF and PS when used with simulated, noise-free observation data, as will be shown in Section 4.6.1. In our experiments, a large initial partition of all torso parameters including the thighs and head was iterated over $10$ layers before estimating the remaining limb parameters in smaller partitions that were annealed using only $3$ layers (Figure 4.11). This strategy is much more effective and reliable at finding the global maximum of the weight function than APF or PS alone. When compared to APF, the main advantage is the hard partitioning of the state space that limits the number of particles that is necessary for

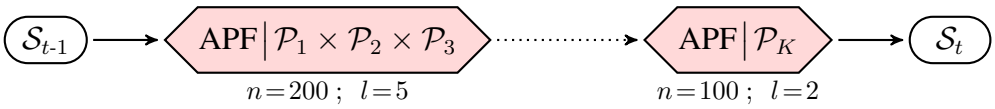


FIGURE 4.10 One timestep of *iterative hierarchical sampling* (IHS) that combines the principle of *partitioned sampling* (PS) with *annealed particle filtering* (APF). The PS algorithm (Figure 4.8) is adapted by exchanging the SIR filters (Figure 4.2) with APF (Figure 4.5) partitions. This facilitates the use of bigger partitions in PS, that can often be evaluated more accurately due to less ambiguity in the local weight functions.

successful exploration. Furthermore, the partitioning takes into account kinematic dependencies of the articulated model, and thus prevents the aforementioned problem of randomizing the limb parameters over the course of several iterations. When compared to PS, the main advantage of IHS is that the size of the partitions can be set much larger, which enables us to choose partitions that can be evaluated independently of the remaining parameters at good accuracy. Intuitively, a silhouette resulting from a projection of the complete torso including thighs and heads is much more distinctive when viewed from different directions than the lower torso only. Therefore, chances of incorrect estimates that propagate over all partitions are considerably reduced.

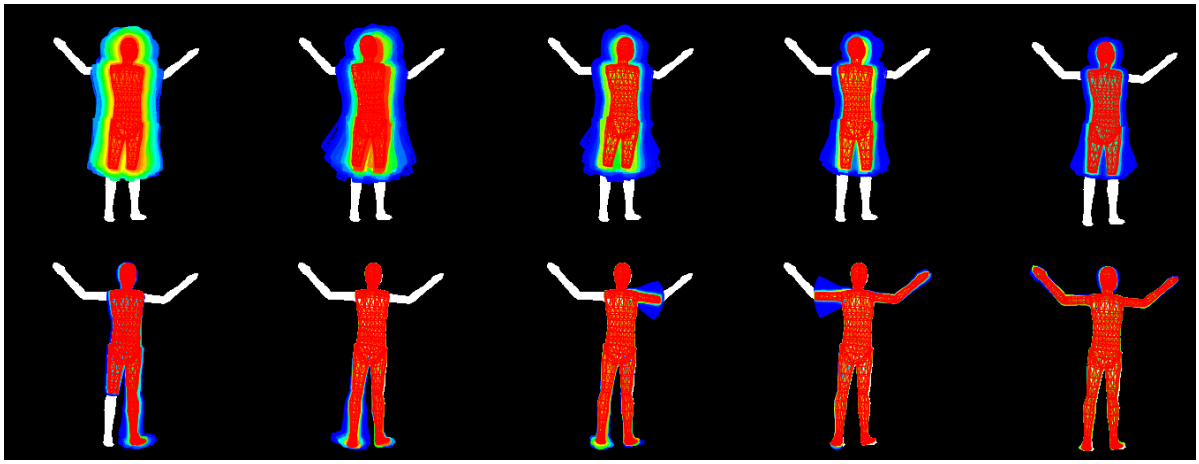

FIGURE 4.11 Visualization of the particle set during the progression of iterative hierarchical sampling (only selected steps are shown). Particles are rendered with their corresponding pose estimate, and higher color temperature corresponds to higher weights. Only the parts of the state space estimated so far are rendered. Notice the larger initial partition (when compared to Figure 4.9) that is estimated using an APF (first row) before descending the remaining limb partitions (second row). The remaining limb partitions are also estimated using an APF, but the number of iterations is reduced. The partitioning used in this Figure has been selected for illustration purposes and does not necessarily correspond to the best partitioning for human pose estimation.

### 4.5.2 Parallel Partitioning Schemes

IHS as proposed in the last section is an excellent hierarchical sampling strategy for human pose tracking when used with noise-free observation data (see Section 4.6.1). However, one of the remaining weak spots of PS is not addressed, namely its pure sequential hierarchical decomposition. Because of this, errors made will propagate to the remaining partitions without prospect of recovery. In perfect conditions, when the observations are free from noise, wrong estimates can only result from self ambiguities. These can be eliminated to large extents when using sufficiently many cameras (Section 4.6.1.1). In real tracking data, such perfect conditions are rarely encountered, and the sequential estimation of hierarchical partitions becomes a source of error.

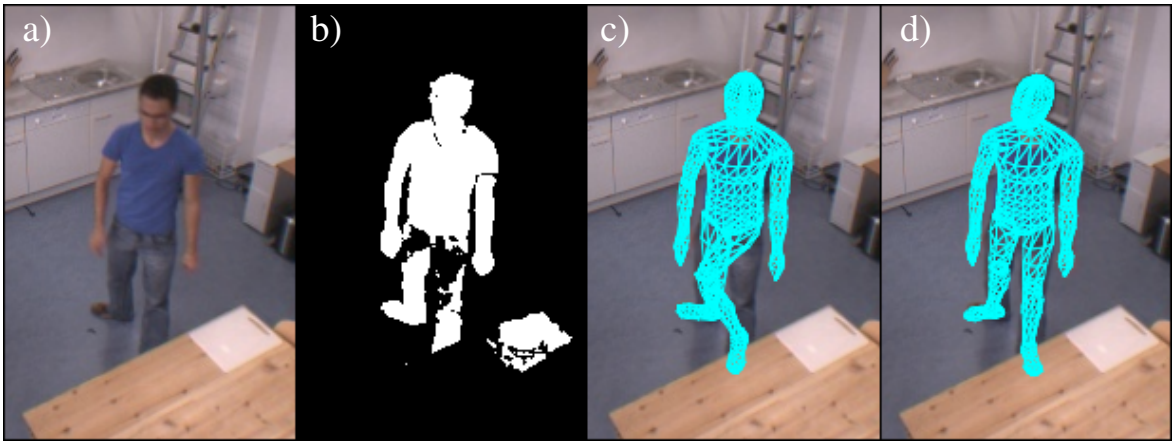


FIGURE 4.12 How the order of the limb partitions can influence the outcome when observations are noisy: a) original image b) noisy foreground mask c) left leg first partitioning scheme d) right leg first partitioning scheme. For this example, strategy d) works best.

Figure 4.12 shows an example from a real video sequence where noisy observations lead to estimation errors due to the sequential estimation in PS or IHS. In this example, most parts of the left thigh are unobserved in the extracted foreground silhouettes due to the similarity of the subject's pants with the background. When using a sequential estimation, where the sequence of partitions is *upper left leg* $\rightarrow$ *lower left leg* $\rightarrow$ *upper right leg* $\rightarrow$ *lower right leg*, the legs get incorrectly switched due to the error introduced when evaluating the upper left leg (Figure 4.12c). However, when the order of partitions is switched, *i.e.* the right leg is estimated prior to the left leg, the final estimate is correct (Figure 4.12d). In that case, the left thigh cannot be incorrectly associated with the right leg, as the corresponding foreground regions have previously been associated correctly with the right leg.

Based on the observations made in Figure 4.12, it becomes evident that the order of partitioning has a large influence on the final outcome of the pose estimation. Depending on image

noise, a wrong sequence of partitions can lead to particles being trapped in a local maximum of the observation weight function. Due to the sequential estimation in PS and IHS, there is no way to escape such a maximum.

To overcome this problem, we propose to divide particle evaluations into parallel pipelines where the partitioning sequences differ in the number and size of the partitions and in the order of evaluation. As an example, the estimation of the legs could be divided into three different partitioning sequences, one where the left leg is estimated before the right leg, one vice versa, and one where both legs are estimated in parallel. Afterwards, particles emerging from different partitioning sequences are combined and reweighted, so that the best strategy wins. The reweighting step can be seen as a voting for the best partitioning sequence, in that the particles with the highest weights emerge from the sequence that best explains the observation from a global perspective. In other words, while particles might get stuck in local maxima for some partitioning sequences, others might avoid these local maxima on the ascend towards the global maximum. The high diversity of local sampling strategies thus turns out to be the key to robust behavior in the presence of observation noise.

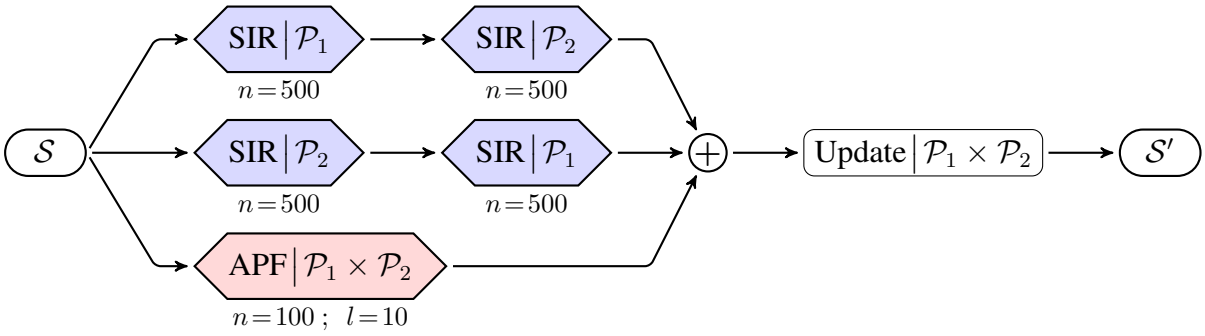


FIGURE 4.13 Branched partitioned sampling. Particle evaluations are split into several parallel partitioning sequences. As long as the overall parameters estimated by each sequence are identical, particles can be rejoined by creating the union of the branched particle sets (as denoted by $\oplus$). A final update step after rejoining the sets is necessary to recalculate the particle weights in relation to all other particles in the newly combined particle set according to Equation 4.27. In practice, the last update step in each branch before merging can be omitted to save computational resources. To simplify future diagrams, we will use $\uplus$ as a short notation for the combined step of concatenation followed by an update.

The aforementioned idea has been published by Maccormick and Isard [93] in a slightly different context as *branched partitioned sampling*. They applied it to improve occlusion handling in multi-target tracking, when the ordering of the tracked targets is unknown. Mitchelson and Hilton [99] presented a related hierarchical sampling strategy with parallel evaluations, but they are forced to use a less efficient heuristic for merging parallel particle sets as they allow

for parallel evaluations of differing and thus incompatible state partitions.

Figure 4.13 depicts the principle of branched partitioned sampling extended to IHS. When assigning $\mathcal{P}_1$ to the left leg and $\mathcal{P}_2$ to the right leg, the diagram corresponds to the previously mentioned legs example.

To be able to split particle evaluations into parallel pipelines as in Figure 4.13, some prerequisites have to be fulfilled. First, all of the partitioning sequences that are part of a parallel partitioning scheme have to provide an estimate for parameters corresponding to an identical subset of the state space $\mathcal{X}$. Second, the hierarchical dependencies of the parameters must not be violated in any of the partitioning sequences, *i.e.* hierarchical predecessors must be estimated first. When these prerequisites are met, the particles from each sequence can be rejoined by simply merging the branched particle sets. As the particles have estimated a similar subset of parameters, such an operation is valid and does not violate probabilistic correctness. A final update step after merging the branched particle sets is necessary to recalculate particle weights in relation to all other particles in the new particle set according to Equation 4.27.

While it would seem that evaluating several strategies increases the computational load, it is in fact advisable to split the number of particles for each partitioning pipeline depending on the branching factor $bf$, *i.e.* the number of parallel sequences. Therefore, the total number of particle evaluations stays constant, as does the computational load. Unless some of the sampling pipelines are extremely inefficient, the accuracy of the final result will be at least on par, as the same number of particles is used to estimate the same number of unknown parameters. At the same time, robustness and reliability of the algorithm in everyday scenarios is improved.

The fusion of concepts from the *iterative hierarchical sampling* (Section 4.5.1) with *branched partitioned sampling* is what we term *branched iterative hierarchical sampling*. We have applied this strategy successfully to the problem of human pose tracking, as validated by our experiments in Section 4.6 and Section 5.3.

It remains to be discussed how exactly one should partition the search space and proceed with the estimation, as there are several potential partitioning schemes that can be chosen. We propose a BIHS partitioning scheme for articulated human pose tracking that was modeled in a straightforward manner based on human reasoning. The complete scheme is depicted in Figure 4.14. In conjunction with the human model presented in Chapter 3 and the motion and observation models presented in Sections 4.2 and 4.3 respectively, this sampling strategy constitutes our approach on human pose tracking.

The estimation of body pose parameters in our BIHS strategy is split into three logical steps (Figure 4.14). The first step corresponds to the traditional APF as proposed in [46], but its

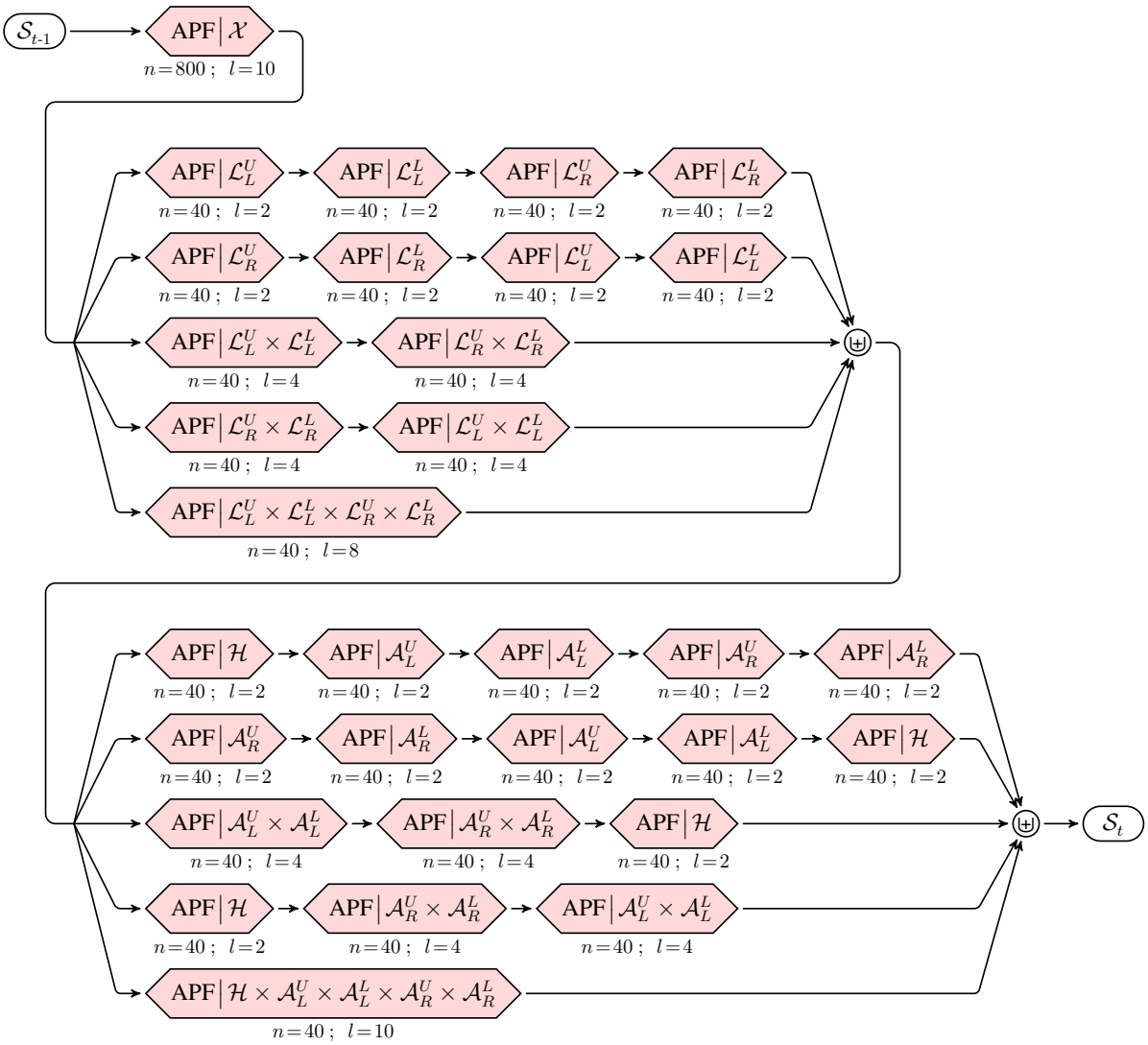


FIGURE 4.14 One timestep of branched iterative hierarchical sampling (BIHS) for human pose tracking. In the first step, an APF over the full state space $\mathcal{X}$ is used to get a good estimate of the torso parameters. While the APF over the full state space is good at estimating the torso parameters, it is not well suited for estimating the remaining limb parameters. These are estimated in two additional steps, using parallel partitioning schemes first for the legs and then for the arms including the head. The legs are estimated based on 5 different combinations of the minimal partitions *upper left leg* $\mathcal{L}_L^U$, *lower left leg* $\mathcal{L}_L^L$, *upper right leg* $\mathcal{L}_R^U$ and *lower right leg* $\mathcal{L}_R^L$. The upper limbs are estimated based on 5 different combinations of the minimal partitions *head* $\mathcal{H}$, *upper left arm* $\mathcal{A}_L^U$, *lower left arm* $\mathcal{A}_L^L$, *upper right arm* $\mathcal{A}_R^U$ and *lower right arm* $\mathcal{A}_R^L$. Using parallel partitioning schemes results in improved robustness when the observation data is noisy. Note that other partitioning schemes are possible, but the presented combination produced good results in all tested sequences.

aim is to estimate the torso parameters only. As already mentioned, the APF is very effective in estimating torso parameters (or more general in estimating parameters that are close to the kinematic origin), but less effective in estimating limb parameters. We have found that it is more effective to run the initial APF on the full state space $\mathcal{X}$, as the limb positions - even when inaccurate - provide additional cues for the quality of the torso estimate. In general, this strategy will quickly result in good estimates for the torso parameters, but the limb estimates will remain inaccurate. The next two steps consist of estimating the remaining parameters for the lower body (*i.e.* the legs) and for the upper body (*i.e.* the arms and the head). These two larger partitions are estimated in parallel partitioning schemes with $5$ different partitioning sequences each. The sequences are selected such that the order of partitioning and the size of the partitions differ substantially. The number of parallel sequences has been set to $5$, so that there is enough diversity in particle sampling without over-complicating the algorithmic design.

In the next section we will show how to automatically select the number of particles and iterations for each partition.

### 4.5.3 Adaptive Sample Sets

The BIHS strategy presented in the previous sections is very powerful when it comes to tracking of articulated structures such as human models (see Section 4.6 for experimental evaluation). Assuming that a rough initialization of the pose is given, our sampling strategy performs well on sampling the high-dimensional pose space and estimating the global maximum of the weight function, which given a suitable observation model should coincide with the true human pose. However, although our proposed sampling strategy is already more efficient than the current state-of-the-art Bayesian approaches presented in Section 4.4.1 and Section 4.4.2, tracking of human postures by using variants of particle-filters generally remains a computationally expensive approach. The computational costs are dominated by computing the particle weights, which typically involves costly image operations for every particle that is being evaluated. Considering that we need to evaluate particles over several iterations and hierarchies, the number of particle evaluations necessary to track a human model is in the order of magnitude of $10^4$. These account for most of the processing time, as we need to render and perform image to image comparisons for each particle ($\sim 20$ seconds per frame in our single-threaded implementation).

We will now show how to adaptively set the number of samples for each iteration to reduce particle count as far as possible without cutting back on accuracy. This also helps to simplify algorithm development by removing the necessity to specify particle and iteration count for

each partition manually.

For each annealing iteration, we can specify the initial particle count $N$ and the (maximal) iteration count $M$ as follows:

$$N = \frac{20 \cdot d}{bf} \tag{4.40}$$

$$M = \lfloor 2 \cdot \sqrt{d} \rfloor \tag{4.41}$$

These formulas based on the dimensionality $d$ of the current partition were derived empirically and have been shown to give good lower bounds throughout our experiments. When splitting evaluation to several branches as in Figure 4.13, the particle count is reduced by the branching factor $bf$, *i.e.* the number of parallel branches. One should also set a minimum particle count (*e.g.* 20) for each partition.

$N$ need not stay constant throughout all iterations. As the diffusion variance is reduced by a factor $\alpha_m$ in each iteration, particles are spread more densely and become more effective. This is certainly desired behavior, still $N$ is often unnecessarily high for late iterations, as it has to be chosen such that it does not miss out on peaked maxima during early iterations. A straightforward approach is to couple the particle count $N_m$ in iteration $m$ directly on the amount of diffusion reduction specified by the schedule $\langle \alpha_M, \alpha_{M-1}, \ldots, \alpha_m \rangle$. When setting $\alpha$ constant for each iteration, we can calculate a good value for $\alpha$ based on the aspired fraction $r$ which is the ratio of the standard deviation of the diffusion in the last iteration to the first iteration (we use $r = 0.1$ throughout):

$$\alpha = \sqrt[M]{r} \tag{4.42}$$

We can then calculate $N_m$ based on $\alpha$:

$$N_m = \alpha^{(M-m)} \cdot N \tag{4.43}$$

Note that although we constantly reduce $N$, the density of the particle set still improves with each iteration due to the reduction of the diffusion and the thus reduced extents of the particle distribution. This can be seen by looking at the measure $c_d$ for the effective particle count, which corresponds to the number of particles available for each dimension when distributing all particles according to an uniformly spaced grid across the state space:

$$c_d = \sqrt[d]{N} \tag{4.44}$$

For large dimensionality $d$, a (near) linear reduction of $N$ will result in a sublinear reduction of $c_d$, whereas the standard deviation in this dimension is also reduced (near) linearly, thus making the particles more effective.

In Section 4.4.1.1 we have shown how covariance scaled diffusion can be used to distribute particles more effectively. During this process, we have computed the eigenvalues $\lambda_i$. If we make sure that the absolute values of all state parameters are directly comparable (by computing the *z-scores* before computing the covariance matrix $\mathbf{\Sigma}t, m{+}1$, *i.e.* scaling each dimension to its standard deviation $\sigma$), the squareroots of the eigenvalues give us a notion on how *large* the state space will be in the next iteration. We could thus also couple $N_m$ to the average change of the eigenvalues:

$$N'_m = \frac{1}{d} \sum_{i=1}^{d} \sqrt{\lambda_i} \cdot N \tag{4.45}$$

Equation 4.45 improved over Equation 4.43 in that the reduction of $N$ is now directly coupled to the uncertainty of the state estimation. What is more, we can use it to terminate the iterative process once the aspired resolution is reached, *i.e.* when $\frac{1}{d} \sum_{i=1}^{d} \lambda_i < r$. In practice, we have observed that it is best to reduce $N$ by averaging over Equations 4.43 and 4.45:

$$N''_m = \frac{1}{2} N_m \cdot N'_m \tag{4.46}$$

We attribute this to the fact that $\mathbf{\Sigma}t, m{+}1$ cannot always be estimated precisely due to the undersampling especially in early iterations.

Fox [53] has presented an algorithm to adaptively set the number of particles in the context of robot localization and mapping by comparing the sampling distribution with the predictive distribution. However, this algorithm is not applicable to our problem due to the higher dimensionality and the large divergence between predictive and posterior distribution in human pose tracking.

Our experiments (Section 4.6.2) have shown that using the presented modifications, we can increase performance between $50$ and $100$ percent, while keeping the accuracy of the final estimate constant. Note that the modifications presented in this Section are equally applicable to the original APF sampling strategy presented in Section 4.4.1.

## 4.6 Experimental Evaluation

Evaluating algorithms for human motion tracking is a complex task in itself that needs careful consideration. Unfortunately, there is no single best way to approach the problem, and the best solutions that can be thought of are not always feasible. One of the main problems is the difficulty to come up with ground-truth data for the tracking sequences that would allow for the computation of a meaningful error measure. Other problems arise from the differences in human models and the variety of possible parameterizations for a human pose, which makes comparison between related approaches challenging.

One of the more obvious approaches for evaluation is the manual inspection of the tracked sequence. Usually, a rendered version of the human model in its estimated poses is overlaid on the original video, to provide visual feedback on the accuracy of the estimates. Especially in the multi-view case, where the human subjects were recorded by several cameras, such a method can provide the observer with a reasonably good understanding on the robustness and accuracy of a particular algorithm. However, it is unsuited for comparing different approaches among themselves, as the results will be dependant on human experience and judgement. Still, we will use this method as part of our evaluation as it provides the viewer with immediate graphic feedback that gives a clear indication on the quality of the tracking algorithm.

For a more scientific evaluation of the tracking accuracy, the availability of ground-truth data is essential. The problem here is the inability to come up with the true ground-truth data, as this would require knowledge about the exact joint locations of a human at all time during the tracking. Unfortunately, this task can only be solved approximately. One possible solution is to hand-label the joint locations in the image plane of all cameras, however this is a tedious and time-consuming task. In addition, labeling the joint positions independently can violate constraints on the inner shape parameters $\phi_I$ (*i.e.* limb lengths) of the used models. And finally, accurate labels can only be provided by people with a good understanding of human anatomy (*e.g.* clinicians), as the location of joints is not directly observable unless X-ray imaging is used. A better solution is to use commercial marker-based tracking systems [147, 146, 153]. However, such systems are intrusive and require complex setups for acquisition with additional manual post-processing of sequences. Furthermore, access to these expensive systems is often unavailable for researchers on a budget.

To evaluate the algorithms presented in this chapter, we take a two-step approach. First, we create simulated sequences from available *ground-truth* motion capture data by generating perfect observations based on the model projection of the ground-truth poses. Therefore, the ground-truth estimate corresponds to the global maximum of our weight function from

Section 4.3. The underlying idea is that this allows us to focus our evaluation on the quality of the sampling strategy without having to consider side effects resulting from the choice of observation model or noisy image data. We will present our evaluations on simulated sequences in Section 4.6.1. The second approach is to evaluate the methods on the HUMANEVA data set [136]. This data set consists of real video sequences with corresponding ground-truth measurements that have been retrieved by a commercial marker-based tracking system. The HUMANEVA sequences have been adopted by the community as a benchmark for different approaches to provide a common platform for comparison. We present our evaluations on this data set in Section 4.6.2.

A remaining issue is the choice of an error measure that should be used as an indicator of accuracy and for comparison with other approaches. A straightforward choice is to directly compare the pose parameters $\boldsymbol{\psi}$ (Equation 3.8) between the ground-truth and the estimated data. However, as $\boldsymbol{\psi}$ corresponds to joint angle values that are model specific, this makes it impossible to compare tracking approaches that have used human models with different kinematic structures. In addition, joint angle parameters are kinematically dependant, and adding up the joint angle differences will increase the error measure while in fact they might just be necessary to compensate inaccuracies resulting from preceding parameters. Thus, a large error accumulated over several small deviations in joint parameters might actually correspond to a good estimate of the current posture, while a smaller error from just a single parameter close to the kinematic origin might correspond to a poor estimate.

A better choice is to compare the Euclidean joint positions in the 3D world reference frame, as suggested by Balan *et al.* [14] and adopted in the HUMANEVA benchmarks [136]. The main advantages are that joints positions are independent and different human models can be compared by computing the mean 3D Euclidean distance error over all identical joints (*e.g.* shoulders, elbows, knees). For the human model we use, the joint positions correspond to the origins of the local part coordinate systems $\mathbf{H}_{\mathrm{bp}}$ as computed in Equations 3.1 and 3.2. Sometimes it is advisable to also calculate and compare the maximal 3D Euclidean distance error of all joints, as local errors such as a poor estimate of a single arm are often difficult to detect when looking at the mean error due to their small influence on the overall error. In summary, constantly small mean and maximum errors are excellent indicators for a reliable and accurate pose estimation method.

### 4.6.1 Simulated Data

A lot of research in the field of human pose estimation is targeted at improving the observation models to improve the overall tracking results. As a result, the influence of the sampling strat-

egy on the overall results is often underestimated. However, especially in high-dimensional tracking tasks such as articulated pose tracking, the sampling strategy has a significant impact on the tracking outcome. To put it differently, without the ability to sample the relevant parts of the state space $\mathcal{X}$, the best observation models are bound to fail.

We have created a test environment where we can focus our evaluation specifically on the ability of the sampling strategies to find the global maximum of the weight function. If the observation model is reasonably well chosen, this global maximum will be a valid estimate of the true (hidden) state that we seek. Our test environment is based on simulated sequences of model projections as seen from virtual cameras (Figure 4.15). The simulated sequences were created from ground-truth motion capture data by creating perfect model projections $I_P^\star$ with an instance of our human model. During evaluation, we then use the ground-truth model projections $I_P^\star$ as a replacement for the foreground masks $I_F$ (see Chapter 4.3 and Equation 4.26 in particular), thus removing the necessity for foreground segmentation by providing perfect, noise-free image observations. When using the same instance of the human model that was used to create $I_P^\star$ for the evaluation of different sampling strategies, we can ensure that the global maximum of the weight function given by Equation 4.27 corresponds exactly to the known ground-truth pose. Thus we can directly compare the efficiency of different sampling strategies by examining the ability to reach the global maximum of the state space $\mathcal{X}$.

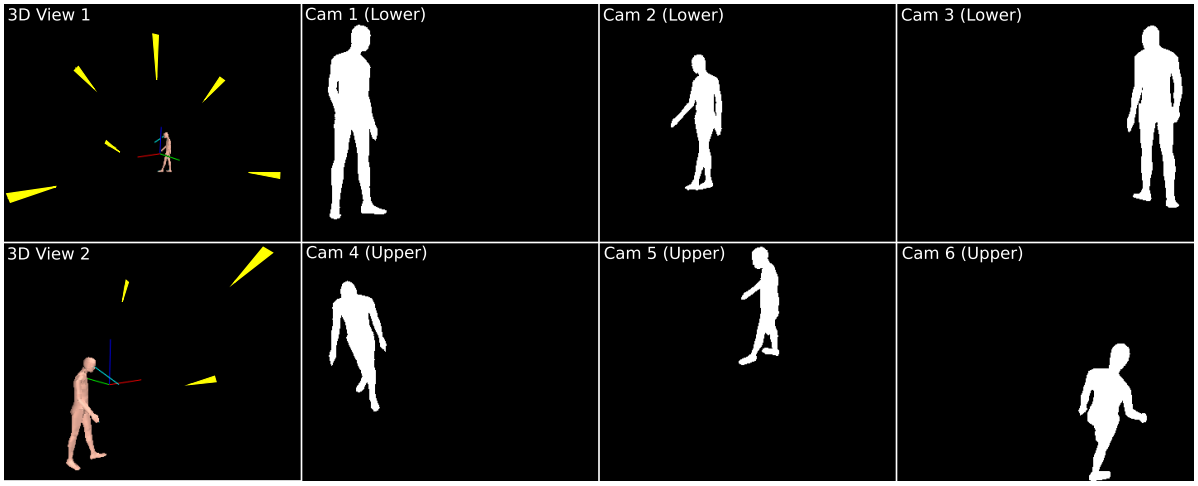


FIGURE 4.15 Simulated camera positions (first column) and the corresponding generated model projections $I_P^\star$ from ground-truth motion data. The model projections $I_P^\star$ are used to replace the foreground masks $I_F$ in Equation 4.26 to create perfect image observations.

As we do not have access to a commercial marker-based motion capture system, we have created the ground-truth from motion capture data that we recovered from real video sequences captured by three cameras. We recorded two sequences at 25 f.p.s., one that features only up-

per body motions with the lower body fixed on the spot (500 frames), and one that features a mixture of full body motions like walking and gymnastic exercises (1700 frames). The motion capture data was recovered using the PS approach from Section 4.4.2 with an exaggeratedly large number of particles (10000 particles for each of the 11 partitions). We have ensured the quality of the data by manual inspection and post-processing of obvious tracking errors. Furthermore, we used temporal smoothing as described in Section 3.4 to remove motion jitter. The resulting motion sequences are fairly realistic and provide a good basis for our evaluation. Note that although the sequences have been recovered by an algorithm that is part of our evaluation, we do not believe that the resulting evaluation will be biased towards that algorithm. The reason for our confidence is that all evaluated methods use the same motion model with random diffusion as presented in Section 4.2, and thus have the same initial capabilities to explore the state space $\mathcal{X}$.

Based on the simulated sequences, we have run a series of experiments to test several aspects of the hierarchical sampling strategies presented in this Chapter. We will discuss these experiments in the following subsections.

#### 4.6.1.1 Impact of Camera Count and Camera Placement on the Tracking Accuracy

When using the match between observed and predicted binary shape masks to determine the particle weights as proposed in Section 4.3, one has to take into account that different human postures can result in identical image projections, *i.e.* there is a many-to-one mapping between pose parameters $\boldsymbol{\psi}$ and binary projection masks $I_P$. This holds at least when considering only a single viewpoint, due to self-occlusions of the human body. To overcome these ambiguities, one must incorporate more than a single viewpoint into the observation model, which is then known as *multiple-viewpoint tracking*.

In our first experiment we chose to evaluate the number of cameras that are necessary for successful tracking when using only silhouette shapes for comparison. This is equivalent to determining the number of viewpoints that are necessary to resolve all possible ambiguities that can result from self-occlusions of the human body. Balan *et al.* [14] have performed a similar experiment suggesting that three cameras are needed for successful tracking, albeit they used real video sequences with noisy image observations and strong motion priors.

In our experiments we evaluated the influence that the number of virtual cameras has on the tracking accuracy by running the PS algorithm as described in Section 4.4.2 (with 5000 particles and 11 partitions) on our simulated full body motion sequence. We have varied the number of cameras from 1 to 6 between the experiments. All virtual cameras were places such that they differ in their viewpoint by at least 30°, and that no two cameras are opposite of each

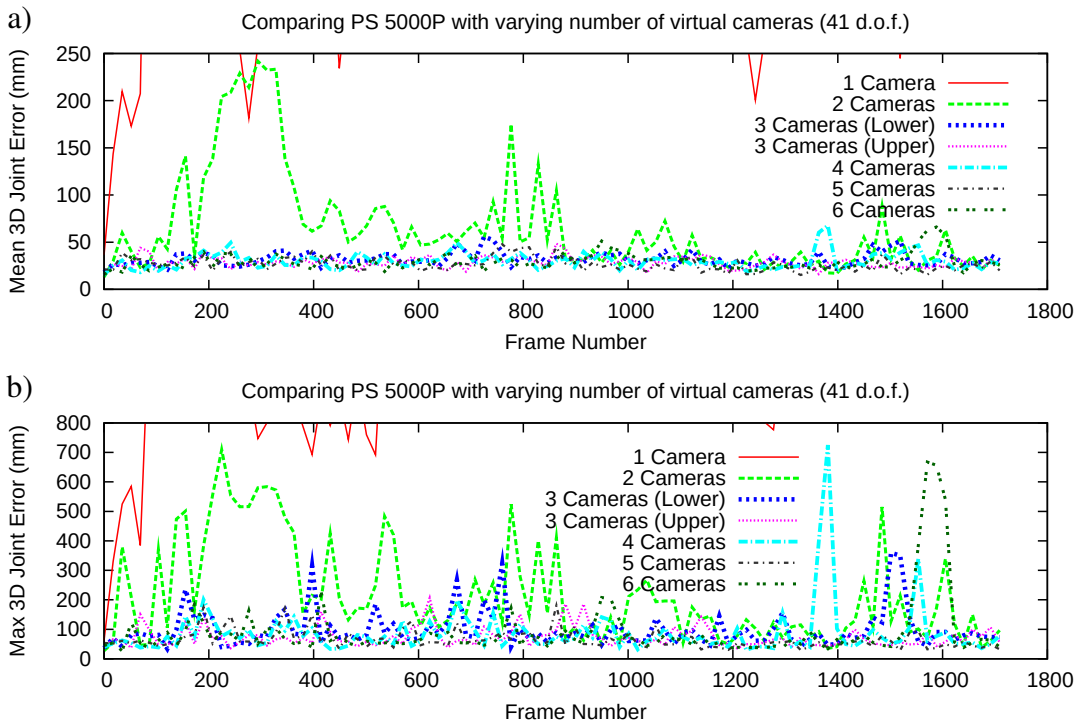


FIGURE 4.16 Evaluating the impact of the number of cameras on the tracking accuracy: a) mean Euclidean distance errors between estimated and ground-truth joint positions on a simulated sequence of full body motions (41 d.o.f.) for 1 to 6 virtual cameras; b) corresponding maximal Euclidean distance errors. Three cameras provide good accuracy, without further improvement when adding more cameras.

other, as that would result in mirrored silhouettes without information gain. We have placed three of the cameras parallel to the ground plane, and three other cameras as if they were mounted on the corners of a ceiling, pointing slightly down towards the scene (Figure 4.15). The latter is often favorable in real world scenarios, as it allows for a less intrusive setup. Therefore, we have been additionally interested whether one of the two camera placement strategies is superior.

The resulting mean and maximum joint position errors are shown in Figure 4.16. As can be seen, tracking with a single camera immediately fails, as occlusions of the limbs cannot be disambiguated and there is little indication on the correct depth of the pose. Tracking with two cameras also quickly fails, although the additional viewpoint clearly helps to improve the overall accuracy. When inspecting the tracking results visually in the video, the model at least seems to spatially follow the person around the scene in the two-camera case, as a result of the improved depth localization. We have tested several combinations of two cameras for tracking, yet none seemed to be particularly superior when compared to the others (the plot in Figure 4.16 contains only one of the two-camera selections for clarity). When adding

a third camera viewpoint, tracking suddenly seems to become stable. This observation is in accordance to the experiment conducted by Balan *et al.* [14], and suggests that three cameras that have a substantially different viewpoint are in fact sufficient for resolving the ambiguities resulting from self-occlusions of the human body. Interestingly, neither accuracy nor stability is further improved by adding even more cameras to the scene. What is more, our experiments indicate that the minimal achievable error is only determined by the capabilities of the hierarchical sampling strategies, and not by the number of cameras used for tracking. This is another clear pointer for the importance of the sampling strategy in human pose tracking.

Finally, the placement of the cameras seems to be negligible, as the tracking results for the lower three cameras and for the upper three cameras are approximately on par. Based on the conclusions drawn from the experiments presented in this section, we will use the three lower cameras for tracking in all of the remaining experiments on simulated sequences.

#### 4.6.1.2 Tracking of Upper Body Motions (21 d.o.f.)

In our next experiment we evaluate the accuracy of the hierarchical sampling strategies presented in Section 4.4 when compared to the standard SIR particle filter. The APF (Section 4.4.1) has so far been shown to provide good accuracy up to 34 d.o.f. [46] on a very short sequence ($< 4$ sec), while PS (Section 4.4.2) has been used for contour-based hand tracking with a comparably low 7 d.o.f. [94]. Our aim is to test the behavior at around 20 d.o.f., for which we created a simulated sequence where only the upper body of the human is moving. The sequence lasts for about 500 frames and features a human subject throwing darts at a wall, with subtle motions of the spine and shoulder blades as well as fast motions of the arms.

To provide a fair comparison of the different methods, we have set the parameters such that the number of particle evaluations is approximately comparable. For instance, APF with 20 layers and 2000 particles is comparable to SIR with 40000 particles, and also approximately to PS with 5000 particles and 7 partitions. On a side note, PS evaluations are often a bit faster as only parts of the model need to be rendered and evaluated for each partition. However, the necessity of partial rendering might reduce the potential for optimization in a specific implementation. Thus, a completely fair evaluation would also have to take into account implementation details that might affect the total running time of the algorithms.

Figure 4.17 depicts the resulting error plots for our experiment. It shows that both APF and PS perform very well with a mean error below 20 mm. As expected, standard SIR particle filtering performs much worse (despite 40000 particles). This can be associated to the exponential increase in the number of necessary particles as the dimensionality of the problem grows. Especially towards the end of the sequence when the motions become faster, SIR

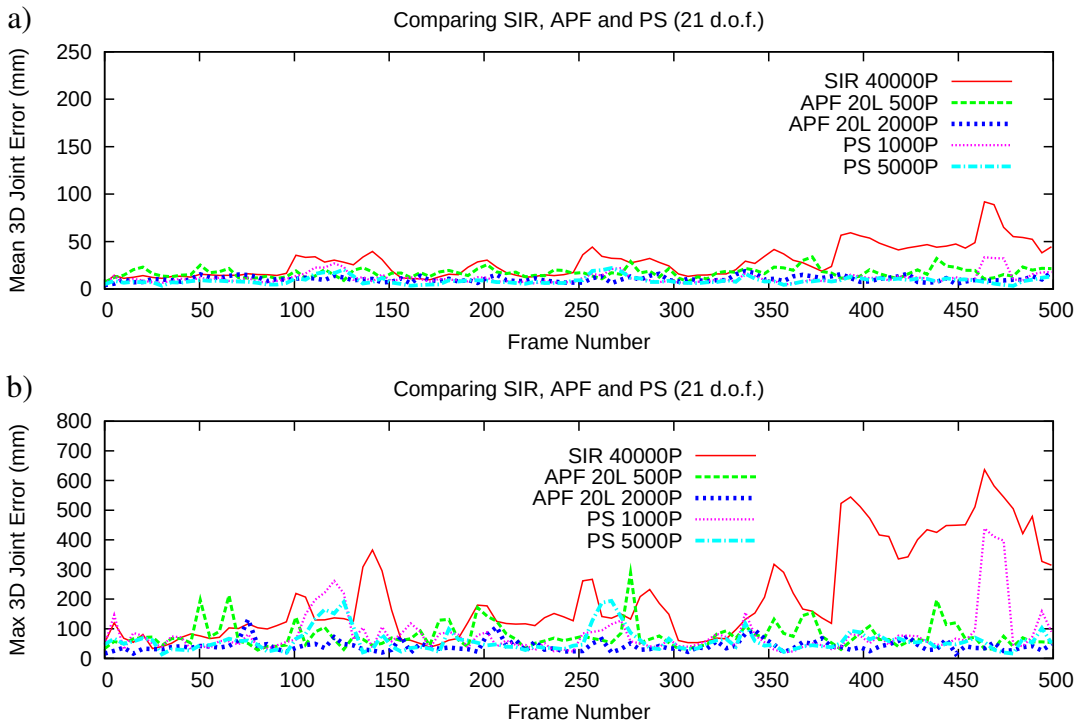


FIGURE 4.17 Evaluating the SIR, APF and PS sampling strategies on a simulated sequence with upper body motions (21 d.o.f.): a) mean errors between estimated and ground-truth joint positions for SIR (40000 particles), APF (20 layers and 500 respectively 2000 particles) and PS (11 hierarchies and 1000 respectively 5000 particles); b) corresponding maximal errors. While SIR fails towards the end of the sequence (faster motions), both APF and PS provide good results, even with reduced particle count.

looses its tracking while APF and PS succeed. Interestingly, when conducting the same experiment with only a fifth of the particle evaluations, both APF and PS are still able to track the sequence with only slightly increased mean errors.

#### 4.6.1.3 Tracking of Full Body Motions (41 d.o.f.)

The upper body tracking experiment presented in the previous section has shown that APF and PS are capable of tracking articulated structures with about 20 d.o.f. However, accurate human motion capture requires models with much higher complexity. Especially in biomechanical or ergonomic applications, simple human models around $20 - 30$ d.o.f. are unsuited when it comes to analyzing human motions. We have thus created a simulated full body tracking sequence using the 41 dimensional human model according to Figure 3.8b. Based on this sequence, we have evaluated the behavior of APF and PS when tracking such challenging models. Note that most of the d.o.f. in our model correspond to critical parameters in the

spine and the shoulders. These are much harder to estimate than *e.g.* hands or feet, that are easier to localize from the image observations $I_F$.

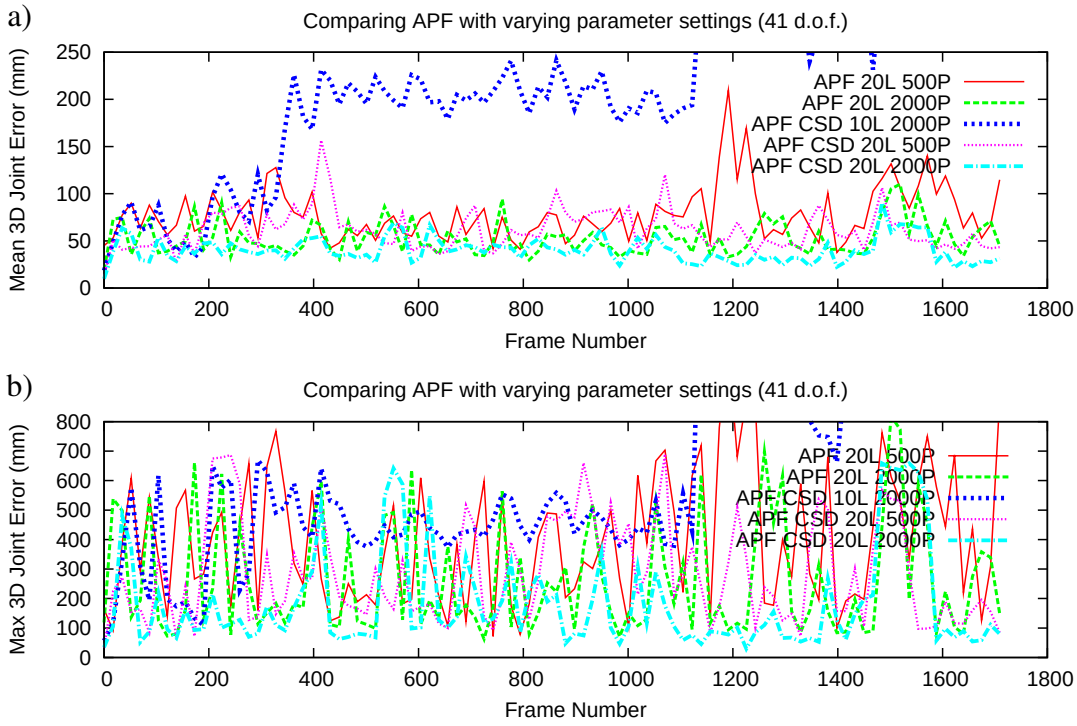


FIGURE 4.18 Evaluating the APF sampling strategies on a simulated sequence with full body motions (41 d.o.f.): a) mean errors between estimated and ground-truth joint positions for plain APF with 20 layers and 500 respectively 2000 particles, and for APF with covariance scaled diffusion (CSD) with 10 layers and 2000 particles or with 20 layers and 500 respectively 2000 particles; b) corresponding maximal errors. The APF strategy has a hard time in this high-dimensional tracking task. More layers are clearly beneficial, as is CSD.

Our first experiment on the full body sequence is aimed at evaluating APF strategies with different parameterizations. Therefore, we have varied the number of particles and layers, and we have investigated the effect of *covariance scaled diffusion* (CSD) as presented in Section 4.4.1.1 on the tracking outcome. The resulting plots are shown in Figure 4.18 for both the mean and the maximum 3D Euclidean joint distance errors. As can be seen, the APF strategy clearly suffers from the increased dimensionality. Even the mean errors show several spikes, which is an indication for tracking loss at a large scale (*i.e.* several body parts loose track, or the body orientation is reversed). The accuracy is improved when increasing the number of layers, and also by utilizing CSD. The best result was achieved when using APF with CSD and 20 layers and 2000 particles. In this case, the mean errors are reasonably small, but the maximum errors still show large spikes, which is attributed to one or more incorrectly estimated limbs. We attribute the observed difficulties to the absence of a hard partitioning in the

APF strategy, as all parameters are estimated in parallel. Furthermore, kinematic dependencies between parameters are not taken into consideration. CSD can alleviate these problems by providing a soft partitioning, *i.e.* the exploration of the state space is guided based on the uncertainty of the parameter estimates from the last iteration.

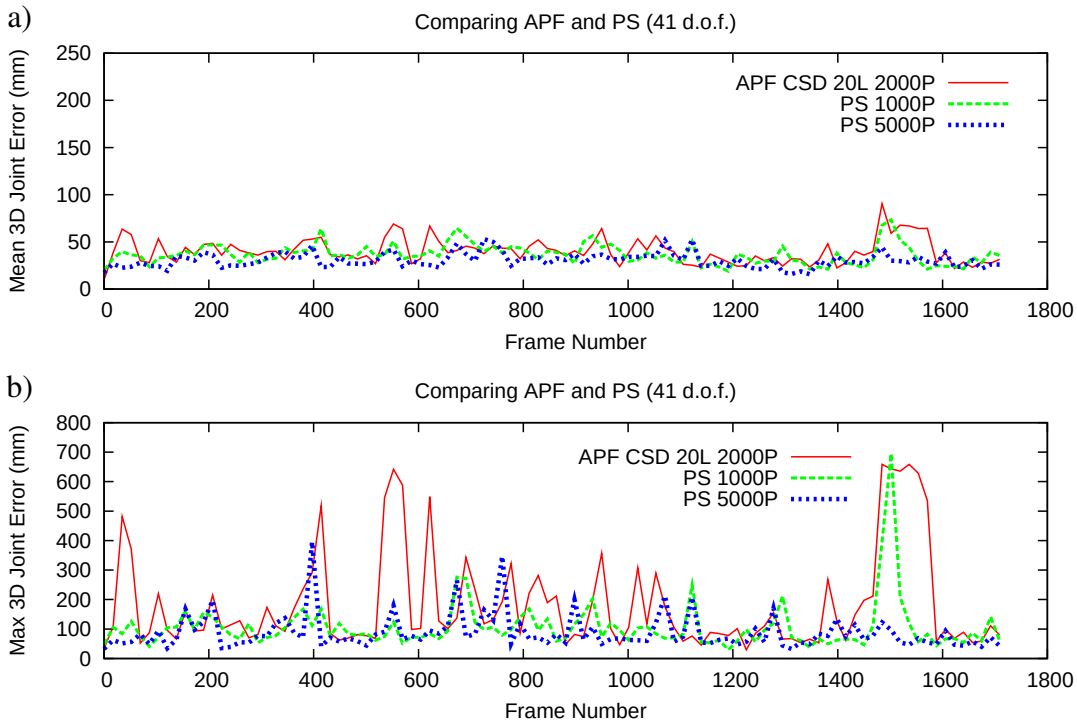


FIGURE 4.19 Comparing the APF and PS sampling strategies on a simulated sequence with full body motions (41 d.o.f.): a) mean errors between estimated and ground-truth joint positions for the best APF strategy from Figure 4.18 (CSD, 20 layers, 2000 particles) and for PS with 11 partitions and 1000 respectively 5000 particles; b) corresponding maximal errors. PS performs better in this experiment, even with reduced number of particles.

In a second experiment, we compared the most successful APF method found in the aforementioned experiment (*i.e.* APF with CSD, 20 layers and 2000 particles) with two instances of PS that use 1000 respectively 5000 particles with 11 partitions (upper/lower torso, upper/lower left/right leg, upper/lower left/right arm, head). The resulting plots are shown in Figure 4.19. While the mean error does not seem to favor either method, the maximum error shows that PS is more robust, even when using only a fraction of particle evaluations. Large maximum errors that coincide with small mean errors indicate partial tracking loss of body parts, *e.g.* when one of the arms is lost. We attribute the fact that PS outperforms APF at higher dimensions to the hard partitioning of the state space $\mathcal{X}$, which enables us to track high-dimensional articulated structures without the need for exponentially many particles. However, we should note that the PS strategy benefits from the accurate outer shape of the human model presented

in Chapter 3, which makes it easier to disambiguate neighboring body parts. We expect PS to perform worse when used with *e.g.* cylindrical human models.

We have also investigated different sequential partitioning schemes for PS, *i.e.* we have compared breadth first partitioning (where all limb hierarchies are descended alternately) with depth first partitioning (where each limb hierarchy is descended to the bottom before moving to the next limb). However, the order of partitioning did not have any influence on the accuracy or robustness of the tracking during our experiments.

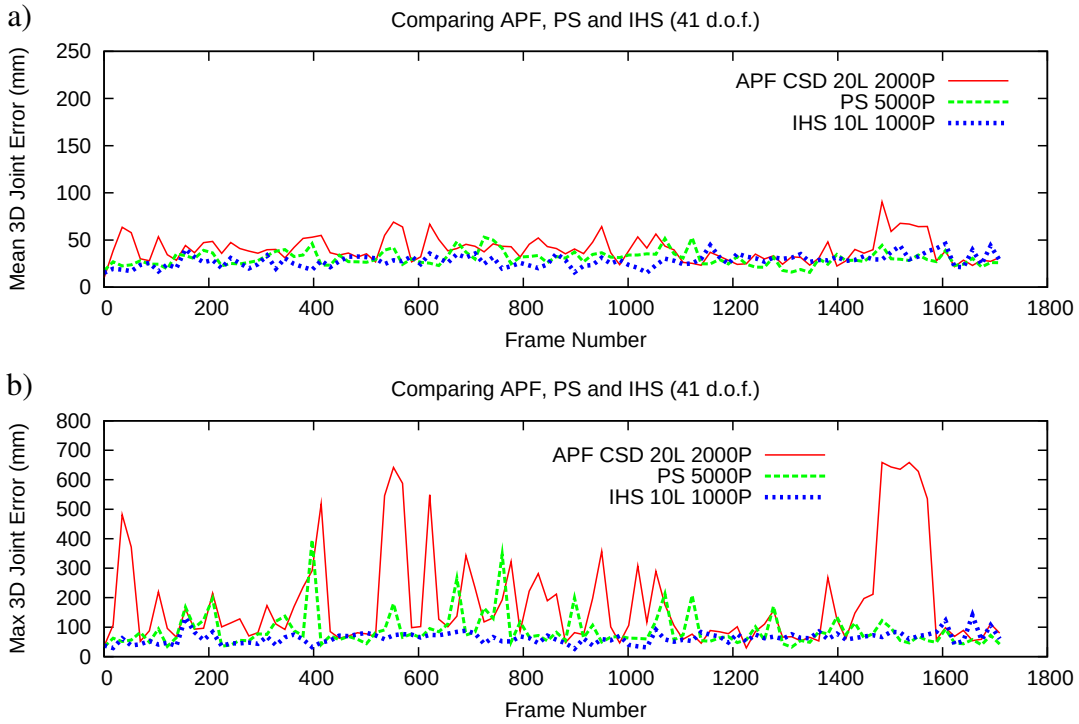


FIGURE 4.20 Comparing the APF, PS and IHS sampling strategies on a simulated sequence with full body motions (41 d.o.f.): a) mean errors between estimated and ground-truth joint positions for APF (CSD, 20 layers, 2000 particles), PS (11 partitions, 5000 particles) and IHS (10 layers and 1000 particles for the initial partition, 2 layers and 500 particles for the 9 remaining partitions, see Figure 4.11); b) corresponding maximal errors. IHS is very stable and outperforms both APF and PS, despite using less than half of the particle evaluations.

Based on the insights from the experiments so far, we have developed a novel sampling strategy termed *iterative hierarchical sampling* (IHS) that is described in Section 4.5.1. This strategy combines the complementary strengths of APF and PS by exchanging the SIR filters in each PS partition with APF. In our experiment, we have used IHS with an initial torso partition that comprises 19 d.o.f. and that is estimated using 10 annealing iterations and 1000 particles. The remaining 9 small limb partitions are estimated using 2 iterations and 500 particles, which is more than sufficient for partitions of this size. The sequential partitioning

scheme is also depicted in Figure 4.11.

The resulting boost in performance becomes evident when looking at Figure 4.20, where we have compared the best APF and the best PS parameterization from all our experiments with the newly proposed IHS strategy. Despite using less than half of the particle evaluations and running more than twice as fast than APF and PS, IHS clearly outperforms both of these. While the mean error plot is not clearly distinctive, the maximum errors show that IHS reliably tracks all of the body parameters over the full sequence. The mean joint errors are constantly below $50$ mm, and the maximum joint error is almost constantly below $100$ mm. According to a dependent t-test on the mean joint errors, IHS performs better than PS at $99.9\%$ confidence level, and PS performs better than APF at $99.9\%$ confidence level.

### 4.6.2 Ground-Truth Motion Sequences

The experiments from Section 4.6.1 have been conducted on simulated sequences and give a good impression about the behavior and accuracy of the presented sampling strategies. However, they are unable to give a definite comparison of these methods under real-world conditions. Real observation data is often noisy, cluttered and imprecise, which leads to undesired effects such as the one shown in Figure 4.12. Therefore, additional evaluation of the sampling strategies on real motion sequences is indispensable.

Real motion sequences with corresponding ground-truth data are difficult to come by. Their acquisition either requires hours and hours of cumbersome hand-labeling work for each image, or access to a professional marker-based motion capture system [147, 146, 153]. Fortunately, Sigal and Black [136] provide two publicly available data sets for evaluation of human motion capture algorithms that have been recorded with such a motion capture system. The first HUMANEVA data set consist of several sequences of single human subjects performing specific actions such as walking, jogging or gesturing in an empty room. It contains training, validation and test sequences of several subjects. The second, newer HUMANEVAII data set consists of two additional sequences (two subjects) for which only the test sequences are available. The ground-truth data is kept back by the authors, and evaluation of the tracking error is performed via a XML-based web-interface. This prevents the misuse of ground-truth data to alter the tracking results or to provide strong motion-specific priors in order to reduce the complexity of the problem.

For our evaluations, we decided to use the HUMANEVAII data set. Both sequences consist of a person walking in a circle, then accelerating into a jogging motion, and finally changing into a slightly unusual stretching motion (Figure 4.21). The transitions between the three types of motion are especially critical for tracking algorithms that rely on strong motion priors, *e.g.*

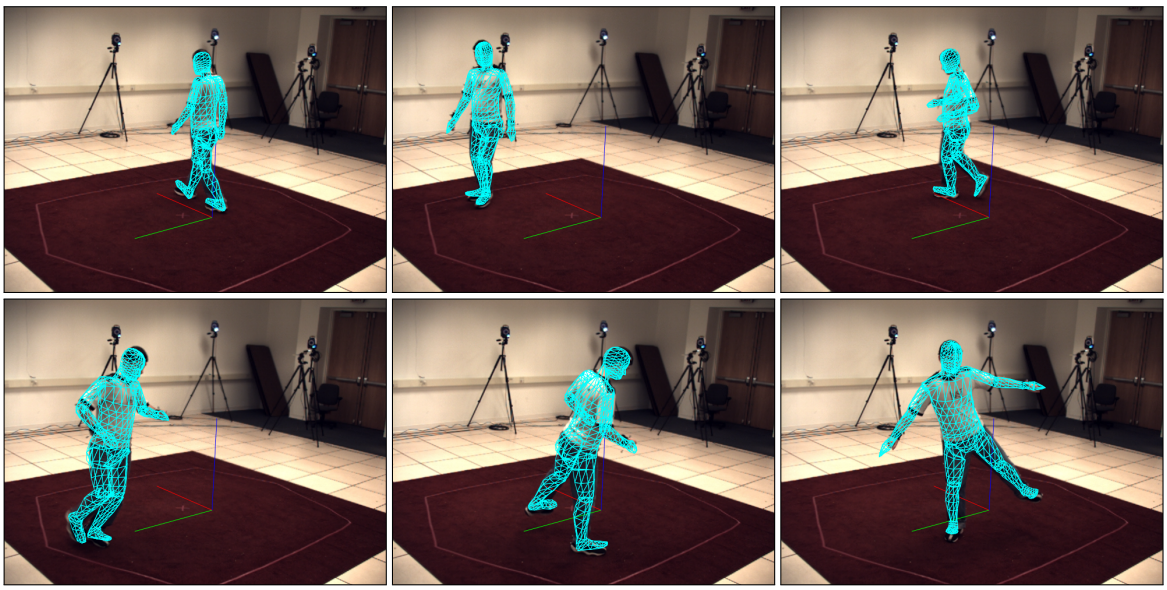

FIGURE 4.21 Tracking results on the HUMANEVAII S4 sequence for the BIHS strategy (10 layers, 800 particles). We show random frames for one of the four camera views.

when a motion model has been learned from exemplars of walking motions, the transition to jogging often fails. The sequences have been recorded by four cameras at a resolution of $640 \times 480$. The foreground segmentation is challenging due to color overlap between the clothing and the background. As a result, the extracted foreground masks $I_F$ are quite noisy (Figure 4.22). Furthermore, subjects wear casual clothing, and especially the pants are quite loose. We have adapted our human model manually to fit the observed human shapes as good as possible.

In our tests we used the unconstrained motion priors from Section 4.2 and the shape-based observation model from Section 4.3 to evaluate several variants of the APF, the PS and our proposed BIHS sampling strategies. The resulting mean 3D joint error plots for sequence S4 of the data set are depicted in Figure 4.23. Note that we have omitted the estimation of the lower spine parameters as this results in more stable estimates for most kinds of upright motions (we have done so in particular to favor the APF and PS strategies; BIHS is shown in Figure 4.24 to compensate well for the additional d.o.f.).

We first evaluated the APF (Figure 4.23a) with three different variants for setting the amount of diffusion applied to each body part with each iteration. In the initial version, the diffusion is reduced using a constant factor ($\alpha_M = \alpha_{M-1} = \ldots = \alpha_1 = 0.5$, see Algorithm 4.4). In the second version, the amount of diffusion is controlled using *variance scaled diffusion* (VSD), *i.e.* according to the state variance in the particle set from the last iteration. In the third version, particles are diffused using the state covariance matrix from the particle set in the last iteration.

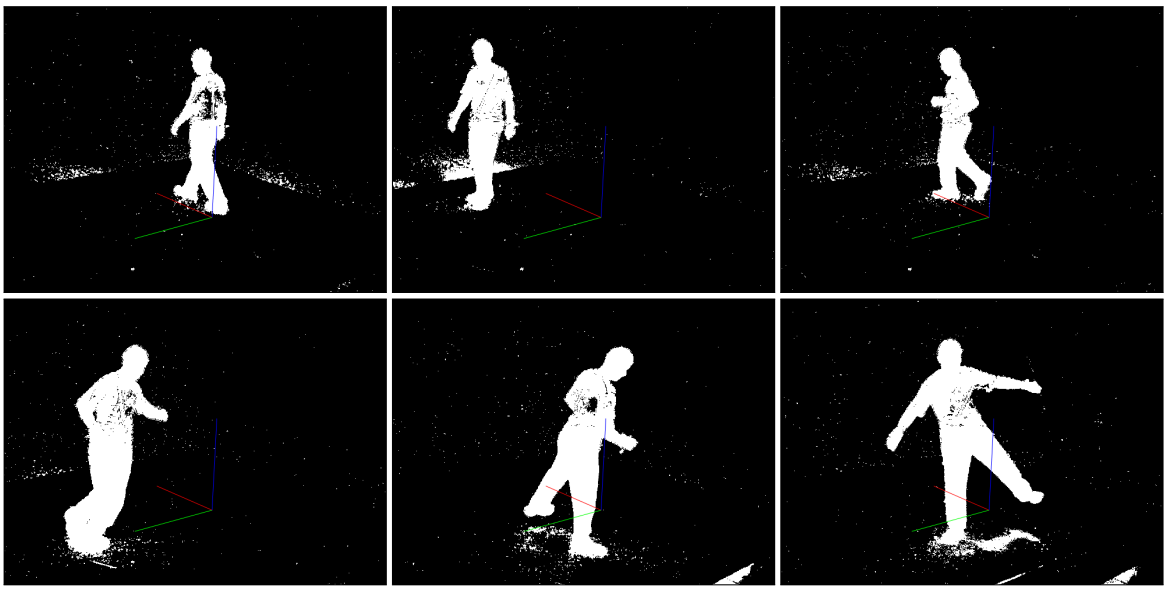

FIGURE 4.22 Foreground masks $I_F$ as segmented using the codebook method [80] for the HUMANEVAII S4 sequence. The masks correspond to the frames shown in Figure 4.21.

This corresponds to *covariance scaled diffusion* (CSD) as presented in Section 4.4.1.1. The last strategy also provided the best results, although the overall performance of the APF is unsatisfying. A visual inspection of the results shows that while the torso is correctly estimated most of the time, the limb positions are often wrong. This comes as no surprise, as it coincides with the observations made in our simulation experiments in Section 4.6.1.3.

We then tested PS (Figure 4.23b) at comparable processing times (PS with 1600 particles $\sim$ APF with 10 layers and 800 particles). The order of partitioning is pose and lower torso, upper torso, left upper leg, left lower leg, right upper leg, right lower leg, left upper arm, left lower arm, right upper arm, right lower arm, and head. The tracking results are again unsatisfying, which we attribute to the large ambiguity in the local weight function when evaluating the lower torso. We have observed that contrary to the results on the simulated sequences with perfect observations, the PS strategy is affected when the shape of the human model is an inaccurate approximation of the true human shape.

Finally, we have evaluated the novel BIHS strategy as presented in Section 4.5. This strategy has been directly derived from the observation that APF manages to find good estimates of the torso parameters but fails in estimating the limbs, while PS is good at estimating the limbs in case that it got the torso parameters right. When compared to the IHS strategy that we evaluated in our simulation experiments in Section 4.6.1, BIHS is capable to evaluate parallel partitioning schemes, which should give it increased robustness and accuracy in the case of noisy image observations (see Section 4.5.2). The error plots as depicted in Figure 4.23c

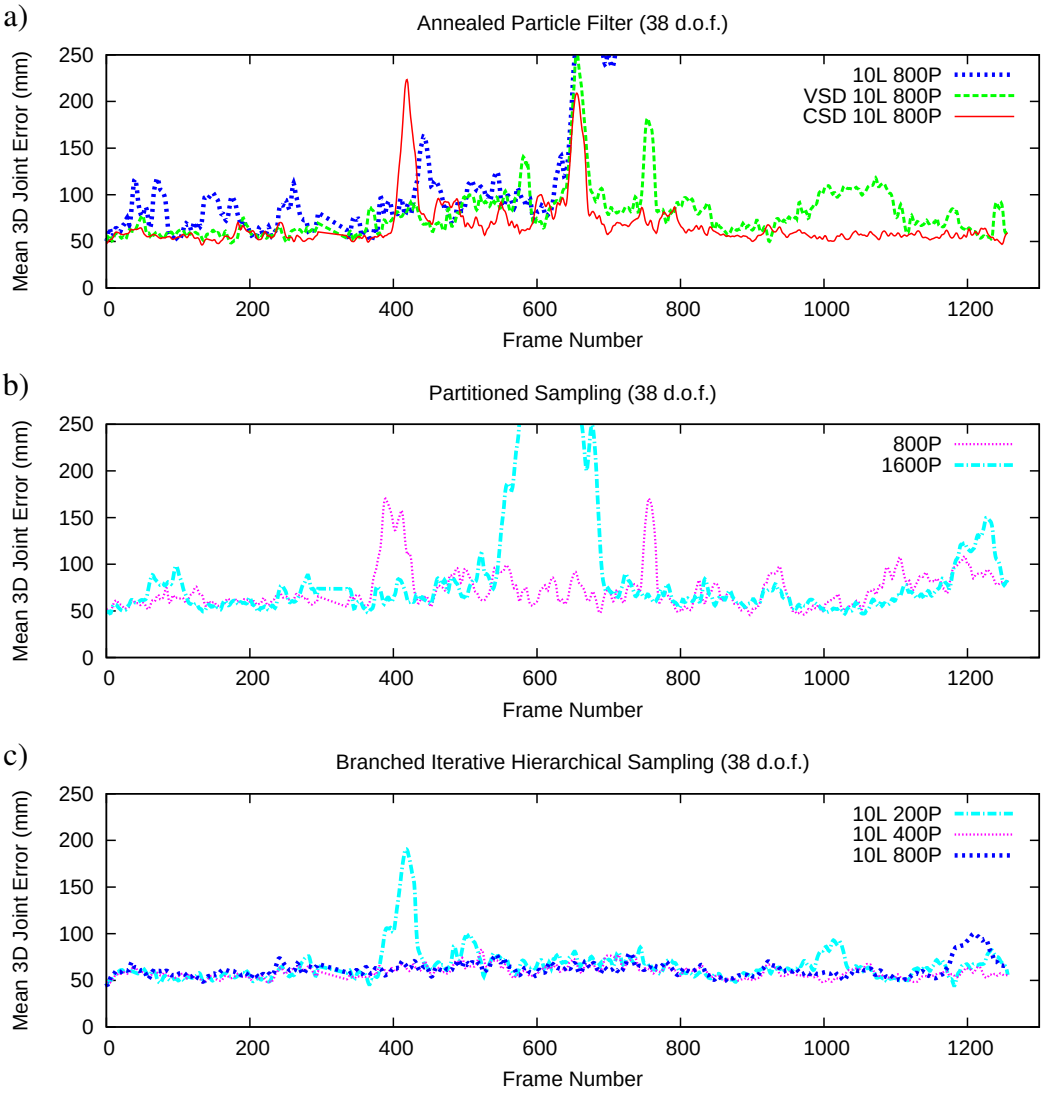


FIGURE 4.23 Tracking results on the HUMANEVAII benchmark (sequence S4) for: a) the APF with 10 layers and 800 particles, once without scaled diffusion, once with variance scaled diffusion (VSD), and once with covariance scaled diffusion (CSD); b) PS with 11 partitions and 1600 or 800 particles; c) BIHS with 10 layers and 800 or 400 or 200 particles. Even when using only about a quarter of the particle evaluations, BIHS outperforms APF and PS and shows robust tracking behavior despite noisy image observations. Note that there seems to be a systematic error (the error never drops below 5 cm) that can be attributed to the differences in relative joint positions between the ground-truth model and ours. Visually, our BIHS method delivers near perfect results (see Figure 4.21, the full video is available from http://memoman.cs.tum.edu).

confirm this assumption. Despite the often noisy and imprecise foreground masks $I_F$ that are segmented from the HUMANEVAII sequences (Figure 4.22), BIHS is capable of producing estimates with constantly low mean errors around 50 mm. Even variants that use only a quarter

of the particle evaluations when compared to APF or PS still give better results, although the robustness is slightly reduced. We believe that the true mean errors for the BIHS strategy are lower than shown in Figure 4.23c, but there seems to be a systematic error that we attribute to the differences in relative joint positions between the ground-truth model and ours. This could explain why the mean errors never drop below 50 mm during 1250 frames, an observation that is unlikely given chance. We believe that the true mean error is around 25 mm, which corresponds to the values observed for IHS in the simulation experiments (Figure 4.20).

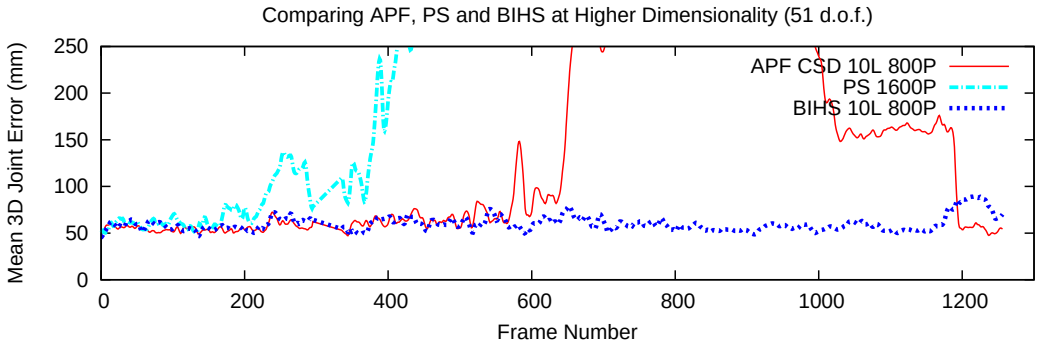


FIGURE 4.24 Comparing accuracy and reliability of APF, PS and BIHS when tracking the full 51 d.o.f. of our human model, including hands and feet. BIHS is nearly unaffected by the increased dimensionality when compared to Figure 4.23, whereas both APF and PS fail.

We repeated the experiments on sequence S4 using the best variants of each algorithm, this time tracking the full 51 d.o.f. of our model, including the lower spine, hands and feet. Both APF and PS completely loose track early in the sequence, while our BIHS strategy provides almost the same quality results as in the 38 d.o.f. case (Figure 4.24d). The reasons for this are the hard partitioning of the search space, where the additional parameters are estimated last without obscuring previous results. To the best of our knowledge, the presented BIHS strategy is the first Bayesian approach to have shown successful tracking of such high-dimensional articulated models without the use of training data. We should also note that all of the evaluations were processed without reinitializing the model pose, *i.e.* the sequences have been completely tracked at once. This is remarkable, considering that many competing methods fail during the transitions between walking and jogging, or between jogging and stretching, as many methods rely on strong motion priors due to inferior sampling or optimization strategies.

In addition to the experiments provided so far, we also present results of the BIHS strategy on the other sequence in the HUMANEVAII benchmark. In Figure 4.25 we show two tracking results on sequence S2 with two different model adaptations, and another tracking result on sequence S4 with different model than the one used in Figures 4.23 and 4.24. Although the results are still good, one can observe a slight decrease in accuracy for some models. In

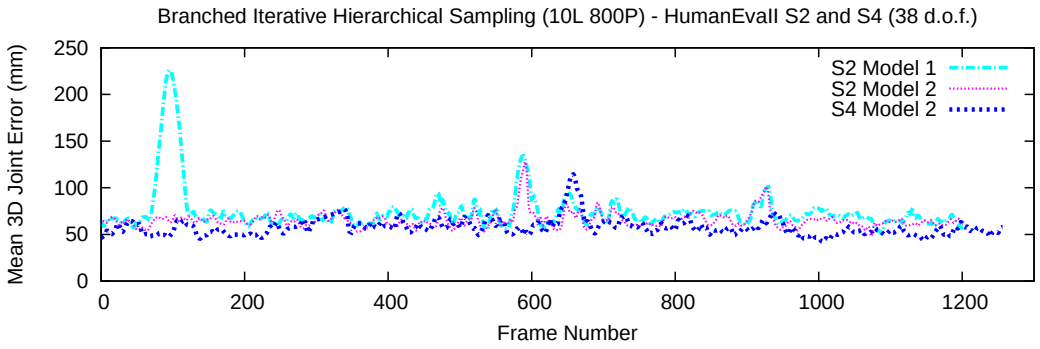


FIGURE 4.25 Tracking results for BIHS on both sequences (S2 and S4) of the HUMANEVAII benchmark in combination with different human models. The tracking quality can be affected when the inner shape parameter $\phi_I$ (*i.e.* the estimates of the limb lengths) are imprecise, as has been the case for model 1 for subject S2. Model 2 for subject S4 is also less accurate than the model used for the results presented in Figures 4.23 and 4.24.

these cases, the inner shape parameter $\phi_I$, (*i.e.* the limb lengths of the skeleton) were slightly inaccurate, which has a negative effect on the quality of the estimates. These results confirm our believe that accurate tracking of human motions requires the use of accurate models that have been adapted to the subject of interest.

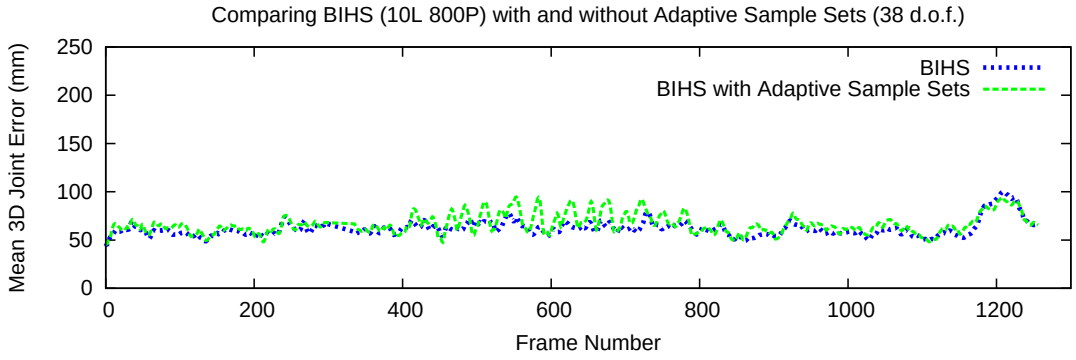


FIGURE 4.26 Comparing tracking accuracy of the BIHS algorithm with and without the use of adaptive sample sets. The variant that uses adaptive sample sets runs almost twice as fast.

Figure 4.26 shows a comparison between the BIHS algorithm with and without the use of adaptive sample sets as presented in Section 4.5.3. Adaptive sample sets are a means to improve the efficiency of the BIHS algorithm by coupling the number of particles and iterations for each partition to the uncertainty of the current estimate. The variant in Figure 4.26 has reduced the number of particle evaluations by almost a factor of 2 despite comparable accuracy and robustness.

As a final remark, we want to state that all experiments in Sections 4.6.2, and 5.3 have used the same parameterization for the motion and observation models, without learning or

tweaking the parameters towards specific sequences. For the HUMANEVAII experiments, this means that we have used larger inter-frame motion limits (Section 3.3.2) that are suitable for 25 Hz recordings even though HUMANEVAII has been recorded at 60 Hz. Thus we believe that by tightening the inter-frame motion limits, we could further improve the results on the HUMANEVAII benchmark when using fewer particles.

### 4.6.3 Other Motion Sequences

We have evaluated our methods on several other real video sequences that we have recorded but for which we do not have corresponding ground truth measurements. We will now give a brief overview on some of these sequences to prove the general applicability and robustness of our algorithms. As we cannot give quantitative results, we will evaluate performance based on the manual inspection of the resulting video sequences with overlaid model projections. The resulting videos are available in full length at `http://memoman.cs.tum.edu`.

One of the first sequences that we have recorded to test our algorithms is a 6.5 min sequence in an empty room that has been captured with 3 cameras at a resolution of $384\times288$ pixels and a sampling rate of 25 Hz. This corresponds to a total of over 10000 frames of motion. We have processed the sequence using our BIHS sampling strategy and the same parameterization of our methods as in Section 4.6.2. We were able to process the full sequence at once without the need for reinitialization of the model. Figure 4.27 shows selected screenshots from the resulting video sequence. The subject was told to perform random motions at his own will, ranging from walking to gymnastic exercises and clumsy dancing. Most of these motions have been correctly estimated at good accuracy. Some errors occurred during the dancing motions when the subject was quickly waving his arms in a random fashion. Our tracking algorithm was able to recover from these errors shortly after they occurred. Minor imperfections could also be observed when the subject was bending towards the ground, although the overall estimation quality of these challenging motions remains impressive. The processing time per frame in this sequence is about 10 sec (without any parallelization).

We have also collaborated with sports scientists to record video sequences of gymnastic exercises performed by professional gymnasts. We have recorded two subjects, a male and a female, during floor exercises. Due to the large spacial extent of the exercises that were performed in a gym, we used three consumer cameras with *Full HD* resolution ($1920\times1080$) that we placed far away from the scene to capture the subjects at all time without having to move the cameras. Processing at this high resolution takes about 25 sec per frame due to the computational efficiency of the runlength-coded region masks used in our observation model (Section 4.3). This corresponds to a sublinear increase in processing times with the number

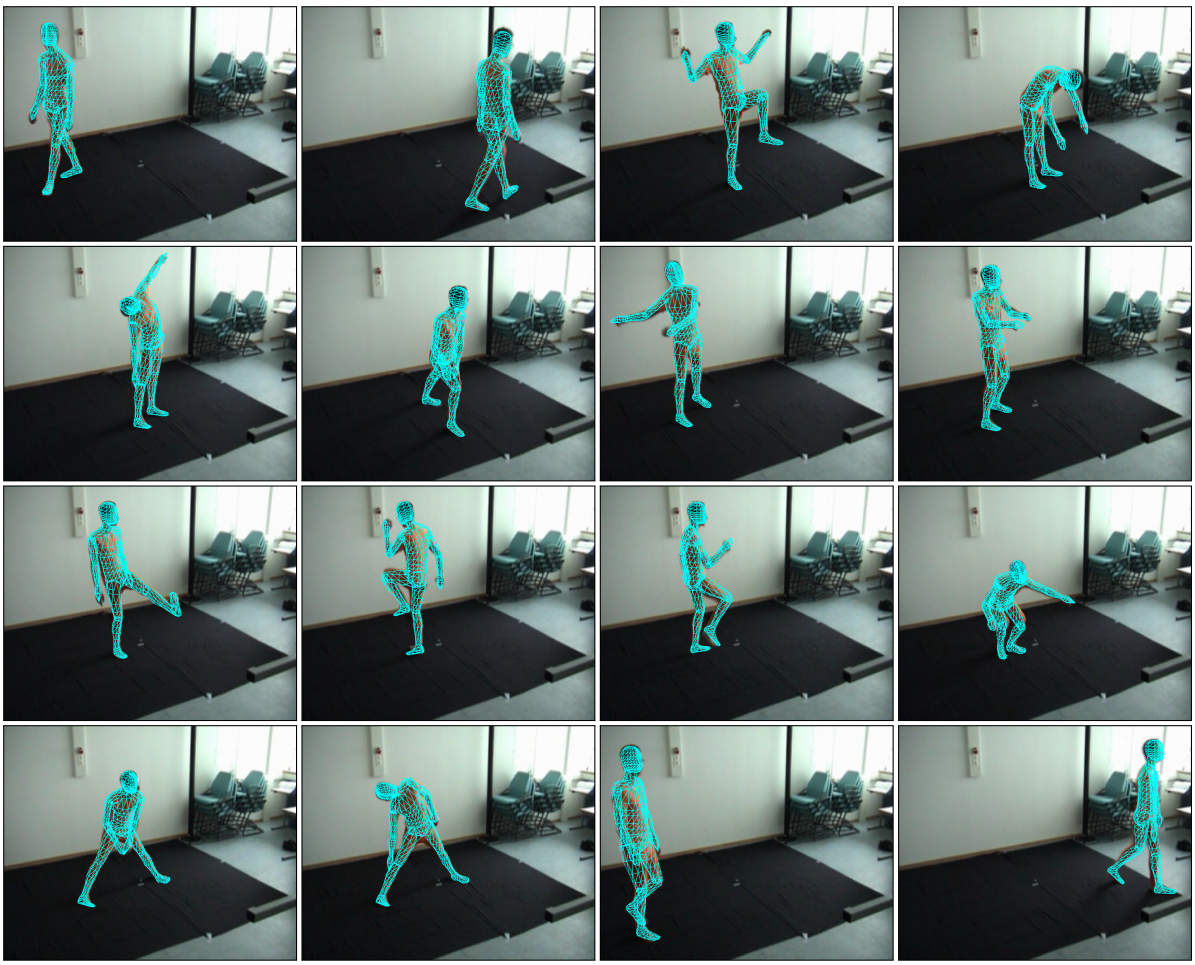

FIGURE 4.27 Screenshots from a 6.5 min sequence of random motions. The full sequence (over 10000 frames) has been tracked at once without reinitialization. One of three camera views is shown.

of pixels when compared to the 10 sec per frame for the aforementioned video sequence at a resolution of 384×288 pixels. Also, to cope with the extremely fast motions of the professional athletes, we had to record the video at a sampling rate of 60 Hz. The parameterization and especially the biomechanical inter-frame motion limits from Section 3.3.2 correspond to the ones used in our other sequences recorded at 25 Hz, with the following exceptions. Due to the fast relative motion of the pelvis *e.g.* during handstands, we had to increase the inter-frame motion limits for the initial 6 d.o.f. pose of our human model. Furthermore, we have tracked the full 51 d.o.f. of our model including the hands and feet, as they are frequently bent *e.g.* to support the body during a handstand. Screenshots of the resulting video sequences are presented in Figure 4.28 and Figure 4.29. As can be seen, our tracking results are to the largest part accurate and robust, despite the extremely challenging motions of the athletes. Especially

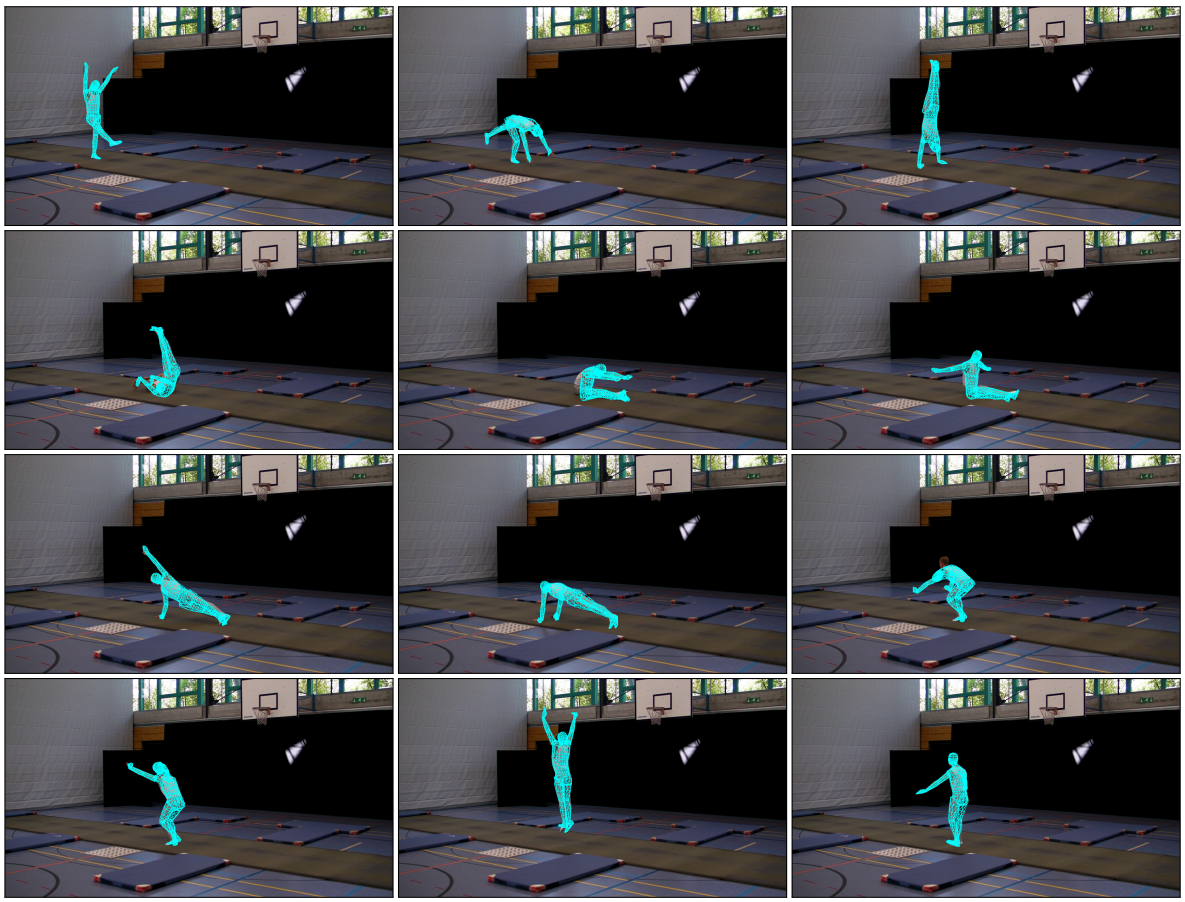

FIGURE 4.28 Screenshots from a video sequence featuring gymnastic exercises of a male subject (750 frames or 12 sec). One of three camera views is shown. The final results show some minor imperfections when lying down due to shadows on the ground that sometimes cause the legs to get switched. However, the flexible motions are accurately estimated due to the realistic biomechanical model and the efficiency of our BIHS sampling strategy.

the strong bending and stretching of the human body during the exercises is perfectly captured by the combination of our realistic human model and our tracking algorithm. Upon closer inspection, we noticed some errors mostly related to the limbs. In the sequence with the female athlete, the right lower leg deviates once, and one of the arms loses track towards the end. In the sequence with the male athlete, the pose estimates get inaccurate and at one point the legs are switched when the athlete is sitting down. When investigating these errors, we noticed that strong shadows close to the ground caused problems when the athlete was lying down, and that the clothing of the athletes was not visually distinctive from some regions in the background. All of these problems thus seem to be related to the foreground segmentation.

The final sequence we want to present in this section is also the first sequence that we recorded outdoors under direct sunlight. It features an athlete exercising in shot-put. The

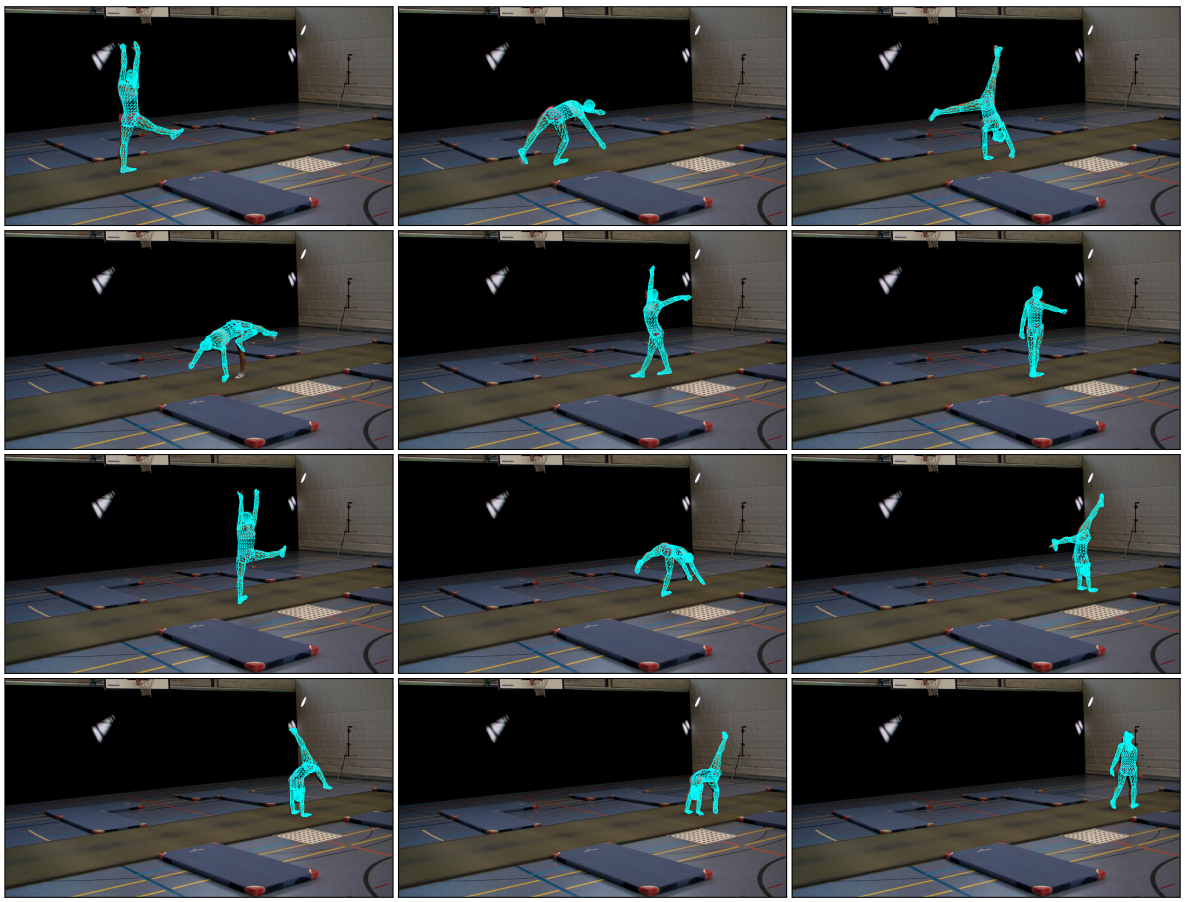

FIGURE 4.29 Screenshots from a video sequence featuring gymnastic exercises of a female subject (640 frames or 10 sec). One of three camera views is shown (differing from the one in Figure 4.28). The extreme bending motions are accurately estimated due to the realistic biomechanical model. Minor errors corresponding to the temporary loss of one arm or one leg are caused by problems with the foreground segmentation.

observed action of throwing the weight is performed in one sudden and fast motion. To be able to capture this motion and to avoid motion blur, we again had to record the sequence at 60 Hz. We used the same setup as in the gymnastic sequences and processed the scene at *Full HD* resolution (1920×1080) in one go. Figure 4.30 shows screenshots from the resulting sequence. The estimated motion is smooth and fits very well to the observed human, with the exception of the left arm that lost track early in the sequence. After investigating this tracking loss, we observed that the left arm was kept close to the body during the first half of the sequence. Due to suboptimal placement of the cameras (two of the cameras were nearly opposite of each other, which resulted in mirrored silhouette shapes), the left arm was not visible in any of the binary foreground masks. Thus, the tracker made uninformed estimates of the arm's motion and was not able to recover the arm's position once it suddenly reappeared during a fast swing

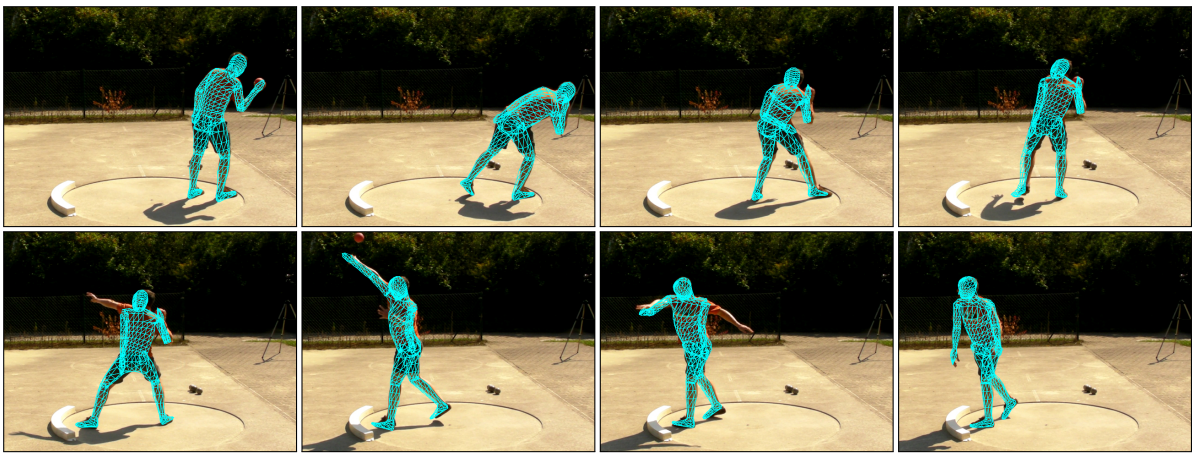

FIGURE 4.30 Screenshots from an outdoor shot-put sequence (580 frames or 10 sec). The motion is extremely fast and had to be recorded at 60 Hz. The estimated motion is realistic and smooth, with the exception of the left arm that lost track early in the sequence. We attribute this to a bad placement of the cameras on our behalf, as two of the three cameras used were placed almost opposite of each other. The resulting pose ambiguities of the left arm when moving close to the body could not be resolved. Images have been cropped.

motion away from the body. This shows that at least three cameras with substantially differing and non-opposite viewing directions are needed to resolve the ambiguities resulting from self-occlusions of the human body. On a side note, the strong shadows on the ground caused by the direct sunlight did not distract the tracker, as the resulting shadow segments were well separated from the human silhouette shape.

As has been shown, most of the problems encountered during our experiments can be attributed to badly extracted silhouette masks from the foreground segmentation. This is often a weak spot when applying human motion capture methods in uncontrolled environments. During our experiments we took special care that subjects wore clearly distinguished clothing and we tried to record with as much ambient light as possible to reduce shadows caused by the tracked subjects. We acknowledge that more sophisticated methods for foreground segmentation will be needed in practice.

Further sequences will be presented in the experimental evaluation section at the end of the next chapter of this thesis (Section 5.3). These sequences feature activities that include manipulations of objects and parts of the environment as well as interactions with other humans. The methodology to cope with such challenging sequences will be presented in Chapter 5.

## 4.7 Summary

In this chapter we have presented our approach on human pose tracking. Tracking is performed using recursive Bayesian estimation, which is the preferred choice for most tracking tasks due to the power and flexibility of the underlying probabilistic framework. It allows us to model the belief about the current state (or pose) as a probability density function (pdf) that accommodates the uncertainties that inevitable come with the task at hand. Given an initial estimate, the pdfs can be computed incrementally for each timestep by first predicting a new *prior* distribution based on the dynamics of the system as defined in the *motion model*, and then computing the updated *posterior* distribution by incorporating the current image observations as defined in the *observation model*.

When designing an algorithm for human pose tracking, one has to choose appropriate motion and observation models. For the motion model, we constrict ourselves to an unconstrained model as we want to be able to track arbitrary motions. We assume that the pose stays constant and only add random diffusion to account for the growing uncertainty induced by the unknown motion between two timesteps. Our observation model is based on a comparison between segmented foreground regions in multiple camera views and the corresponding projections of our human model for all predicted poses.

In addition to the motion and observation model, one has to decide on a Bayesian estimator that makes inference of the *posterior* pdf tractable. We have discussed commonly used estimators such as the *Kalman filter*, the *extended Kalman filter*, or the *particle filter*, and investigated their applicability to the problem of articulated human pose tracking. Among these estimators, particle filters are the most promising in the context of human pose tracking. However, when faced with the high dimensionality of articulated human models, standard particle filtering quickly becomes computationally intractable.

We have then presented two hierarchical variants of particle filtering that have been proposed in the literature, namely *annealed particle filtering* (APF) [46] and *partitioned sampling* (PS) [94]. While APF is a multi-layered search strategy that has been inspired by *simulated annealing*, PS is an approach at hierarchical decomposition of the state space that is well-suited for articulated models. We have extensively evaluated their performance in human pose tracking to assess strengths and weaknesses of both approaches. While both approaches perform well for tracking models with about 20 d.o.f. (evaluated in the context of upper body tracking), they fail to robustly track higher dimensional models such as the model presented in Chapter 3.

Based on the insights from our comparison of APF and PS, we have developed a novel so-

phisticated sampling strategy for particle filters. This *branched iterative hierarchical sampling* (BIHS) as introduced in Section 4.5 combines the multi-layered search strategy from the APF with the hierarchical decomposition of PS to create a more robust and accurate sampling strategy. In addition, it evaluates several partitioning schemes in parallel to improve robustness in the presence of noisy observation data. We have also shown how *adaptive sample sets* can be used to improve computational efficiency by coupling the number of particles and layers in each partition to the uncertainty of the current estimate.

The newly proposed BIHS strategy has many considerable advantages when compared to current state-of-the-art pose tracking algorithms. First, our exploration strategy is much more efficient when compared to APF, and much more reliable when compared to PS. It works by dividing the search space into manageable chunks that are then efficiently sampled by identifying the relevant regions inside these partition and focusing particle evaluations in these regions. Second, we split particle evaluations into parallel pipelines where the size and order of the search space partitions is varied to create a large diversity in paths that the particles take during exploration of the search space. As a result, our algorithm does not have the tendency to get stuck in local maxima as opposed to PS or also deterministic optimization algorithms [12]. Finally, our sampling strategy scales much better to the high dimensionality of the state space than any other unconstrained Bayesian approach that is known to us.

All conclusions in this chapter have been derived from extensive experimental evaluation. For that, we have created simulated sequences from ground-truth motion data to focus our evaluation on the sampling strategies and to eliminate possibly distracting influence from noisy image observations. In addition, we have evaluated all presented methods on the HUMANEVAII data set to compare the mean joint errors that have been computed in relation to the ground-truth data from a marker-based motion capture system on real sequences. The achievable accuracy of the body joint estimates from our pose tracking algorithm is around 2 cm, as indicated in the simulation experiments. Other sequences we have used to evaluate our algorithms feature challenging activities such as floor exercises or shot-putting. Our experiments confirm the high accuracy, reliability and also efficiency of our BIHS strategy when compared to APF or PS. The visual results on most of the tracked sequences are close to perfect, and all sequences could be processed without reinitialization.

# CHAPTER 5

# Appearance and Environment Modeling

## Tracking Complex Manipulation Activities in Everyday Environments

Most approaches to human pose estimation work under the implicit assumptions that (•) the person tracked is the only person in the scene, (•) the person is clearly distinguishable from its surroundings, (•) the person does not interact with objects and the environment, and (•) there is an unobstructed view from all cameras towards the person acting. While these assumptions are convenient and easy to achieve in constrained research environments, they almost never hold for real world applications. Still, little work to date has been targeted on meeting the aforementioned requirements. As an example, consider the quasi-standard HUMANEVA benchmarks [136] that have been adopted by the research community as a means to compare different approaches [156, 26, 161, 55]. They consist of single subjects performing predefined actions like walking, running or boxing in a large and uncluttered environment, without any interactions.

The pose tracking methods as presented in Chapter 4 suffer from similar restrictions, due to the observation model in Section 4.3 being dependant on the extraction of silhouette shapes. The restriction on shapes is critical once the aforementioned implicit assumptions are violated, as the segmentation of complete and unobstructed human silhouettes gets difficult.

In this chapter we will show how to extend the shape-based observation models with color-based appearance models. Up to now we have used color information only for modeling the background, yet color is also a valuable cue for modeling the foreground (*i.e.* the human). While this conclusion is easy to accept, color-based appearance modeling is seldom encountered in human motion tracking. We attribute this to the large computational complexity adding to the already high computational load. We will present an efficient approximation of color distributions to limit the additional complexity and thus enable the use of color-based appearance modeling (Section 5.1). In addition, we will show how appearance modeling can be used to obtain implicit models of the environment. Using these layered environment mod-

els (Section 5.2), we are able to deal with two frequent cases of environmental occlusions. First, we show how to filter dynamic regions of an environment such as doors being opened or objects being manipulated by the human subject. Second, we show how to model objects such as furniture that can both occlude and be occluded by human subjects. This allows us to apply the human pose estimation methods presented in the last chapter to sequences involving humans performing everyday manipulation tasks in typical human living environments and to other challenging scenarios (Section 5.3).

## 5.1 Color-Based Appearance Models

The observation model presented so far in Section 4.3 is purely based on shape cues. When using three or more cameras, silhouette shapes are a sufficiently rich and unambiguous descriptor of human poses ([14] and Section 4.6.1). However, as the observed environments get more complex and dynamic, the segmentation of silhouettes becomes increasingly difficult. Adding additional cues such as color can help to disambiguate segmented foreground objects, as we will show later in this chapter. Furthermore, color information can contribute by enhancing the weight function for pose estimation. When using observation models based on silhouette comparisons, a common problem is that the shapes of some body parts are not indicative on their orientation. This holds especially for heads, that appear round-shaped from all viewing angles. Here, adding color information helps to distinguish between the face and the back of the head.

Color- or texture-based appearance models are seldom used in human pose estimation. This is partially associated to the large computational expenses of common methods, that often require *e.g.* the calculation of Mahalanobis distances to Gaussian color distributions. In contrast to background subtraction methods that can afford to use such techniques as evaluation takes place only once per frame, appearance models used for human tracking have to be evaluated for every particle. Still, some work in this area has been presented. Balan and Black [15] have proposed the use of adaptive image templates that are updated over time. Kehl and van Gool [79] attach color information to the voxels of visual hulls to improve the correspondence search between the model and the visual hull. Gall *et al.* [55] use an analysis-by-synthesis approach to detect point correspondences between the image and a rendered projection of a textured surface model. The computational costs are kept within reasonable bounds, as the evaluation is restricted to only a few salient points. As a downside, subjects need to be wearing prominently textured clothing.

### 5.1.1 Fast Approximate Color Representation Using Bitmasks

In this section we present a fast approximate method to encode color distributions that will enable us to efficiently render and evaluate color-based versions of our human model. Our method supports the encoding of colors $\mathcal{C}$ and of color clusters/distributions $\mathcal{D}$, and it provides an efficient test function $isSimilar(\mathcal{C}, \mathcal{D})$ for deciding whether a color corresponds to a color distribution. In practice, the color $\mathcal{C}$ often corresponds to an observation that is compared to a trained color distribution $\mathcal{D}$ of an appearance model. To be able to do this comparison efficiently, we divide the 3D colorspace (*e.g.* RGB, HSV, HLS, Lab, *etc.*) into voxels of equal size and represent each color channel with a bitmask. When using 32 bit Integers for each color channel, the colorspace is divided into a grid of $32 \times 32 \times 32 = 32768$ different colors. The set bits in each bitmask correspond to grid indices for the color voxels, and the intersections of the set bits in the three channel bitmasks mark the color voxels belonging to the respective color cluster (Figure 5.1a).

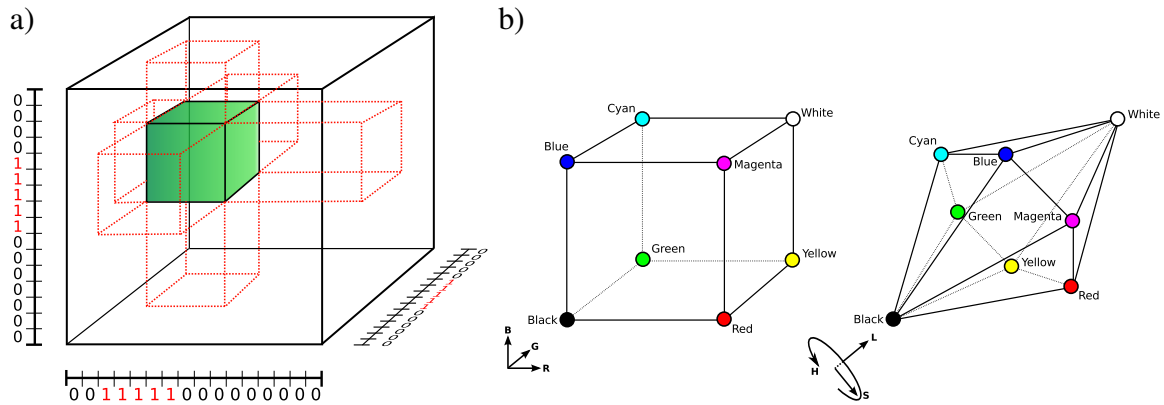


FIGURE 5.1 a) Bit-indexed color representation, where the bits set in three bitmasks encode a cuboid color distributions in 3D colorspace. b) RGB cube and HLS dual cone color representations at comparable spatial alignment. When using bit-indexed color representations, the HLS dual cone representation is able to encode color distributions that are closer to the ones observable in human environments. The reason for this is that its underlying axes (hue, luminance and saturation) are better aligned with the principal directions of variance in typical color distribution.

We distinguish between individual colors $\mathcal{C}_{XYZ}$ and color clusters or distributions $\mathcal{D}_{XYZ}$ (where the subscript $XYZ$ denotes the used colorspace). Single colors $\mathcal{C}_{XYZ}$ are defined by three corresponding voxel indices $\langle i_X, i_Y, i_Z \rangle$ (one per color dimension). Color clusters or distributions $\mathcal{D}$ are defined by three voxel index intervals $\langle [i_X : j_X], [i_Y : j_Y], [i_Z : j_Z] \rangle$, that when expressed in terms of bitmasks (*i.e.* each index in the interval corresponds to a set bit) represent a cuboid voxel cluster in the colorspace $XYZ$ (as in Figure 5.1a). Other types of

possible bitmask encodings (with gaps between set bits) are not allowed.

The core advantage of this bitmask representation is that the function $isSimilar(\mathcal{C}, \mathcal{D})$ for testing whether a color $\mathcal{C}$ corresponds to the color cluster $\mathcal{D}$ is now efficiently implemented using bitwise logical AND operations. Correspondence is established when all three resulting Integers are non-zero, *i.e.* at least one bit is set. Such computations can be efficiently optimized in software using techniques such as loop-unrolling, or in hardware by taking advantage of SIMD (*single instruction, multiple data*) extensions or direct parallelization on GPUs.

A downside to this representation is that it is only possible to encode rectangular clusters of color in the 3D colorspace. This restriction is critical when using the RGB colorspace, as typical color distributions resemble ellipsoids whose main directions of variance approximately correspond to the brightness direction (*i.e.* the diagonal between black and white) [80]. Therefore, we propose to use the HLS colorspace (*hue*, *luminance* and *saturation*), where the brightness direction is directly given via the luminance component (note that we do not use the HSV space as the brightness component there is non-linear). Rectangular color clusters in the HLS space thus corresponds to non-rectangular clusters in the RGB space that approximate the observed color distributions more closely. Figure 5.1b shows the RGB and HLS colorspaces at a comparable spatial alignment, which depicts the diagonal orientation of the HLS space when compared to the RGB space.

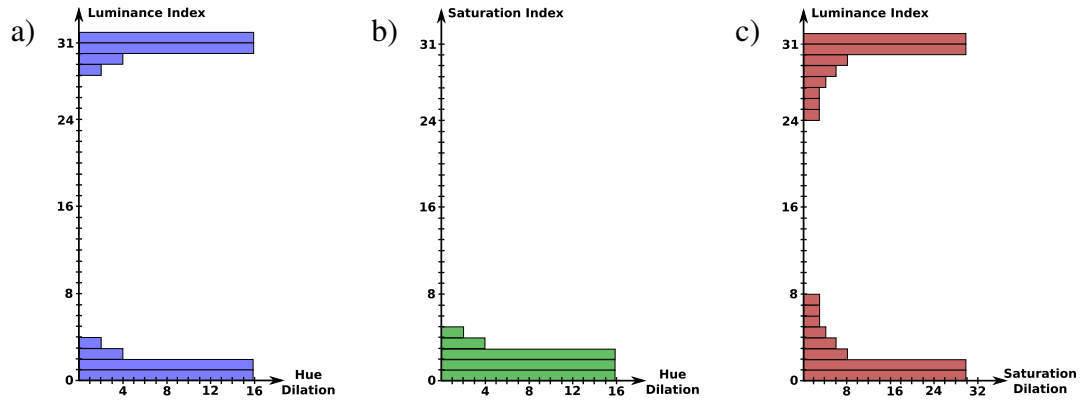


FIGURE 5.2 Minimal dilation values for HLS color bitmasks $\mathcal{C}_{HLS}$ as perceived by humans: a) hue dependant on luminance index $i_L$ b) hue dependant on saturation index $i_S$ c) saturation dependant on luminance index $i_L$. Given the luminance (a and c) or saturation (b) indices of a color $\mathcal{C}_{HLS}$, the dilation values approximately encode how many neighboring color voxels in the hue (a and b) or saturation (c) direction are nearly indistinguishable by human perception. The values have been estimated in a simple GUI based perception experiment.

A restriction of the HLS colorspace is the fact that the hue and saturation channels loose significance at high and low luminance, and that hue also looses significance at low saturation. Therefore, when comparing *e.g.* two very dark colors, the hue and saturation indices should not

be influential in deciding on color correspondence, as they might differ significantly despite humans perceiving these colors as similar. This issue can be fixed by dilating the respective bitmasks of the test color $\mathcal{C}_{HLS}$ such that color channels with low significance will always provide non-zero values when being evaluated using the bitwise AND operation, regardless of the initial bitmask indices. Applying dilation to a bitmask means that all bits inside a fixed size neighborhood of initially set bits get also set, thus effectively increasing the color cluster by adding neighboring colors. Note that dilation is circular for the hue channel, *i.e.* neighborhoods wrap around at the highest and lowest bits of the hue bitmask. Figure 5.2 depicts the neighborhood size of the dilation that should be added to the hue respectively saturation channels of the test color $\mathcal{C}_{HLS}$ depending on its luminance and saturation indices $i_L$ and $i_S$. The corresponding dilated bitmasks for each color $\mathcal{C}_{HLS}$ can be precomputed in a look-up table for fast access. The dilation values in Figure 5.2 have been estimated manually based on whether neighboring color voxels could be perceived as different colors.

| Channel | Value | Bitmask (32bit, most-significant bit left) |
|---|---|---|
| Hue | 0 | 00000000000000000000000000000001 |
| Lum | 128 | 00000000000000010000000000000000 |
| Sat | 255 | 10000000000000000000000000000000 |
| Hue | 0 | 11000000000000000000000000000111 |
| Lum | 128 | 00000000011111111110000000000 |
| Sat | 255 | 11100000000000000000000000000000 |
| Hue | 96 | 11111111111111111111111111111111 |
| Lum | 10 | 00000000000000000000000000000010 |
| Sat | 170 | 11111111111111111111111111111111 |

TABLE 5.1 Example bitmasks corresponding to the following colors (top-down): 1) bright red color 2) bright red color with optional dilation to increase the allowed variation (especially in the luminance/brightness component) 3) very dark red (almost indistinguishable from black); the hue and saturation channels were dilated according to Figure 5.2 to account for their insignificance at low luminance.

Another potentially useful application for dilation is to reduce the expected level of similarity between a color $\mathcal{C}_{HLS}$ and a distribution $\mathcal{D}_{HLS}$. The more dilation is applied to $\mathcal{C}_{HLS}$, the less similar does the color need to be to the trained distribution. By applying this principle to *e.g.* the luminance component, we can allow for larger brightness differences between colors. In Table 5.1 we provide some example bitmask representations of colors that depict how

dilation is applied depending on context.

To account for changes in lighting conditions, it is possible to bit-shift the luminance components of learned color distributions, to approximate the expected appearance under these changed conditions. The amount of shifting can be estimated by a histogram comparison of the current images and the images used when training the color distribution.

### 5.1.2 Human Appearance Models for Pose Estimation

In the last section we have presented a fast approximate color representation based on bit-masks that allows us to compare observed colors $\mathcal{C}$ with learned color distributions $\mathcal{D}$ at a low computational cost. We will now show how this can be used to enhance the shape-based observation model from Section 4.3 by incorporating color cues. The idea is to learn a color distribution for each triangle on the surface mesh of our human model from Chapter 3, *i.e.* to augment the human model with an appearance model. Figure 5.3 shows three examples of human model instances that have been augmented with color information that was extracted directly from the camera images.

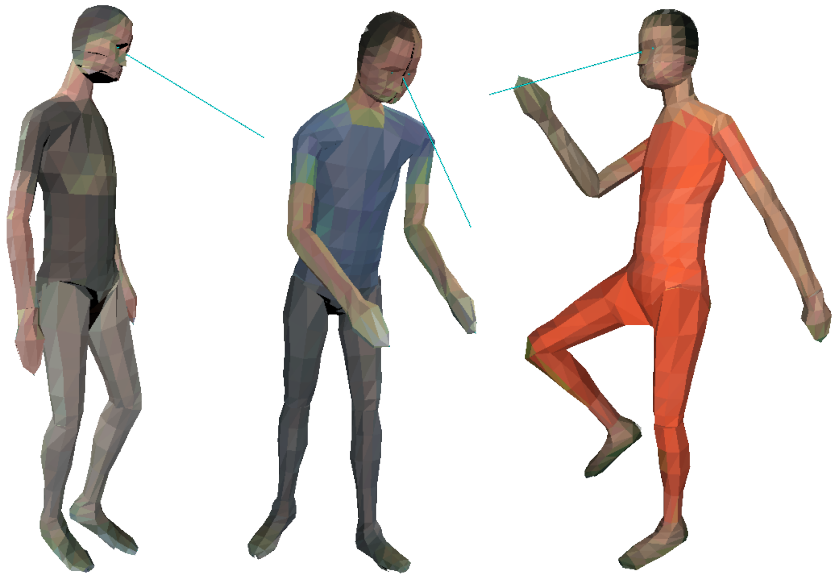

FIGURE 5.3 Renderings of three human model instances in different poses with learned mean colors for each surface triangle.

We learn color distributions for each surface triangle by getting a first estimate from the initial pose before tracking starts. This estimate is then refined, as tracking progresses. The association of image colors to the surface triangles is achieved by rendering the triangles of the model in the known or estimated pose with the *z-buffer* algorithm using a unique global

triangle index as color. As a result, each pixel on the model projection (and thus its color in the current image) is associated with the index of the triangle that it most likely belongs to. Using the z-buffer algorithm for rendering ensures that distant triangles are drawn before closer ones, and thus that pixel colors are only associated to visible surface triangles. To prevent the association of outliers or noisy colors to color distributions, we only add new colors when they are reasonably close to the current cluster mean, and when the triangle normal is approximately facing towards the camera from which the current image was taken. By updating only triangles that are faced towards the camera, chances that the visible pixel corresponds to the human subject (and more precisely to that specific triangle) are higher than when updating triangles that are facing sideways. The HLS color bitmask representation $\mathcal{D}_{HLS}$ that best represents the color samples of an associated triangle is calculated based on the sample mean of all colors in the HLS space and the sample variances in the direction of the hue, luminance and saturation channels. For long-term observations, we recommend to update the means of the luminance channel based on brightness changes that are observed by histogram comparison.

Given a color distribution for each surface triangle of the human model, we can render colored model projections for each camera view and perform a pixelwise color comparison with the current image. Therefore, we estimate how well a rendered color projection of our model matches with the image colors. We do this for every particle state $\mathbf{x}_t^{(n)}$ by calculating the appearance error $e_a^{(n)}$ that corresponds to the ratio between the *number of projected pixels that differ in color from the current image* and the *number of all projected model pixels* (*i.e.* the number of set pixels in the binary projection mask $I_P^{(n)}$):

$$e_a^{(n)} = \frac{\mathrm{COUNT}(I_P^{(n)}) - \mathrm{COUNT}(\,\mathrm{SIMILAR}(C_P^{(n)},\, C_I)\,)}{\mathrm{COUNT}(I_P^{(n)})} \tag{5.1}$$

Here, $C_P^{(n)}$ corresponds to the HLS color bitmask rendering of the human model based on the state $\mathbf{x}_t^{(n)}$ of the $n$-th particle, and $C_I$ corresponds to the HLS color bitmask representation of the current image. The operator $\mathrm{SIMILAR}(C_a, C_b)$ in Equation 5.1 converts two color bitmasks $C_a$ and $C_b$ into a binary region mask that is set (1) for all pixels that are similar in color, and unset (0) for all pixels that differ in color. This corresponds to a pixelwise evaluation of the $isSimilar$ function introduced in Section 5.1.1. Note that all pixels outside of the rendered projection in $C_P^{(n)}$ will evaluate to $0$ when their corresponding bitmasks are also set to $0$. The $\mathrm{COUNT}$ operator counts the number of non-zero pixels as introduced in Equation 4.25.

To incorporate the new error measure $e_a$ from Equation 5.1, we need to update the calcu-

lation of the particle weights $w^{(n)}$ from Equation 4.27 to consider this additional source of information:

$$w_s^{(n)} = 1 - \frac{e_s^{(n)} - \min_n e_s^{(n)}}{\max_n e_s^{(n)} - \min_n e_s^{(n)}} \tag{5.2}$$

$$w_a^{(n)} = 1 - \frac{e_a^{(n)} - \min_n e_a^{(n)}}{\max_n e_a^{(n)} - \min_n e_a^{(n)}} \tag{5.3}$$

$$w^{(n)} = \alpha \cdot w_s^{(n)} + \beta \cdot w_a^{(n)} \tag{5.4}$$

Now, $w^{(n)}$ corresponds to a combined weight based on shape and color cues. The constants $\alpha$ and $\beta$ are used to balance the contributions of the shape weights $w_s^{(n)}$ and the appearance weights $w_a^{(n)}$ to the overall weight $w^{(n)}$. In our experiments we found $\alpha = 0.7$ and $\beta = 0.3$ to be suitable values, as the shape-based weights are usually more reliable than the ones based on appearance. Finally, the weights are adapted as in Equations 4.29 or 4.30 to control the survival rate, and then normalized such that all weights sum up to 1.

By rendering a fully colored model of the human subject that is evaluated for every pixel, our observation model is to large extents independent from the type of clothing that is worn. Therefore, we do not rely on the presence of salient features that require textured clothing to work well. On the other hand, detailed textures on the clothing will not be captured by our method, as all pixels corresponding to a surface triangle are equally colored and represented by a single distribution. However, the level of detail on the surface mesh is high enough to provide good approximations even for textured clothes.

## 5.2 Layered Environment Models

In Section 4.3 we introduced a shape-based observation model that compares binary foreground masks $I_F$ that are extracted using foreground-background segmentation methods with model projection masks $I_P$ that are rendered shape projection of the current pose state. Such observation models are prone to errors once scenes become dynamic and subjects start interacting with the environment. In these cases, the extracted foreground masks $I_F$ can contain dynamic parts of the environment in addition to the human subjects. Furthermore, humans might be contained only partially in the foreground silhouettes due to occlusions from the environment. Therefore, commonly used shape-based observation models such as the one presented in Section 4.3 become unusable. Color-based observation models such as the one

presented in Section 5.1.2 can be used to enhance shape-based observation models, but they do not provide a replacement, as color information is less reliable than shape. They also suffer from occlusions to some extent, especially when humans wear homogeneously colored clothes and the amount of available color cues that can be used to distinguish between different poses becomes critically low. The most promising solution to these kinds of problems is to model the environment, such that occlusion reasoning can be performed.

Cases where humans are interacting with a dynamic environment have to the best of our knowledge not been considered to date in the CV community. The quasi-standard HUMANEVA benchmarks [136] of single actors performing in an uncluttered environment without environmental interactions illustrate the current state of the art in research. Some approaches have used models of the ground plane to reduce the common problem of foot skate during tracking, *e.g*. Vondrak *et al*. [161] use a motion prior enforcing physical constraints from the ground plane contact. Rosenhahn *et al*. [123] incorporate the ground plane as a constraint into their optimization framework. However, full 3D models of the environment have not yet been used in human pose estimation, as real-time acquisition of such models in itself is an unsolved problem.

We propose the use of layered environment models to approach the problem of humans interacting with dynamic environments. These are simple 2D models that are capable to implicitly model occlusions and dynamic regions directly in the image plane. They take advantage of the learned appearance models presented in Section 5.1.2, and require only minimal initialization by the user. This makes them faster and also much easier to use in practice than approaches that would require explicit 3D models of the environment. Still, they are capable to deal with all cases of occlusions and dynamic objects or dynamic parts of the environment.

To deal with cases of dynamic non-human foreground objects and environmental occlusions, we introduce a new binary layer mask $I_B$ into our observation model, that will be used to block regions from processing. This mask is set (1) for all pixels that should be processed, and unset (0) for regions to be blocked. Blocking is then achieved by masking out the respective parts in both the foreground mask $I_F$ and the projection mask $I_P$ before evaluating the shape error $e_s$ from Equation 4.26 and the appearance error $e_a$ from Equation 5.1:

$$I_F = \mathrm{AND}(I_F, I_B) \tag{5.5}$$

$$I_P = \mathrm{AND}(I_P, I_B) \tag{5.6}$$

The filtering of blocked regions is done by a pixelwise logical $\mathrm{AND}$ operation between both the foreground mask $I_F$ and the blocking layer $I_B$ and between the projection mask $I_P$ and

the blocking layer $I_B$. This principle is graphically depicted in Figure 5.4 for the example of blocking an occluding table. As a result, the blocked regions in the image plane are removed from $I_F$ and $I_P$. Thus, silhouette differences between $I_F$ and $I_P$ in these blocked regions do not add to the shape error $e_s$, nor can these regions be used to favor incorrect model projections when containing incorrectly detected foreground shapes (*e.g.* dynamic objects).

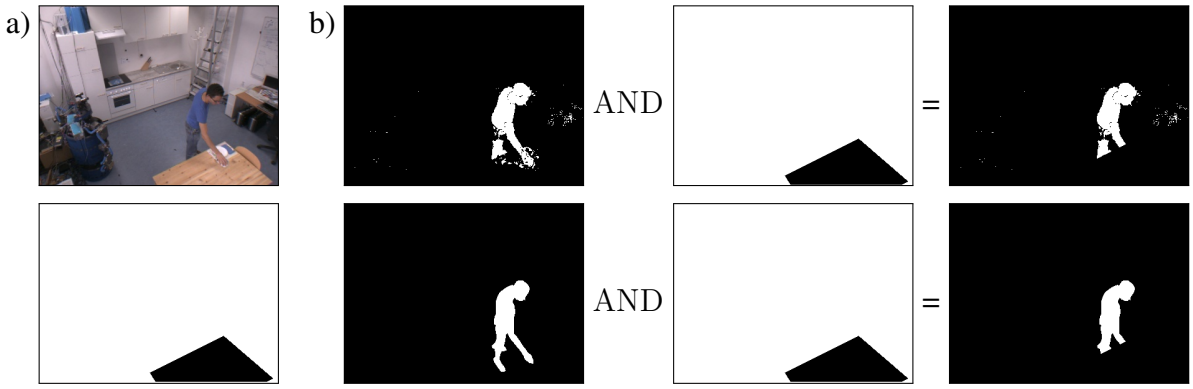


FIGURE 5.4 Principle of blocking layers: a) original camera image (top) and corresponding blocking layer $I_B$ with marked table (bottom) b) the blocking layer is used to remove uncertain regions from both foreground mask $I_F$ (top row) and projection mask $I_P$ (bottom row) by means of a bitwise AND operator. Therefore, the regions marked as blocked in $I_B$ (black) are effectively ignored when calculating the shape error $e_s$ (Equation 4.26).

In the next sections, we will show how these blocking layers can be used to filter dynamic non-human foreground objects and how they can help in modeling occlusions.

### 5.2.1 Filtering Dynamic Non-Human Foreground Objects

Environments in which humans act (and others also) are constantly evolving, *e.g.* objects change their location, doors are being opened and closed, or furniture is moved around. Much of these changes are correlated with human actions, *e.g.* through object manipulations. When trying to observe humans in these environments, such dynamic changes have to be taken into account. Remember that the foreground masks $I_F$ are extracted based on training examples of the scene background (Section 4.3). In the case of long-term changes, such as furniture being moved around once in a while, sophisticated foreground segmentation techniques can be used to adapt to changing background [80]. However, object manipulations such as carrying of objects or opening of doors/cupboards/drawers need to be considered by the pose estimation algorithm, as they will most certainly appear as additional segmented foreground in the foreground mask $I_F$ and mix with the human silhouettes.

Our approach is to distinguish the foreground regions based on their color appearance. To


FIGURE 5.5 Implicit filtering of dynamic obstacles shown on example frame with opened cupboard: a) original images b) unmodified foreground masks $I_F$ c) tracking results without using appearance filtering d) foreground masks $I'_F$ after non-human foreground removal e) tracking results using appearance filtering. Each column shows one camera view. White color in the mask represents set bits.

filter dynamic objects, we introduce another 2D layer with the human appearance mask $I_H$. This binary mask is set (1) whenever a pixel's color resembles a color in the appearance model learned in Section 5.1.2. Such a mask can be calculated efficiently using a binary lookup table for each color voxel index that is calculated once when the appearance model is updated. When using 32 bit integer bitmasks the lookup table will have to index 32768 color voxels. We then remove non-human parts from the foreground mask $I_F$ and add them to the blocking

mask $I_B$ using the following operations:

$$I'_F = \mathrm{AND}(I_F, I_H) \tag{5.7}$$

$$I'_B = \mathrm{AND}(I_B, \mathrm{NOT}(\mathrm{DIFF}(I_F, I'_F))) \tag{5.8}$$

Adding non-human parts to the blocking region is important, as it is unknown whether the dynamic objects are occluding the human. Therefore, the filtered regions will be ignored during processing. This also weakens the influence of erroneously removed parts of the humans, as they will not penalize the final particle weights. The effectivity of these simple operations is shown in Figure 5.5 for the example of opening a cupboard door.

### 5.2.2 Modeling Environmental Occlusions

Another critical case in everyday environments is that the human subjects are often occluded by the environment. When these occlusions are caused by dynamic objects or parts of the environment, we can cope with them by filtering the dynamic non-human foreground as described in Section 5.2.1. However, occlusions are also often caused by static objects such as tables or other kinds of furniture. These objects are permanently modeled as background and do not appear in the foreground masks $I_F$, thus they cannot be filtered by the aforementioned method.

We can use the blocking masks $I_B$ to model these static occluding parts of the environment. We mark regions that are candidates for occlusions (*e.g.* tables) by unsetting these regions in the blocking mask $I_B$. This needs to be done once during camera setup and can be done by a user within seconds by choosing a polygonal region to be considered. Such regions will by default be ignored during evaluation. However, one needs to consider that such objects do not only occlude the humans, but can also be occluded by the humans, as there is no persistent spatial ordering. To prevent valid observations of human body parts to be blocked, *e.g.* when arms are visible above a table, we exclude all human-like foreground regions from blocking:

$$I''_B = \mathrm{OR}(I'_B, I'_F) \tag{5.9}$$

The impact of this occlusion modeling is shown in Figure 5.6 using an example of a human putting down a cup on a table such that his arms are visible above the table but his legs are occluded in some of the camera views. When using the proposed occlusion modeling, the otherwise incorrectly tracked leg positions are correctly recovered.

The amount of occlusion that can be compensated depends on the number of cameras used

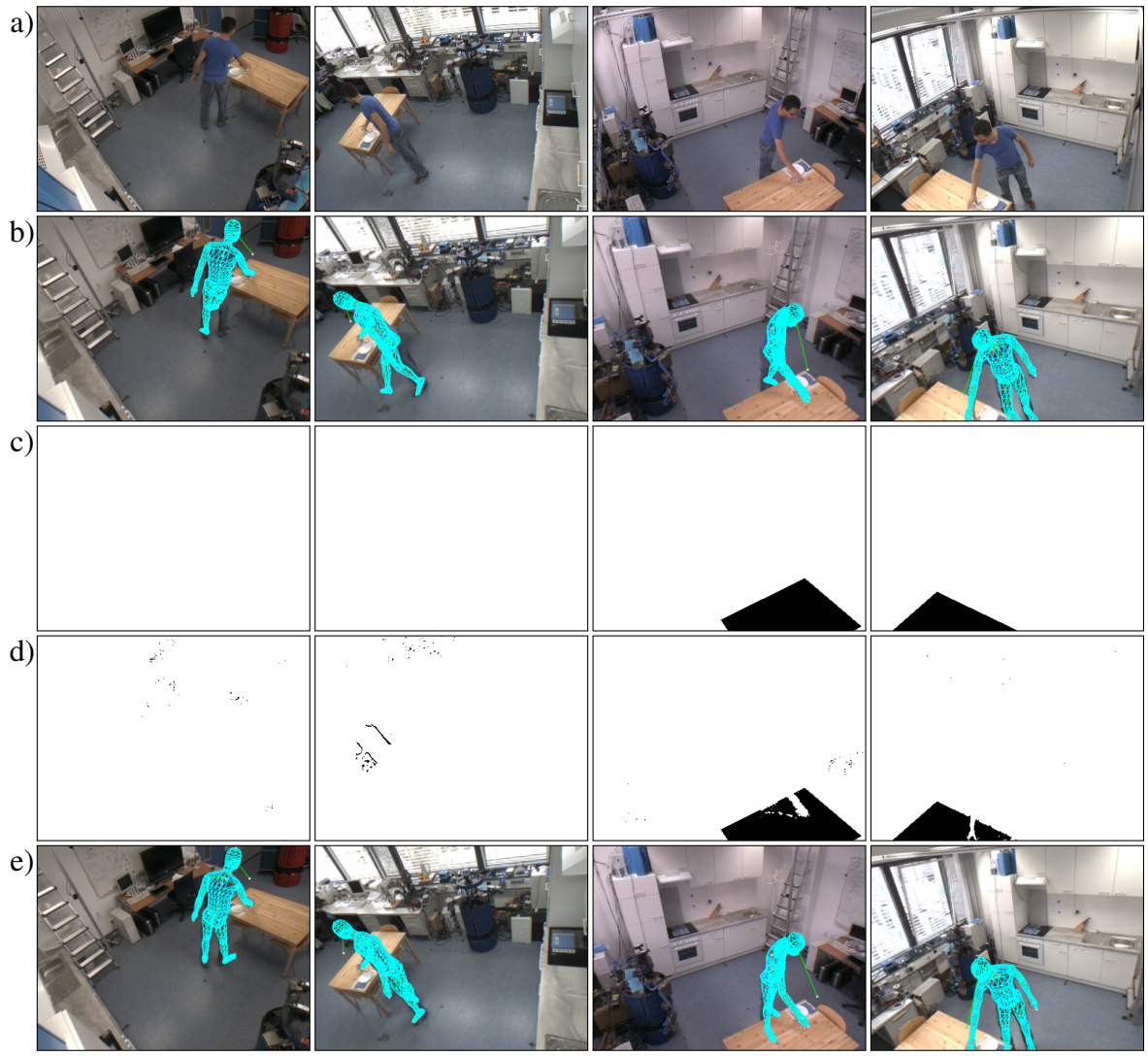


FIGURE 5.6 Implicit occlusion handling shown on example frame with table: a) original images b) tracking results without using blocking layers and appearance filtering c) original user-specified blocking masks $I_B$ d) blocking masks $I''_B$ after exclusion of human-like parts e) tracking results using blocking layers and appearance filtering. Each column shows one camera view. Blocked regions correspond to black areas in the blocking masks.

and on their placement. As a rule of thumb (based on the experiments in Section 4.6.1 on the number of cameras needed for handling ambiguities in human pose estimation), each body part should always be observable from three different camera views. When the occlusion is minor (*e.g.* only the lower legs), two unoccluded camera views are often sufficient. Thus, scenarios where the amount of occlusion is large require more cameras to succeed in tracking the subject. However, the number of cameras needed for successful tracking will still be much lower than when using a comparable occlusion handling with approaches relying on visual

hull extraction.

## 5.3 Experimental Evaluation

To evaluate the methods presented in this chapter, we have recorded several sequences in environments that have a greater level of realism than what the HUMANEVA data set or similar sequences can provide. In particular, this implies that we allow the tracking subjects to be partially occluded by the environment. Furthermore, the subjects may interact with the environment, *e.g*. by manipulating objects or dynamic parts of the environment. Unfortunately, we cannot provide ground-truth data to these sequences, as we do not have access to marker-based motion capture systems and manual labeling of a large amount of image data is beyond our resources. Therefore, we will rely on visual inspection of the tracking results based on wire-frame versions of our human model that are projected on top of the original video sequences. In the multi-camera case, where these projections can be inspected for several viewpoints, this gives a good impression on the accuracy and robustness of the evaluated methods.

We have recorded sequences in two different environments. In Section 5.3.1 we present a scenario where a human subject is getting in and out of a car mock-up. Such a scenario is common in ergonomic applications, *e.g*. when the comfort of passenger compartments is assessed. The main challenges here are the occlusions that are caused by the mock-up. All of the remaining sequences have been recorded in a kitchen environment, where the main challenge is given by the interaction of the subjects with objects and parts of the environment, in addition to occlusions by parts of the environment. We will present sequences of table setting tasks performed in this environment in Section 5.3.2. In Section 5.3.3 we present the TUM KITCHEN data set as a compilation of all recorded sequences of kitchen activities that we have processed. To the best of our knowledge, this is the first publicly available data set where markerless motion capture data has been provided in addition to video sequences. Finally, we show how human appearance models can be used to track multiple targets at once in Section 5.3.4.

In all of the following evaluations, we have used the *branched iterative hierarchical sampling* (BIHS) strategy as presented in Section 4.5 with 10 layers and 800 particles. The parameter settings for the motion model (*i.e*. inter-frame motion limits, annealing schemes, *etc*.) correspond to the settings used in the experiments in Section 4.6.2. In addition, we have learned human appearance models as in Section 5.1 for every recorded subject. We will explicitly denote when we use them to augment the observation model with a color-based weight term as in Section 5.1.2. The implicit environment modeling through layered environment models

as described in Section 5.2 has been used in all of the following experiments.

### 5.3.1 Partial Occlusions in a Car Mock-Up

In our first setup we evaluate the reliability of our motion tracking algorithm in a more cluttered environment with several occluding artifacts. The scenario consists of a person getting into and out of a car mock-up that has been assembled in a large industrial hall. The car mock-up serves as an approximation of a passenger cabin and consists of a driver's seat, armatures, steering wheels, foot pedals, gear shift, and a surrounding skeleton made from steel bars. Due to the missing body panels, mock-ups are well suited for recording the human subjects from different viewpoints. The motion sequence of getting into and out of this mock-up is similar to the motion that would be needed to get into and out of a real car.

Analyzing these kinds of motion sequences is of interest to car manufacturers that want to study the ergonomic characteristics of a new car design. Instead of having to build a detailed and expensive one-to-one copy of a new design to analyze user comfort and safety issues, an assembled mock-up of equivalent size can serve this purpose. At the time of writing however, the automated estimation of human full body motions in such a task is difficult even for marker-based tracking systems, due to the occlusions from bars and other parts of the mock-up. Therefore, the only viable solution so far has been to manually adapt the model to the recorded sequence frame by frame.

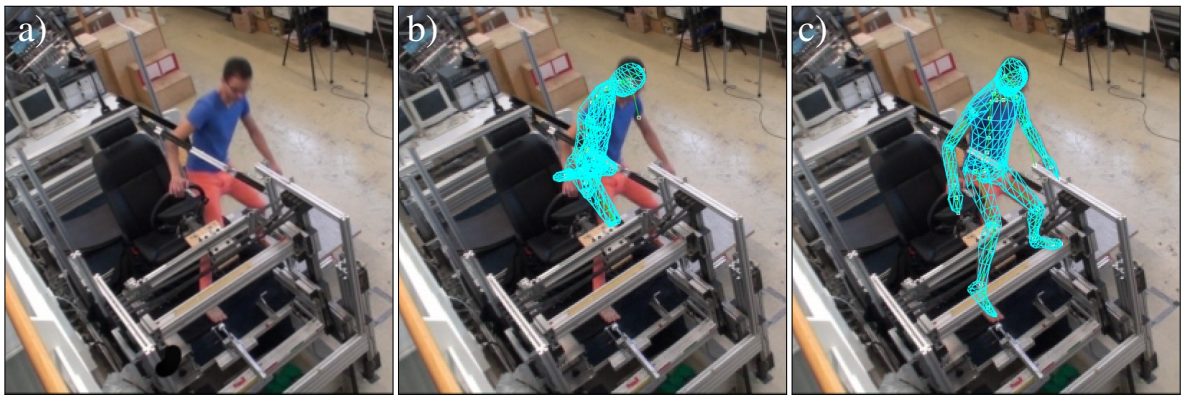


FIGURE 5.7 Example frame from a car mock-up sequence: a) original frame b) tracking result without layered environment model c) tracking result with layered environment model.

We have recorded a lengthy sequence of 2500 frames (or 100 sec) of a subject that gets into and out of the mock-up several times. The sequence also includes the simulation of driving behavior by the subject (steering, shifting the gear, reaching for the glove compartment, *etc.*). Four static cameras have been placed in the environment to observe the subject from different

viewpoints. When sitting in the driver's seat, only one camera has an unoccluded view on the subject. About half of the subject is occluded in two of the three other cameras, and about a fourth of the subject in the last one. During initialization of our setup, we have manually marked the occluded regions in all of the camera views to create the initial blocking masks $I_B$ for the layered environment model as presented in Section 5.2. The video frames have been processed at a resolution of $480 \times 270$.

Due to the large amount of occlusion in the scene, shape-based observation models as the one presented in Section 4.3 without additional environment modeling are bound to fail. Using the layered environment model, we have been able to track the subject through the whole sequence at good accuracy. The difference that our implicit environment modeling makes on the tracking results is shown for an example frame in Figure 5.7. In addition, we show several representative screenshots from the resulting sequence in Figure 5.8. The full video is available from `http://memoman.cs.tum.edu`.

### 5.3.2 Everyday Manipulation Tasks in a Kitchen Environment

The second setup in which we evaluated our methods is the ASSISTIVE KITCHEN [23] at the TUM. This kitchen environment features a kitchenette with typical kitchen appliances (fridge, oven, sink, cupboards and drawers) and a dining table. The kitchen is observed by four ceiling mounted cameras. In addition, a smart sensor network is embedded into the environment, consisting of additional sensors such as laser scanners, RFID readers or magnetic (reed) sensors. However, the additional sensors are not used in our experiments.

We recorded a sequence of a person that was told to set the table. This involved several pick and place actions of objects that were located at different positions in the environment. A place-mat and a napkin were located at the counter-top, while plates and cups were located in a cupboard, and silverware was located in a drawer. To create a longer, more robot-like sequence, the person was told to carry the objects one by one, *e.g.* the spoon, fork and knife have been picked from the drawer and placed on the table separately. The resulting minute-long video sequence is composed of about 1500 frames, which have been processed at a resolution of $384 \times 288$.

The specific challenge in this sequence is to track a person that interacts with the environment by opening cupboards or drawers, that carries objects around to place them elsewhere, and that is partially occluded by the table in two of the four camera views. We have tackled this problem by learning a human specific appearance model as described in Section 5.1. This model was then used to filter dynamic parts in the environment such as revolving doors or objects being carried around (Section 5.2.1), and to correctly model the occlusions from

the dining table without ignoring the visible human activity above the table (Section 5.2.2). For the latter, we have initially created the blocking masks $I_B$ by marking a polygonal region corresponding to the projection of the table in the two affected camera views. The different tracking results when using our layered environment models as compared to no environment modeling are shown in Figure 5.9.

The full sequence was processed in a single tracking run without interruption and yielded visually accurate tracking results. Figure 5.10 shows several screenshots as seen from all four camera views. The position of the torso and all limbs have been correctly tracked throughout the sequence. If at all, smaller inaccuracies can be observed around the feet, where imprecise segmentation due to shadows and the ground leads to foot skate. Also, the orientation of the head is unstable at times. The precision of the resulting pose estimates is illustrated in Figure 5.11, where we show a single frame of the sequence and the corresponding inner and outer model projections. In addition, we show the human model including the learned appearance model from a virtual viewpoint.

In a second experiment, we have evaluated how a combination of the purely shape-based observation model from Section 4.3 with the color-based observation model from Section 5.1.2 affects the tracking behavior. The differences in tracking accuracy were marginal, however we did notice less shaking in the orientation of the head. This makes sense, as the binary silhouette of the head looks almost similar from all sides, while additional color information provides valuable cues on the viewpoint due to the difference in appearance between the face and the back of the head. However, we also noticed in several other sequences that when the tracked subject is wearing uniformly colored clothing, the additional color-based weight term in Equation 5.4 often confuses the tracker and leads to incorrect tracking results. We believe this might be due to inaccurate global maxima of the combined weight function that result from difficult lighting situations or inaccurately learned color distributions. As an example, the tracking algorithm could favor a wrong pose for an arm that associates it with the torso when the observed colors at the true arm location differ strongly from the learned color distributions due to a strong shadow. Our recommendation is thus to use the color-based weight term from Equation 5.4 only when evaluating the model projections corresponding to the head of the human subjects, and to stick to the shape-based observation model from Section 4.3 otherwise.

### 5.3.3 The TUM KITCHEN Data Set

In the last section we have evaluated our method on a table setting task in a realistic kitchen environment. In total, we have recorded more than 40 sequences with 6 different subjects in

this environment. All sequences range from 1 to 5 minutes, and consist mainly of table setting tasks, but also of repetitive pick and place actions.

Out of these sequences, we have processed 21 sequences from 4 different subjects with our markerless motion capture system, out of which 4 are shown in Figure 5.12. The tracking results on these sequences are accurate and reliable despite frequent interactions with the environment, object manipulations, and partial occlusions. Some of the sequences required small amounts of post-processing due to partial tracking failures. The most common failure appeared when subjects were reaching into the rightmost cupboard, and the lifting of the right arm was missed by the tracker. The cause for this is that the lifting motion is not visible in any of the cameras due to a blind spot caused by bad placement of the cameras. Other infrequently observed failures are that one of the arms temporarily sticks to the body or that legs get crossed for a couple of frames only.

To foster scientific exchange, we decided to provide the original video sequences including the retrieved (and whenever necessary post-processed) motion capture data to the research community. The resulting TUM KITCHEN data set [150] is publicly available for download from `http://kitchendata.cs.tum.edu`. In addition to the original calibrated video sequences and our motion capture data in the common `bvh` file format, we have added synchronized sensor readings from the sensor network in the ASSISTIVE KITCHEN. These consist of RFID readings from three fixed readers embedded in the environment (table, counter-top and cupboard) to detect the location of tagged objects such as plates or cups, and of magnetic (reed) sensor readings to detect whether doors or drawers in the environment are opened or closed. In combination with the additionally provided fine-grained semantic action labels (for the torso and for both hands separately), this data set is equally suited for the evaluation of algorithms for human motion capture as well as for motion segmentation or activity recognition. Positive feedback from other researchers and the download statistics for the data set ($\sim$80 GB in the first three months) show that the TUM KITCHEN data set is well received by the research community.

We believe that the data set will aid researchers by providing a comprehensive collection of sensory input data that can be used to develop and verify related algorithms, and that can serve as a benchmark for comparative studies. One of the key advantages of our data set is the high level of realism in both the modeling of the scene and the type of activities that has been recorded. Most data sets to date focus on artificial actions such as *kick*, *punch* or *jumping jack* [128, 25, 42, 43], whereas the TUM KITCHEN data set features natural and subtle motions with smooth transitions between actions. Furthermore, most data sets provide either video sequences [128, 25] or motion capture data [42, 43, 10], but few combine these modalities.

The CMU KITCHEN data set [132] contains multi-modal observations of several cooking tasks, including calibrated cameras and motion capture data from a commercial marker-based motion capture system. However, the actors are heavily equipped with markers and technical devices, which makes it difficult to evaluate markerless motion tracking algorithms on the video data. The TUM KITCHEN data set provides more realistic video sequences due to our unintrusive markerless motion tracking system.

To the best of our knowledge, this is the first time that large amounts of challenging data have been successfully processed by a markerless motion capture system. One exception might be the commercial ORGANIC MOTION STAGE tracking system from INITION [142], however this state-of-the-art system relies on the reconstruction of *visual hulls* that requires much more cameras ($\sim 16$) and an empty environment without clutter. Our method does not suffer from these limitations, due to our sophisticated hierarchical sampling strategy (Section 4.5) in combination with the appearance and environment modeling presented in this chapter. To be able to achieve comparable results for the presented sequences with an observation model based on *visual hull* reconstruction, the number of cameras necessary would likely be intractable.

### 5.3.4 Tracking Multiple Targets in a Kitchen Environment

We have presented layered environment models as a means to distinguish between humans and dynamic objects or dynamic parts of the environment. The same principle can be used to distinguish between multiple humans acting in the same scene. Usually, the combined segmentation of several human silhouettes results in a merged shape that will confuse shape-based trackers. We can use the appearance models to soften this effect by considering the colors corresponding to each human. Of course, this requires that the subjects wear visually distinctive clothes. The respective other subject is then treated as a dynamic object and filtered from the segmentation. However, the filtering is not perfect due to unavoidable overlap in colors between the humans, *e.g.* for the skin colors.

We have recorded a sequence in our kitchen environment where two humans jointly clear the table and load a dishwasher. During this joint activity, several objects are being handed over between the subjects. One of the subject opens the dishwasher to to fill it with the objects it receives, and afterwards closes it again. Furthermore, some objects are placed back into the cupboards.

The sequence has a total length of $1300$ frames or $52$ sec and has been processed at a resolution of $384\times288$ pixels. We have initialized and tracked each subject individually, resulting in two separate runs of our tracking algorithm. For the final video sequence shown in Figure 5.13

we have overlaid the estimated poses from each run with different colors. As the screenshots indicate, the sequence was tracked at a good accuracy with very little imperfections. Despite the motion of the subjects that occlude each other several times in some of the camera views, the tracking quality is comparable to that of single subject tracking. The largest discrepancies between the real and the estimated motion is given when one of the subjects bends down and fills the dishwasher. Here, the estimates for both arms tend to be inaccurate, partially due to the fact that the bend over torso of the subjects blocks the line of sight towards its arms. An additional camera positioned significantly lower than the ceiling mounted cameras could possibly improve the results in this respect.

The whole video can be viewed online at http://memoman.cs.tum.edu.

## 5.4 Summary

Markerless human motion capture methods to date have been restricted to (preferably uncluttered) environments where the cameras can be placed such that an unobstructed view towards the subject is guaranteed. The human silhouette must be easily segmentable from the background as a whole, which excludes all kinds of manipulation activities and scenes that feature dynamic objects. This is best illustrated by the fact that the quasi-standard HUMANEVA benchmarks for markerless human motion capturing adhere to these restrictions, and that the only competitive commercially available markerless motion capture system to date (ORGANIC MOTION STAGE [142]) requires a set-up in a completely empty room with uniformly colored walls.

In this chapter we have shown how to extend markerless motion capturing to cluttered environments with occluded camera views and to activities that involve the manipulation of objects or interactions with other humans or non-static parts of the environment. The main idea is to learn an appearance model for the tracking subject to augment shape information with additional color information. We do this by associating each surface triangle on our human model with a learned color distribution.

Comparing color distributions can be computationally demanding, especially when it needs to be done pixel by pixel for every particle evaluation. We have presented an approximate color model that uses bit-indices to encode color distributions in HLS colorspace. Comparing two colors in this representation can be efficiently implemented with logical bit operations, which is also well-suited for further optimization on a GPU. We can use the color information to augment the shape-based observation model presented in the last chapter with additional appearance cues. The human model can then be rendered in color and compared to the current

image observations. This can help to stabilize the tracking of body parts such as the head, whose silhouette shapes are not easily distinguishable when viewed from different viewpoints.

We have then shown how the human appearance models can be used for implicit environment modeling. This enables us to filter dynamic parts such as objects being carried around or doors being opened from the segmented foreground and to ignore them in our observation model. Furthermore, we are able to model parts of the environment that can both occlude a human and be occluded by a human (*e.g.* chairs and tables). All this can be done efficiently based on 2D layered models without the need for explicit 3D models of the environment.

The presented appearance and environment models enable us to apply our motion tracking algorithm to challenging sequences. In our experiments, we have successfully used our method to track human subjects getting into and out of a car-mockup, and during table-setting tasks in a kitchen environment. Both scenes contain partially occluded camera views, and the kitchen sequences consist of activities where objects and parts of the environment are being manipulated. Our motion tracking is very stable and easy to apply to new scenes due to the unintrusive setup. We were able to process a total of $21$ sequences from $4$ different subjects in the kitchen environment and to publish the results in the TUM KITCHEN data set. To the best of our knowledge, this is the first time that a markerless motion capture system has been used to create a motion capture data set. We can therefore provide motion capture data along with the corresponding video sequences that appear completely natural and without distracting evidence of a motion capture setup. Finally, we have shown that our system is also capable to track joint activities of two interacting subjects in such a challenging environment.

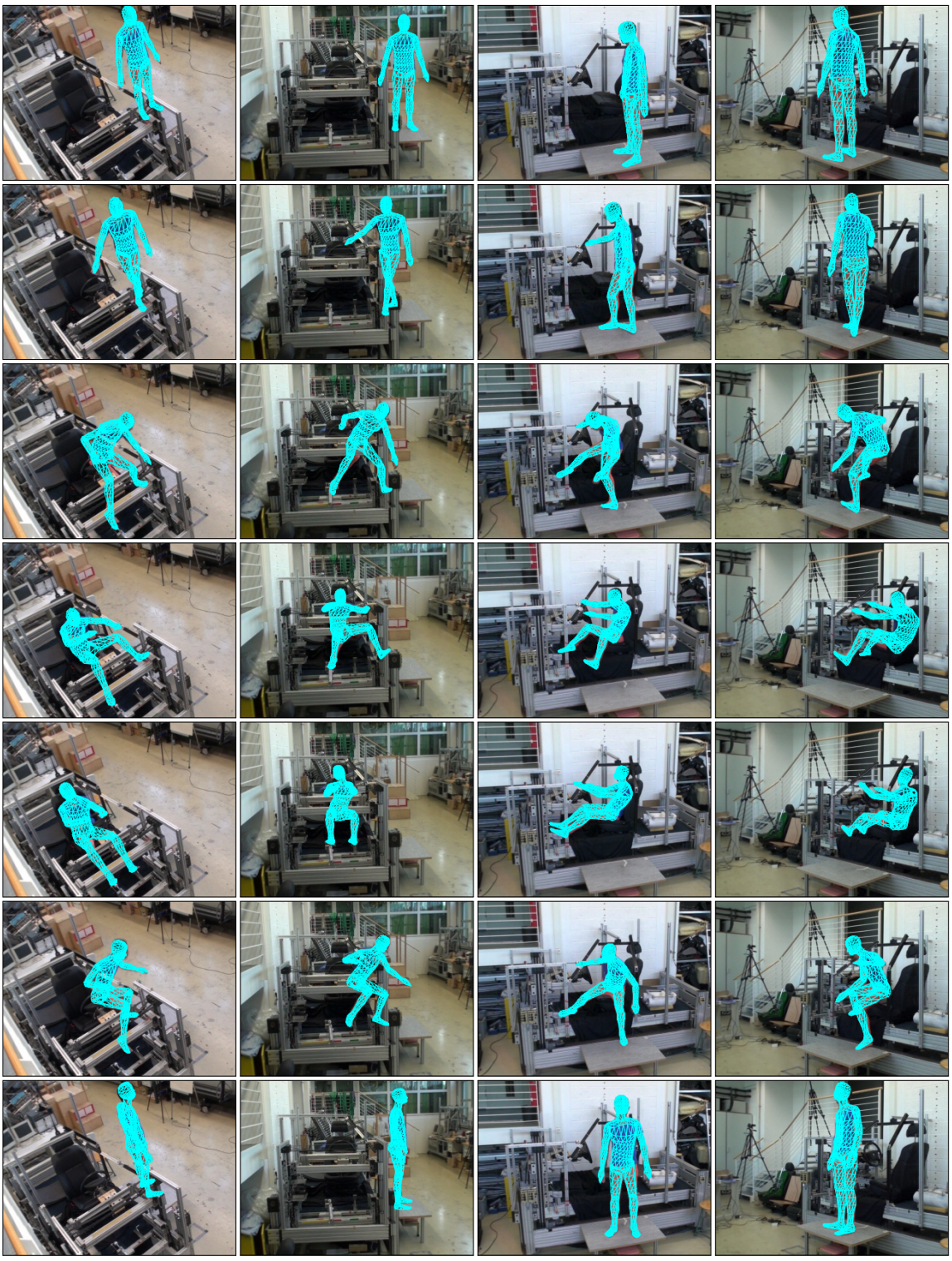

FIGURE 5.8 Screenshots from a car mock-up sequence (2500 frames or 100 sec). Each column corresponds to one of the four camera views. Notice the occlusions from the environment, that are strongest in the second and fourth camera view.

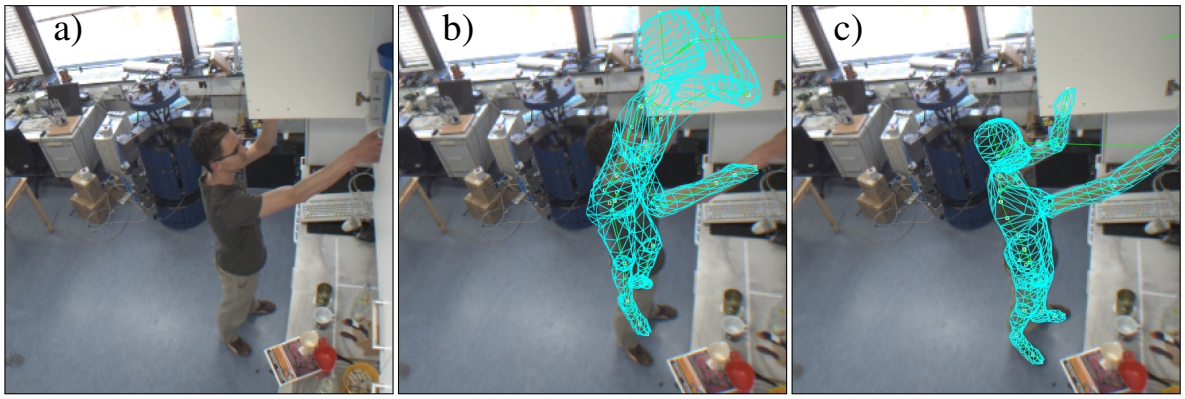


FIGURE 5.9 Example frame from a kitchen sequence: a) original frame b) tracking result without layered environment model c) tracking result with layered environment model.

FIGURE 5.10 Screenshots from a kitchen sequence (1500 frames or 60 sec). Each column corresponds to one of the four camera views. Notice the interactions with objects, drawers and cupboards. The full video is available from http://memoman.cs.tum.edu.

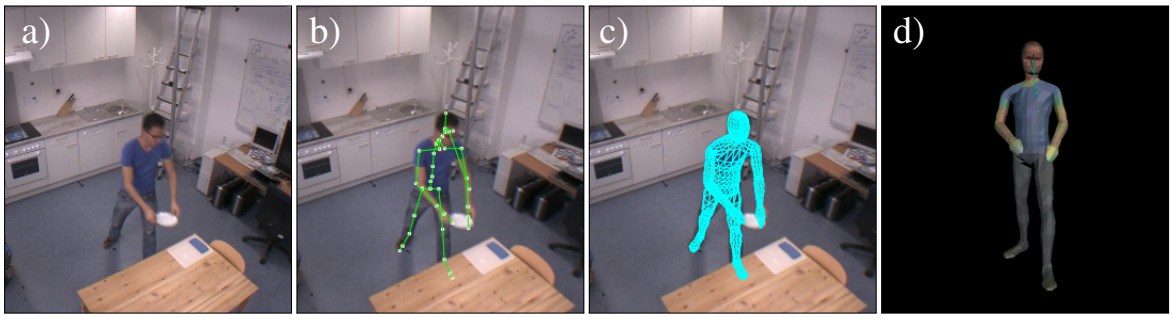


FIGURE 5.11 Example frame from the kitchen sequence while placing a plate: a) original image b) inner model c) outer model d) virtual 3D view with appearance model.

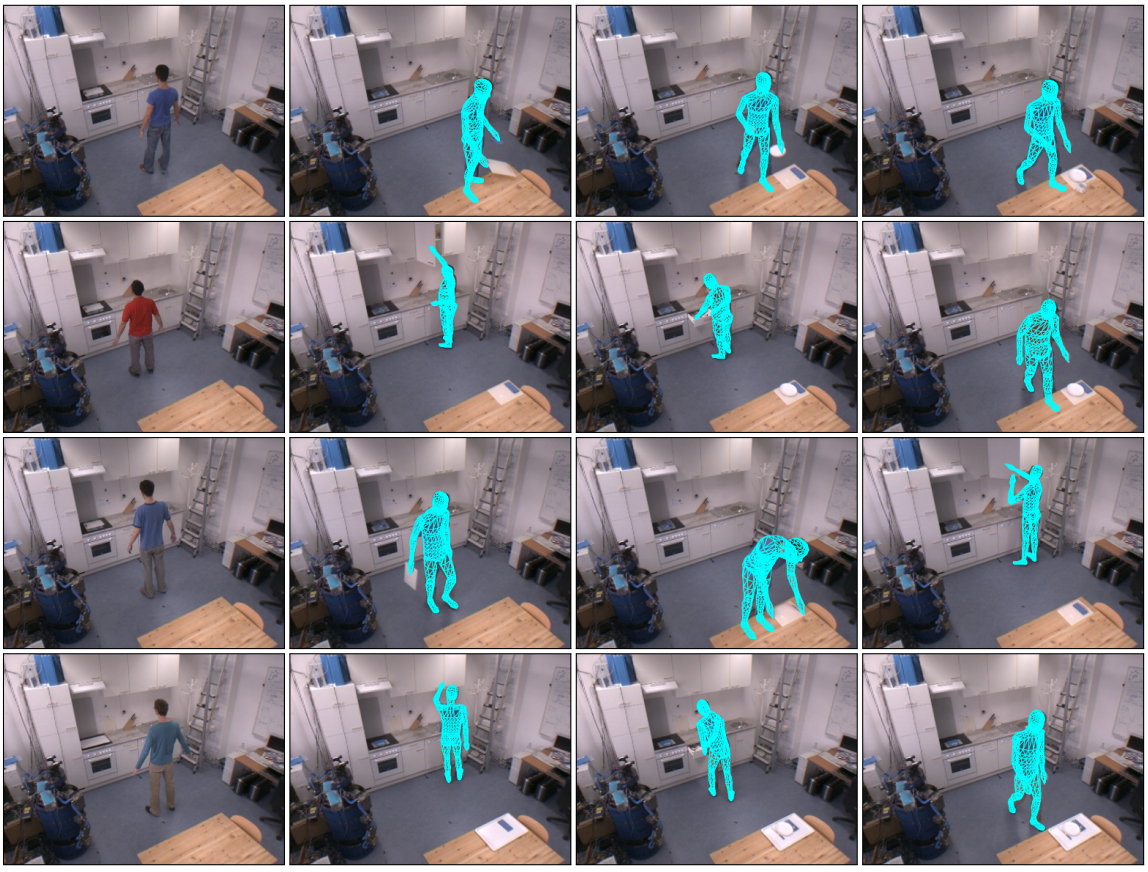

FIGURE 5.12 Screenshots from 4 of the 21 sequences in the TUM KITCHEN data set (one of four camera views is shown). Each row features screenshots from another subject. The first column is shown without model to depict the body shape and clothing of the subjects.

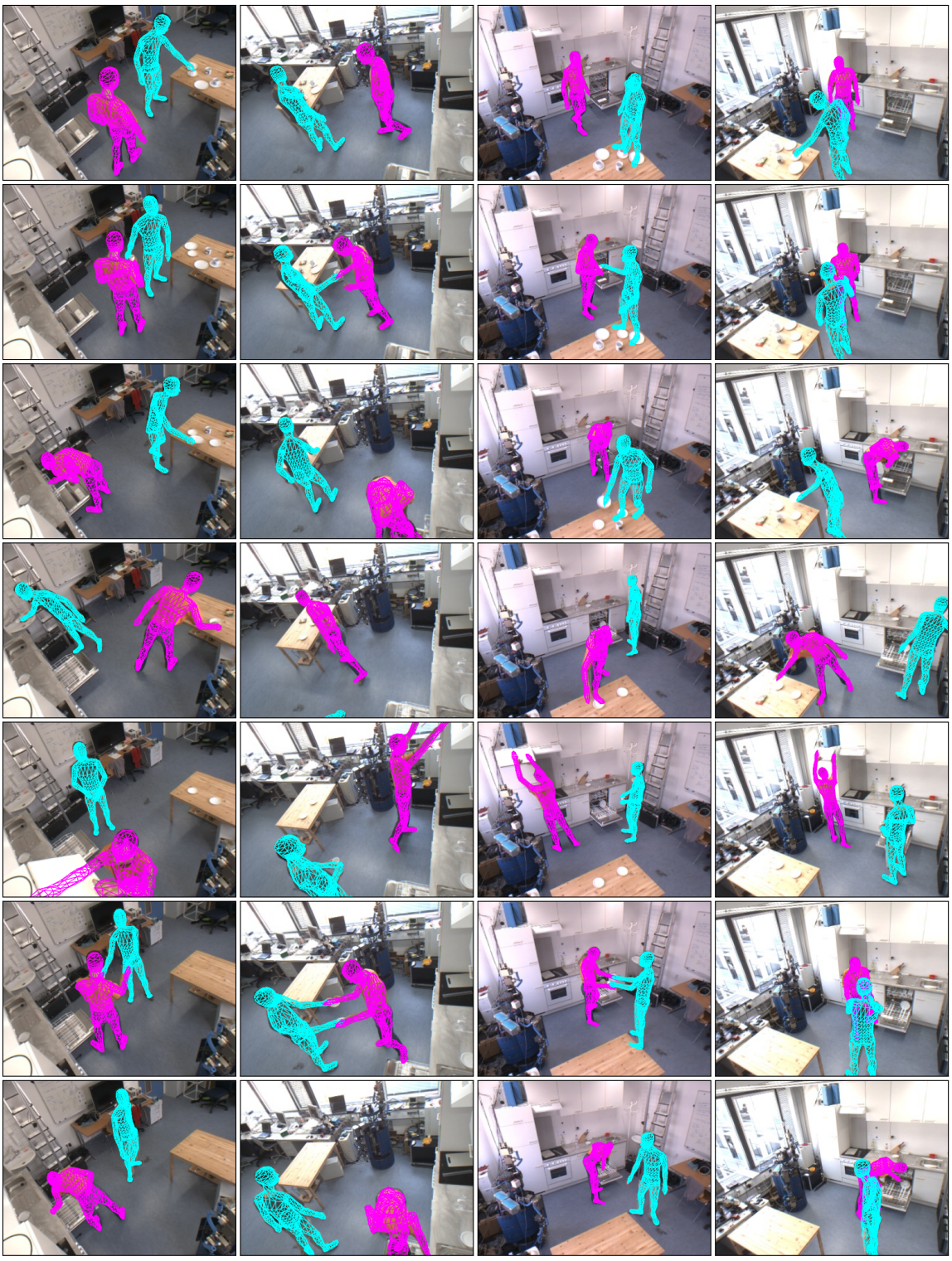

FIGURE 5.13 Screenshots from a kitchen sequence featuring the joint action of two subjects (1300 frames or 50 sec). Each column corresponds to one of the four camera views. The two subjects perform the joint action of clearing the table and loading the dishwasher.

# CHAPTER 6

# Learning Models of Human Motion

## Environment-Specific Models for Prediction and Activity Recognition

So far we have presented a reliable system for tracking of human motions that is able to retrieve fullbody motion data for arbitrary types of motion in natural environments. The capability to observe any kind of motion is a direct consequence of our choice of a general motion model that makes no assumptions about the direction or velocity of body part motions, other than providing biomechanical upper bounds on inter- and intra-frame motion (Section 4.2). The downside of using such a general motion model is the increased complexity of the estimation task due to the vast high-dimensional state space, which requires the use of sophisticated tracking algorithms (Sections 4.4 and 4.5) that are computationally demanding.

Learning-based approaches can provide alternative solutions to human pose estimation that are often more efficient, with the limitation of being restricted to the trained set of motions. Two prevalent research directions can be distinguished. The discriminative approach is to learn a direct mapping from observed image cues (such as silhouettes, *scale-invariant feature transform* (SIFT) [91] or *histogram of oriented gradients* (HOG) [41] descriptors) to body poses [63, 3, 156, 26, 28], which can provide accurate estimates, especially when using monocular vision. These methods are constricted to the types of features they have been trained with and training has to be redone when transferring to environments with different camera perspectives. The second strategy is to learn low-dimensional manifolds of the human pose space to reduce the dimensionality of the problem [158, 157, 163]. Such low-dimensional embeddings are also useful for exposing the underlying structure in the data, which is of potential use in activity recognition.

While showing good performance with respect to known motions, the aforementioned methods have severe limitations when it comes to generalization, as it is almost impossible to extrapolate motions outside of the training set. The problem is the high dimensionality of the human pose space, which makes it infeasible to learn all possible motions beforehand. Al-

though recent progress allows to extend the size of training sets to $\sim 10^5$ exemplars [156, 26], the problem of generating these exemplars remains. Often expensive marker-based motion capture systems are used to acquire the pose representations by observing an actor performing the requested motion. As an alternative, professional computer animation software such as *Poser* [116] or *Maya* [95] is used to create realistic motion sequences. Both approaches are tedious and require substantial manual processing and expertise.

In this chapter we present an integrated approach that combines our fully unconstrained tracking system presented in the last chapters with an incremental learning algorithm that is able to provide good predictions for already encountered motion snippets. The driving idea behind our approach is that a motion capture system should learn to adapt to a specific environment and to the specific activities and motions predominant in it (see Figure 6.1). With time it will be able to predict common activities quickly and to render the full stochastic search unnecessary. The system is fully unsupervised but can additionally be provided with action labels. In such a case it not only provides a generative prediction but also a recognition component for the observed motions.

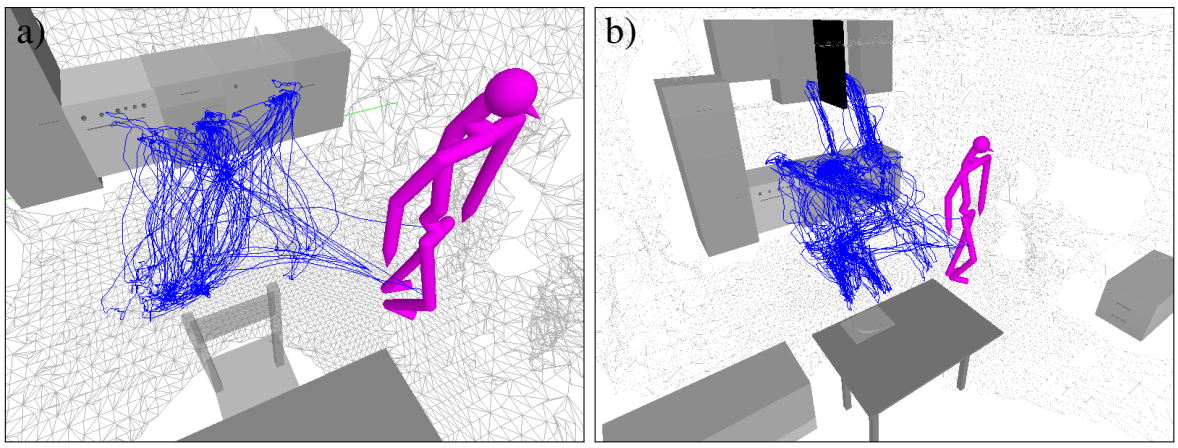


FIGURE 6.1 Visualization of environment specific motion patterns. This visualization from our knowledge base shows body part trajectories in our kitchen environment for 2 subjects and 7 sequences taken from the TUM KITCHEN data set (Section 5.3.3): a) for the pelvis b) for the right hand. Images courtesy of Moritz Tenorth, TUM.

In Section 6.1.1 we will introduce a pose manifold that is represented as a weighted directed graph, with each vertex corresponding to an observed pose including its most recent temporal history. Path lengths between these vertices correspond to geodesic distances and span a manifold inside the high-dimensional pose space. The graph and its creation resembles techniques used in ST-Isomap [74], which has already been shown to be well-suited for representing human motion capture data. We show in Section 6.1.2 how this representation

is used to incrementally learn environment- and task-specific motion models, and how the learned models are used to improve the predictive component of our human motion tracker. In Section 6.1.3 we discuss how to use the same models for online activity recognition. We evaluate the presented methods and concepts in Section 6.2.

## 6.1 Self-Training of Environment-Specific Pose Manifolds

The parameterization of human postures depends on the intended area of application and on the models used. The dimensionality of these models typically ranges around 30 d.o.f. [46, 58, 6] for tracking tasks, but can easily exceed 60 d.o.f. [1] when using detailed biomechanical models for analysis or simulation tasks. The model used in our thesis consists of up to 51 d.o.f. Using such models for unconstrained pose tracking results in a complex estimation problem in a high-dimensional state space. However, the space of valid human poses is a lower-dimensional manifold in this high-dimensional state space, as not all combinations of parameters result in meaningful or physiologically realistic postures.

By learning this manifold, one could reduce the search space and additionally improve the prediction of our motion model in Section 4.2. Unfortunately, the space of valid human poses is still sufficiently large to make it infeasible to learn a manifold comprising all of them. The common approach is therefore to learn a manifold based on training data of a subset of relevant motions. These are usually acquired with marker-based motion capture systems or by computer animation, which is both costly and time-consuming. In the end it often turns out that the training motion is irrelevant for the current area of application, and the pose manifold has to be retrained with a more specific set of motions.

Our approach is to start with an uninformed tracker that uses a general motion model and to learn environment-specific models of motion through observation of ongoing activities in the area of operation. Therefore, the learned pose manifold will correspond to relevant motions for this particular environment. In our kitchen example this will include motions of picking up and placing objects between the kitchenette and the table, and of opening and closing cupboards and drawers in the environment (Figure 6.1). Sports trackers on the other hand will learn the motion patterns corresponding to the observed exercises.

Several methods have been proposed for learning manifolds of human motion. Urtasun and Fua [158] project training motions from the high-dimensional state space to a low-dimensional manifold using *principal component analysis* PCA. Wang *et al*. [163] and Urtasun *et al*. [157] use *Gaussian process dynamical models* to create a low-dimensional manifold with associated dynamics. Jenkins and Matarić [74] propose a spatio-temporal extension to Isomap nonlin-

ear dimensionality reduction [149] to learn such a manifold. All of these methods have in common that they create a low-dimensional embedding of the pose manifold defined by the training data. This embedding is designed to be globally consistent, which helps to visualize the manifold and to uncover the underlying structure in the data. However, these global models of human motion come at a price. On the one hand, they are computationally demanding ($\mathcal{O}(n^3)$) for GPDM and Isomap) which severely limits the size of the training sets, and on the other hand, they tend to produce bad results when the training set consists of a large variety of structurally different motions. The latter occurs when the intrinsic dimensionality of the training data exceeds the dimensionality of the embedding, which makes it difficult to enforce global constraints in the embedding.

As we are interested in improving the prediction of our motion model, we omit the pitfalls of models that try to provide global embeddings of the training data. Instead, we create a graph-based pose manifold that is locally consistent with the training data, which is sufficient for our task. The strategy for improving the efficiency of our tracker is then to reduce the diffusion variance of our estimation algorithm which will allow us to use fewer particles for successful tracking. We can do so because of the informed prediction that will result in a reduced search space due to better initial estimates during tracking. This strategy differs from methods that track directly in the low-dimensional manifold space to take advantage of the reduced dimensionality of the problem. We believe that our strategy is more flexible with respect to extrapolation of poses from the training set and that it avoids the practical problems associated with creating good embeddings of complex motion data.

We will now describe the relevant concepts of our environment-specific motion models in more detail, starting with the graph-based representation used for learned pose manifolds, then discussing how to learn these models incrementally and how to use them to improve our prediction, and finally showing how they can be used for activity recognition.

### 6.1.1 Spatio-Temporal Neighborhood Graphs

Our aim is to create generative models of human motion that are able to predict likely follow-up poses to the current pose estimate during recursive estimation. As mentioned previously, many approaches have tried to improve prediction by learning motion models in terms of low-dimensional embeddings of training poses [158, 157]. Such models are global in that they process the full training set in one step to create a consistent embedding for all data points. However, these global constraints can lead to loss of relevant details in the data when compared to models that aim at local consistency of the data only [44]. While global models do have many important applications, *e.g.* when trying to reveal structural dependencies in

the data points, global consistency is irrelevant in models that are mainly used for near-term prediction.

We propose a graph-based representation of human motion that is derived from ST-Isomap [74], a spatio-temporal extension of Isomap [149]. Isomap (and also ST-Isomap) is an approach for nonlinear dimensionality reduction that tries to estimate the intrinsic geometry of a set of data points in three steps. First, it creates an undirected weighted graph that connects all data points within a local neighborhood. In the second step, this graph is used to compute the *all-pair-shortest-path* matrix for the data points, which incorporates the geodesic distances between all data points. The final step is then to find a consistent lower-dimensional embedding that preserves the relative distances between all data points as good as possible, which is computed using *multidimensional scaling* (MDS).

In our work, we omit the calculation of the *all-pair-shortest-path* matrix and the embedding using MDS, which are unnecessary steps when interested in local prediction. These steps only serve to find a global embedding of the geodesic distance graph computed in the first step, and do not add additional information (in contrast, information might be lost in these steps when the dimensionality of the embedding is chosen too low). As a side effect, the computational savings of omitting the second and third step allow us to use more data points, resulting in a better approximation of the underlying manifold and thus in a better prediction. In detail, the computation of the geodesic distance graph of $n$ data points has time complexity $\mathcal{O}(n^2)$, whereas the computation of the *all-pair-shortest-path* matrix has a minimum time complexity of $\mathcal{O}(n^2 \log n + ne)$ (with $e$ corresponding to the number of edges in the graph) when using *Dijkstra's* algorithm with *fibonacci heaps* as priority queues, and MDS has time complexity $\mathcal{O}(n^3)$. We are thus able to reduce the computational costs associated with creating the pose manifolds from cubic to quadratic time complexity.

Let us now describe the spatio-temporal neighborhood graph that we use to represent human motion patterns in more detail. As in ST-Isomap, we use a directed graph structure instead of an undirected one to model the temporal progression of human poses. Training data is expected to arrive in chunks of sequential order. In a preprocessing step for the training data, we remove all poses that are not sufficiently different from their predecessor to ensure that the sequence of poses corresponds to actual motion. We do this by calculating the mean Euclidean distance between corresponding body joints of both poses, and we remove all poses where the mean distance is below $2$ cm. This also helps to normalize the speed of motions to some extent, as more frames will be removed during slow motions than during fast motions. After this preprocessing step, we create a sequential graph of vertices from the remaining training poses that is connected via directed edges from each vertex to its direct temporal successor.

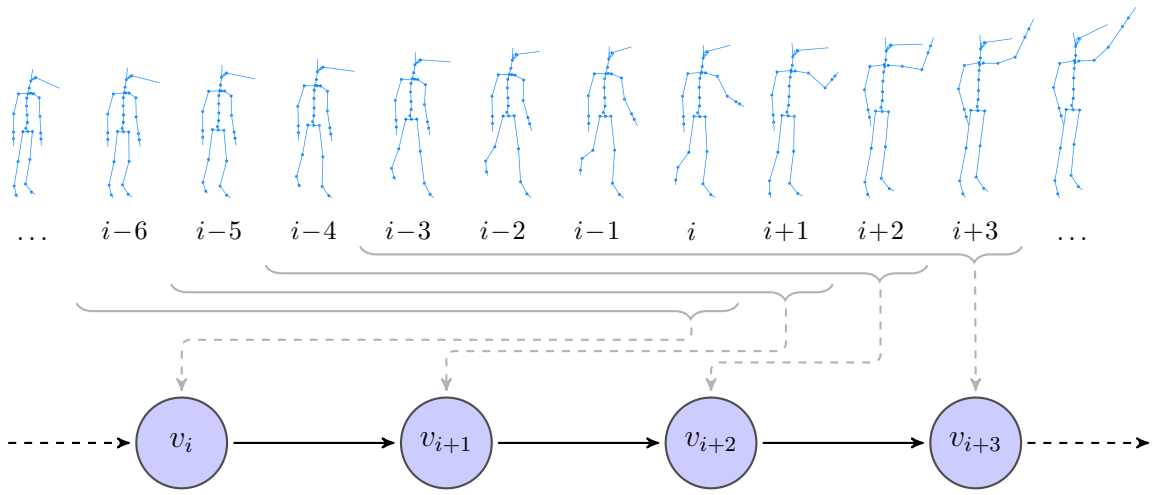


FIGURE 6.2 Sequential creation of a spatio-temporal neighborhood graph. The graph is created from an ordered temporal sequence of $s$ poses ($s = 7$ in this example). One vertex is created for each new pose and associated with a motion snippet based on the pose and its direct temporal history. Directed edges are added to encode the succession of poses. Note that the temporal windows encoded within each vertex are overlapping.

This process is illustrated in Figure 6.2. To capture the temporal structure of human motions, we associate each vertex not only with its corresponding pose, but with a motion snippet that consists of the pose and its short-term temporal history. Therefore, when computing the similarity of vertices, the corresponding temporal windows are compared. As we create a vertex for every pose in the training set, the temporal windows are overlapping (Figure 6.2). If necessary one can compute weights for each edge based on the distance measure of the motion snippets associated with the corresponding vertices. Although these weights are not needed for the prediction as presented in Section 6.1.2, they can be useful *e.g.* when deciding to create a global embedding of the graph structure.

As stated, each vertex $v_i$ is associated with a spatio-temporal motion snippet $\boldsymbol{\varsigma}^{(i)}$ that encodes a short-term motion pattern. It is these motion patterns that we compare when trying to estimate the current motion to predict the next pose. Figure 6.3 shows some examples of motion snippets. As can be seen, it is relatively easy for humans to assess the current motion and to predict the immediately following pose from these short pose sequences. Mathematically we describe a motion snippet as a vector $\boldsymbol{\varsigma}$ of body joint locations for $s$ consecutive poses:

$$\boldsymbol{\varsigma}^{(i)} = \left(\boldsymbol{\rho}^{(i,-(s-1))^{\mathrm{T}}}, \ldots, \boldsymbol{\rho}^{(i,-1)^{\mathrm{T}}}, \boldsymbol{\rho}^{(i,0)^{\mathrm{T}}}\right)^{\mathrm{T}} \tag{6.1}$$

$$\boldsymbol{\rho}^{(i,j)} = \left(\boldsymbol{\tau}_{\mathrm{BEC}}^{(i,j)^{\mathrm{T}}}, \ldots, \boldsymbol{\tau}_{\mathrm{FIR}}^{(i,j)^{\mathrm{T}}}\right)^{\mathrm{T}} \quad ; \qquad -s < j \leq 0 \tag{6.2}$$

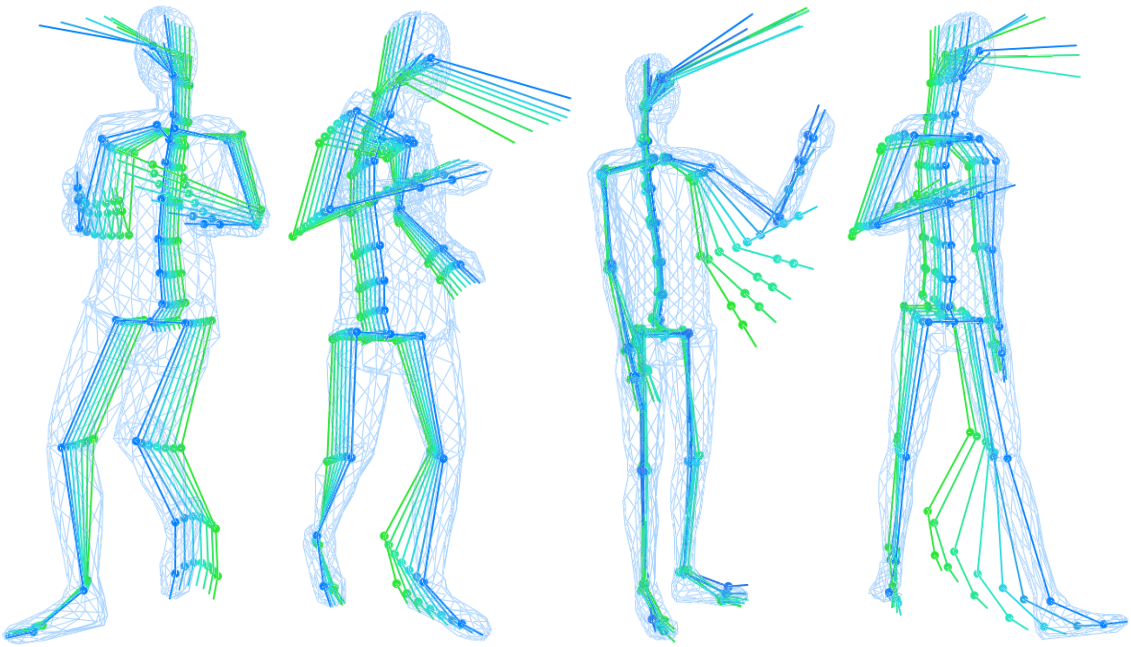

FIGURE 6.3 Examples of spatio-temporal motion snippets. Motion snippets encode short patterns of human motion as a vector of 3D body joint locations of the current pose and its short-term temporal predecessors (marked by the dots in the images). All coordinates are relative to the origin of the current pose (*i.e.* its pelvis position), so that motion snippets can be recognized independently of their absolute position.

Here, $\boldsymbol{\varsigma}^{(i)}$ represents the motion snippet for the $i$-th pose and is computed from poses in its short-term temporal pose history, *i.e.* from all poses in the time interval $[i-(s-1) : i\,]$. It is a concatenation of the 28 body joint locations $\boldsymbol{\tau}_{\text{bp}}^{(i,j)} = (x_{\text{bp}}^{(i,j)}, y_{\text{bp}}^{(i,j)}, z_{\text{bp}}^{(i,j)})^{\text{T}}$ of each of these $s$ poses. The dimensionality $d$ of a vector $\boldsymbol{\varsigma}$ is thus $d = 28 \cdot 3 \cdot s$. Note that the superscript $.^{(i,j)}$ of a parameter vector denotes the index $i$ of the motion snippet and the relative index $j \leq 0$ of the pose inside its temporal window ($j = 0$ corresponds to the current and most recent pose). The global index of a pose indexed in this way is therefore given as $i + j$.

An important aspect of spatio-temporal motion snippets $\boldsymbol{\varsigma}^{(i)}$ is that they are encoded in coordinates that are relative to the origin of the associated pose. In other words, all body joint locations are given in the pelvis coordinate system $\mathbf{H}_{\text{BEC}}^{(i)}$ of the $i$-th pose. In this way, motion snippets are independent of the position in the world at which they have been originally observed. They become more flexible as they can be detected at any place with any orientation, as long as the overall motion pattern is similar. This helps to improve the efficiency of the training set by removing the necessity to store similar patterns occurring at different spatial locations in the world. Note that all poses inside the temporal window of a motion snippet share the same coordinate system, so that the relative spatial extent of a motion is preserved

(thus the term *spatio-temporal*, see Figure 6.3). For each pose $i + j$ that is part of the temporal history of the pose $i$ associated with a motion snippet $\boldsymbol{\varsigma}^{(i)}$, we can compute the relative body joint locations $\boldsymbol{\tau}_{\mathrm{bp}}^{(i,j)}$ in Equation 6.2 from its body part coordinate systems $\mathbf{H}_{\mathrm{bp}}^{(i+j)}$ (see Equation 3.2) as follows:

$$\mathbf{H}_{\mathrm{bp}}^{(i,j)} = {\mathbf{H}_{\mathtt{BEC}}^{(i)}}^{-1} \cdot \mathbf{H}_{\mathrm{bp}}^{(i+j)} \quad ; \qquad -s < j \leq 0 \tag{6.3}$$

$$\mathbf{H}_{\mathrm{bp}}^{(i,j)} = \begin{pmatrix} \mathbf{R}_{\mathrm{bp}}^{(i,j)\star} & \boldsymbol{\tau}_{\mathrm{bp}}^{(i,j)} \\ 0 & 1 \end{pmatrix} \tag{6.4}$$

The relative body joint locations $\boldsymbol{\tau}_{\mathrm{bp}}^{(i,j)}$ correspond to the translation part of the homogeneous transformation matrix $\mathbf{H}_{\mathrm{bp}}^{(i,j)}$.

The distance measure $dist(\boldsymbol{\varsigma}^{(i)}, \boldsymbol{\varsigma}^{(k)})$ for comparing the vector representation of two motion snippets $\boldsymbol{\varsigma}^{(i)}$ and $\boldsymbol{\varsigma}^{(k)}$ is given as the mean Euclidean distance between all body joint locations encoded in the vectors:

$$dist(\boldsymbol{\varsigma}^{(i)}, \boldsymbol{\varsigma}^{(k)}) = \frac{1}{28 \cdot s} \sum_{j=0}^{s-1} \sum_{\mathtt{bp}=\mathtt{BEC}}^{\mathtt{FIR}} \| \boldsymbol{\tau}_{\mathrm{bp}}^{(i,j)} - \boldsymbol{\tau}_{\mathrm{bp}}^{(k,j)} \| \tag{6.5}$$

The distance calculation based on the Euclidean distances of the joint locations in $\mathbb{R}^3$ is a nice feature that gives us an intuitive measure for the similarity of two motion snippets. In Section 4.6.1 we have evaluated the mean Euclidean joint errors and thus the achievable accuracy of our tracking algorithm as approximately 2 cm. Thus we can assume that two motion snippets with a comparable distance are very similar. Such an intuitive distance measure helps to find good distance thresholds $\varepsilon_{sn}$ to detect similarities between motion patterns.

So far the graph corresponds to a linear list, as we only insert edges to denote the temporal succession in the training data. However, motion patterns frequently reoccur and we want to detect these cases and reflect the structural similarities in the graph. Consider the case where we observe a motion pattern several times in our training data, but the immediately following motions differ. This could happen *e.g.* when observing several instances where a person moves towards a table and prepares to grasp an object, but in one case the grasp is then performed with the left hand, and in another one it is performed with the right hand. Our graph should be able to capture these cases by providing multiple successors for the respective vertices. To do so, we adopt the strategy used by Jenkins and Matarić [74] in ST-Isomap. They distinguish between two important types of neighborhood relations (Figure 6.4). *Adjacent temporal neighbors* (ATN) are trivial relations where two poses are direct temporal neighbors. This

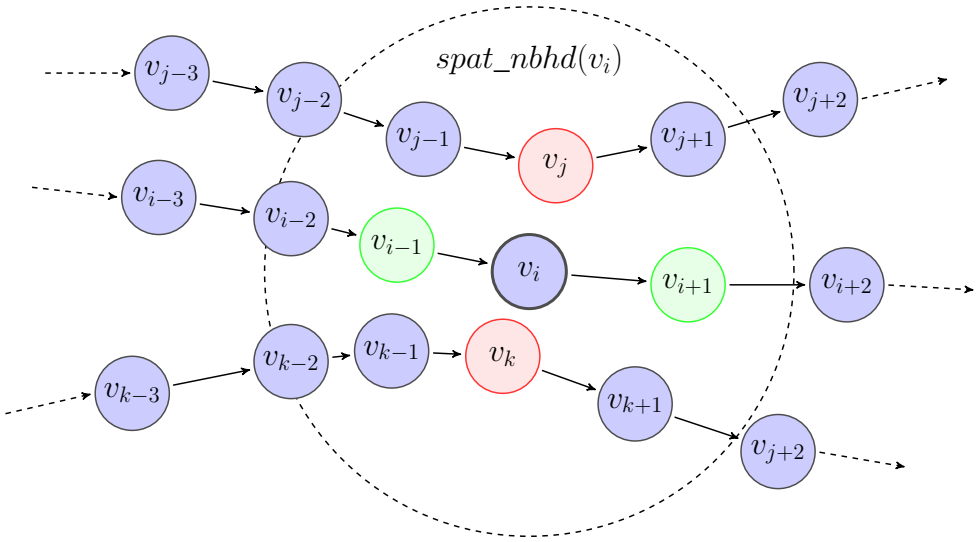


FIGURE 6.4 Neighborhood relations in spatio-temporal neighborhood graphs. The neighborhood relations for a vertex $v_i$ are defined by its spatial neighborhood $spat\_nbhd(v_i)$ and the temporal occurrence of vertices. *Adjacent temporal neighbors* (ATN) are trivial neighbors as they correspond to the direct temporal neighbors on the current trajectory (green vertices). *Common temporal neighbors* (CTN) are non-trivial neighbors that correspond to the spatially closest point on each temporally distant trajectory that passes through the spatial neighborhood $spat\_nbhd(v_i)$ (red vertices).

relation is already captured by our initial graph creation. *Common temporal neighbors* (CTN) are non-trivial neighbors that correspond to the spatially closest points on each temporally distant trajectory that passes through the spatial neighborhood of a vertex $v_i$. To provide a formal definition of CTNs, we first define the spatial neighborhood $spat\_nbhd(v_i)$ and the temporal neighborhood $temp\_nbhd(v_i)$ of a vertex $v_i$ as follows:

$$v_k \in spat\_nbhd(v_i) \quad \Leftrightarrow \quad dist(\boldsymbol{\varsigma}^{(i)}, \boldsymbol{\varsigma}^{(k)}) \leq \varepsilon_{sn} \tag{6.6}$$

$$v_k \in temp\_nbhd(v_i) \quad \Leftrightarrow \quad |i-k| \leq \varepsilon_{tn} \tag{6.7}$$

The spatial distance threshold $\varepsilon_{sn}$ defines whether two motion snippets are sufficiently similar to be spatial neighbors. In our experiments, we have set $\varepsilon_{sn}$ to 5 cm. The temporal distance threshold $\varepsilon_{sn}$ defines whether two motion snippets are considered temporal neighbors, which is the case when they were observed in short succession. We have set $\varepsilon_{sn}$ to 25. The definition

of CTNs for a vertex $v_i$ is then given as:

$$v_k \in CTN(v_i) \quad \Leftrightarrow \quad v_k \in spat\_nbhd(v_i) \tag{6.8}$$

$$\text{and} \quad v_k \notin temp\_nbhd(v_i) \tag{6.9}$$

$$\text{and} \quad \forall v_l \in temp\_nbhd(v_k): \quad dist(\boldsymbol{\varsigma}^{(i)}, \boldsymbol{\varsigma}^{(k)}) \leq dist(\boldsymbol{\varsigma}^{(i)}, \boldsymbol{\varsigma}^{(l)}) \tag{6.10}$$

To summarize, a vertex $v_k$ is a CTN of $v_i$ when they are spatially close (Equation 6.8) but temporally distant (Equation 6.9), and when $v_k$ is the closest point to $v_i$ in its temporal neighborhood (Equation 6.10). Two vertices are considered temporal neighbors when they are separated by at most $\varepsilon_{tn}$ frames. Whenever we detect CTNs, we connect the corresponding vertices with edges in both directions to reflect their relation in the graph. Although it would be slightly more accurate to connect each vertex with all the successors of its CTN, we prefer to add only the minimal amount of edges necessary to the graph.

The idea to use graphs for representing human motion is far from new. *Motion graphs* as introduced by Kovar *et al*. [84] are very similar in spirit and have been used in a generative way to create realistic and controllable motion. Arikan and Forsyth [8] and Lee *et al*. [90] have also shown related work on motion synthesis. Zhao and Yang [174] presented connected neighborhood graphs for use in an incremental variant of Isomap. Yamane *et al*. [169] create clusters of similar poses from sequential motion data and construct node transition graphs between these clusters to recognize and synthesize motions.

Let us conclude this section with a short discussion of alternative representations for motion snippets. We have already stated that we use the pelvis-relative representation of body joint locations instead of absolute body joint locations in the world reference frame to decouple motion patterns from their spatial occurrence. Independence of the spatial occurrence of a motion is also given when directly comparing the joint angle representation as given in Equation 3.8. However, the Euler representation of joint angles is redundant and does not lend itself to distance calculations. A better alternative is to use *quaternions* to represent the joint rotations (see Kuffner [87] for a related discussion). The remaining problem with this representation (and other representations based on relative orientations) is that the mean distance values are deceiving in the case of articulated structures. Small distances can indicate both a similar pose as expected but also a completely differing pose due to error accumulations. Thus, we believe it is best to use absolute values that share a local reference frame, as is the case in our representation.

### 6.1.2 Incrementally Learned Motion Patterns for Improved Prediction

The spatio-temporal neighborhood graphs as presented in the last section can be used to predict poses based on the currently observed motion. One advantage of using spatio-temporal neighborhood graphs when compared to global embeddings of training poses is that they are well suited for incremental construction. Whenever new pose data arrives in sequential order, we can append the vertices to the graph and connect all CTNs (see Section 6.1.1). Furthermore, due to the comparably low computational complexity of $\mathcal{O}(n^2)$ (which can be reduced even more when using approximate nearest-neighbor methods), the graph can grow quite big without becoming computationally intractable. We will now show how to combine this graph-based representation of human motion with the unconstrained tracking algorithm presented in Chapter 4 that has been proven to be effective in tracking arbitrary and unknown motions. By doing so we can improve the efficiency of our pose tracking algorithm in an unsupervised fashion and without the need for explicitly provided training data.

During the recursive estimation of human poses, we need to predict the pose for the current timestep based on the last pose estimate. In Section 4.2 we have presented a general motion model that predicts the new pose by adding random diffusion with body-part dependant variance to the last pose estimate. The variance of the diffusion must be large enough to capture fast movements in any direction. A more informed solution would be to add the random diffusion to a prediction of where the current pose is expected to be. When we can assure that this prediction is closer to the requested pose than the last pose, we could reduce the variance of the diffusion without sacrificing estimation accuracy. Therefore, we want to replace the uninformed motion model from Section 4.2 with a learned motion model whenever possible.

Assuming that we have already observed some motion snippets and incorporated them into our graph structure, we can distinguish between 3 relevant cases during each timestep of the estimation:

1. No motion is detected, *i.e.* the subject does not move.

2. An already known motion is observed.

3. An unknown motion is observed.

We treat these cases differently. Whenever no motion is observed at all, we can skip the estimation and just assume the last pose is still valid. Alternatively, we could perform one estimation step with very low variance to refine the last estimate. When we detect a familiar motion pattern that we have already learned, we can use this knowledge to create a potentially

accurate prediction and then perform an estimation step with low variance to refine our prediction. Whenever we observe a motion pattern that we cannot assign to any of the learned motion snippets, we have to stick to our unconstrained tracking algorithm with full variance from Chapter 4. We can then however incorporate the new pose estimate into the graph to extend our trained motion model, in case a similar pattern will be observed in the future. Algorithm 6.1 sketches the computational steps that are needed for improving human pose tracking with self-trained motion models.

**Algorithm 6.1** Human pose tracking with self-trained motion models.

```
1:  if NoMotionDetected() then                                   /* 1. No motion */
2:     RunTrackerMinimalVariance()
3:  else
4:     ς_c = GetCurrentMotionHistory()
5:     predicted_poses = ∅
6:     for all v_i do
7:        if dist( ς_c, GetMotionSnippet(v_i) ) ≤ ε_sn then
8:           predicted_poses = predicted_poses ∪ GetSuccessors(v_i, maxdepth)
9:        end if
10:    end for
11:    if predicted_poses ≠ ∅ then                                /* 2. Known motion */
12:       ClearParticleSet()
13:       AddToParticleSet( GetCurrentPose() )
14:       AddToParticleSet( predicted_poses )
15:       UpdateParticleWeights()
16:       RunTrackerLowVariance()
17:    else                                                       /* 3. Unknown motion */
18:       RunTrackerFullVariance()
19:       UpdateGraph( GetCurrentMotionHistory() )
20:    end if
21: end if
```

We start by testing whether motion is observed at all to catch the first case mentioned above. For this test we implemented a simple but effective change detection algorithm that takes the current and last images as input. From these we calculate difference images based on the pixelwise absolute difference between consecutive images (converted to greyscale). We then count all pixels whose absolute difference is greater than a threshold $\varepsilon_{ad}$ (we used $\varepsilon_{ad} = 50$ throughout). Whenever this pixel count is below another threshold $\varepsilon_{pc}$ (we used $\varepsilon_{pc} = 30$ in the kitchen sequences, but this has to be selected according to the image resolution), we

assume that no motion has occurred between the current and last frame. In these cases we run our tracker with a minimal variance and particle count (we use a 6th of the particles and the variance).

In cases where motion has been observed, we first compute the motion snippet $\boldsymbol{\varsigma}_c$ corresponding to the current motion history (line 4 of Algorithm 6.1). For this we take the recent history of poses estimated by the tracker and create the motion snippet based on the last $s$ poses according to Equations 6.1 to 6.4. In accordance with the preprocessing step before the creation of neighborhood graphs (Section 6.1.1), we filter all poses from the history that are not sufficiently different from their temporal successor. This results in an approximate temporal normalization of the observed motion. Whenever the history of poses is too small to create a motion snippet of size $s$, we fill the missing poses with duplicates of the oldest available pose.

Once the current motion snippet $\boldsymbol{\varsigma}_c$ has been calculated, we look for corresponding motion snippets in our neighborhood graph (lines 5 to 10). Whenever we find a vertex $v_i$ whose corresponding motion snippet differs from our current motion snippet by at most $\varepsilon_{sn}$, we add the pose associated with this vertex to a set of predicted poses. Additionally, we add all poses associated with the successors of $v_i$ up to a traversal depth of $maxdepth$ to this set (we have been using $maxdepth = 3$ in our experiments). The reason for this is that we want to provide several likely options to increase our chances of hitting a good prediction. Note that all predicted poses have to be transformed such that the relative motion between the matching vertex $v_i$ and the predicted successor of $v_i$ is added to the base pose of the current estimate. This is done by changing the base transformation $\mathbf{H}_{\text{BEC}}^{(p)}$ (*i.e.* the 6 d.o.f. pose in world coordinates) of the predicted poses as follows:

$$\mathbf{H}_{\text{BEC}}^{(p)\,\prime} = \mathbf{H}_{\text{BEC}}^{(c)} \cdot \mathbf{H}_{\text{BEC}}^{(i)\,-1} \cdot \mathbf{H}_{\text{BEC}}^{(p)} \tag{6.11}$$

Here, $\mathbf{H}_{\text{BEC}}^{(c)}$ corresponds to the base pose of the current estimate and $\mathbf{H}_{\text{BEC}}^{(i)}$ to the base pose of the matching vertex $v_i$. Effectively, Equation 6.11 aligns the pelvis positions of the current estimate and $v_i$ and adds the relative motion to the predicted successor. Also note that the time complexity of the correspondence search is linear with the number of vertices in the graph. Usually, this is not a problem unless the graph would grow beyond millions of vertices. In such cases, one could use approximate nearest-neighbor methods to find good matches in sublinear time [103]. For instance, one could restrict the correspondence search to the close proximity of the vertices that have been detected as matches in the previous timesteps.

When we have found corresponding vertices for the observed motion in our graph, *i.e.* the

set of predicted poses is not empty (line 11), we create a new particle set that incorporates these predicted poses. The new particle set combines the current pose estimate (line 13) with the set of predicted poses (line 14). We then perform one update step with the observation model presented in Section 4.3 to calculate weights for each of the particles according to the current observation (line 15). As the first step during BIHS tracking will be a weighted resampling of the particle set, good predictions will multiply in the particle set while bad predictions will be eliminated. Using this particle set, we then run a BIHS step of our tracker with reduced variance and particle count (we use a 4th of the particles and the variance). We can afford these computational savings compared to our unconstrained tracking due to the informed prediction of our motion model.

It should be noted that recognizing a common motion history does not guarantee that there will be a common future. However, we observed that motions do not deviate instantly, but rather over a short period of time. The reduced variance tracker is able to account for these incremental deviations until the motion history eventually differs from all known motion patterns, and we cannot provide good predictions anymore.

When the current motion does not match our trained data we switch to unconstrained tracking, *i.e*. we proceed with a regular BIHS step with full variance and the full particle count as presented in Section 4.5 (line 18). After the new pose has been estimated, we use it to update the graph, so that new motions are incrementally added to our motion model (line 19). We do this only if the new pose is sufficiently different from the last pose. Then, a new vertex is created from the updated motion history and connected to the graph. We keep track of the current position in the graph where the new vertex has to be inserted. When the last tracking steps have also been uninformed, we connect the vertex with the previously inserted vertex. If we have been tracking a known motion sequence however, and the new vertex is the first estimate that differs from a series of predicted poses, we connect the new vertex with the vertex that *won* the last prediction (we keep track of this in the particle set). Finally, to prevent the new motion to be immediately recognized as a correspondence in the following timesteps, we buffer newly created vertices and add them to the graph with a short delay of about 2 sec of recorded video. After that, repeated occurrences of similar motion patterns are likely to be detected in our graph, as our motion model gradually improves its predictive capabilities.

### 6.1.3 Online Activity Recognition

So far in this thesis we have shown how to automatically observe complex human activities in everyday environments. We are able to retrieve the sequence of pose parameters of a human model as a mathematical description of the observed motion. Such a representation is well-

suited for tasks such as animation of virtual characters, collision detection during human-robot interaction, or imitation learning in humanoid robotics. However, this description of human motion lacks the semantics that would allow us to automatically interpret the observed actions or activities and to reason about intentions at a higher level. Providing semantic labels to each frame is the task of activity recognition.

In our previous work on human activity recognition [125, 126] we have taken a discriminative and approach where we learned a direct mapping between observable image features and activities using *support-vector-machines* (SVM). As features we have used *global-point-feature-histograms* extracted from *space-time-shapes* [25, 170, 62]. *Space-time-shapes* correspond to 3D shapes that have been created by stacking extracted foreground silhouettes in temporal direction (Figure 2.8). Other discriminative approaches use holistic features such as silhouettes [27], edge images [164] or optical flow fields [48]. All of these methods have in common that they skip the intermediate estimation of human model parameters and operate directly on image features extracted from the cameras. While this is a fast and elegant approach, the downside is that image plane features are dependant on the viewpoint. The *space-time-shapes* used in our earlier approaches *e.g.* change completely when recorded from a different angle. Thus, activities in everyday environments can only be recognized well when they are performed at a global orientation that is comparable to that of the training data. Furthermore, model-free approaches to human activity recognition often fail to capture subtle nuances in the motions that happen to be important when observing *e.g.* everyday manipulation tasks. As a result, such approaches are mostly effective when dealing with clearly distinct actions such as punching, kicking or jumping jack, which is reflected in the benchmarks commonly used for evaluation [128, 25].

Another class of methods for activity recognition is model-based, where the parameters used for classification are joint angles or joint positions of articulated human models [133, 88]. The advantage of these methods is that the parameters are invariant to translation, scale and rotation. Furthermore, model-parameters better capture the characteristics of human movements especially when it comes to subtle variations. However, due to the difficulty in estimating the motion parameters, almost all methods presented to date use input data recorded by marker-based motion capture systems, which reduces the flexibility of these approaches.

We can use the learned graph structure and our representation of motion snippets to perform activity recognition directly on the motion data retrieved by our markerless tracker. Assuming that a semantic label has been assigned to each motion pattern, we can simply recover the labels whenever we detect a corresponding motion snippet for the current motion history $\varsigma_c$ during tracking. In detail, we are assigning the semantic label that corresponds to the best

matching vertex $v_i$ (respectively the vertex that *won* the prediction according to Section 6.1.2 when there are several matching vertices). If no good matches were found, *i.e.* when the distance of the best matching vertex $v_i$ exceeds the threshold $\varepsilon_{sn}$, we can still use the label assigned to $v_i$ for activity recognition. The reason behind this is that while we need precise matches of the motion history to provide an accurate prediction, activity recognition is less demanding when it comes to the quality of the match. A good solution is to choose a less restrictive value for the threshold $\varepsilon_{sn}$ (*e.g.* $3\times$ the value accepted for prediction) and otherwise classify the activities as unknown.

We will present experimental results on activity recognition in Section 6.2.3 to show the validity of our straightforward extension. Note that although we use a simple nearest-neighbor classification in our work as a proof-of-concept, future work will involve the use of graphical models such as *hidden markov models* (HMM) or *conditional random fields* (CRF) to incorporate transition probabilities between activity states. These extensions should be straightforward given the information contained in our learned graphs. We believe however that temporal aspects of human actions are to some extent already captured in the motion snippets due to their spatio-temporal design that incorporates the recent pose history. More sophisticated methods should nonetheless include additional context such as information about objects being manipulated or the state of the environment to further improve recognition rates [150].

## 6.2 Experimental Evaluation

We will now present several experiments to evaluate the impact of our improved motion models and their performance when used for activity recognition. In all of our experiments we have used the BIHS strategy with 800 particles and 10 initial layers as described in Section 4.5. When tracking with low variance due to the availability of a learned motion correspondence, only 200 particles with 4 initial layers were used, and the biomechanical intra-frame variances from Section 3.3.2 have been reduced by a factor of 4. The length $s$ of the temporal motion history has been set to $s = 7$.

### 6.2.1 Self-Trained Motion Models on the HUMANEVA 2 Data Set

In our first experiment we have evaluated the self-training capacity of our proposed motion model. We have chosen to use the HUMANEVAII S4 sequence due to the availability of ground-truth data that makes it possible to evaluate qualitative aspects of our method. Furthermore, the sequence consists of three different classes of repetitive actions that make it

well-suited for evaluating our incremental learning strategy. During the first 380 frames of this sequence, the subject is *walking* slowly in a circle, before accelerating into a *running* motion that lasts until frame 820. The final segment consists of a random *exercise* that shows no clear repetitive patterns.

This experiment is a repetition of the experiments in Section 4.6.2, but this time we replaced the general motion model from Section 4.2 with our self-training motion model as described in Section 6.1.2. No training data has been provided for this sequence, *i.e.* the tracking started with an empty neighborhood graph that was incrementally growing as new motions were estimated.

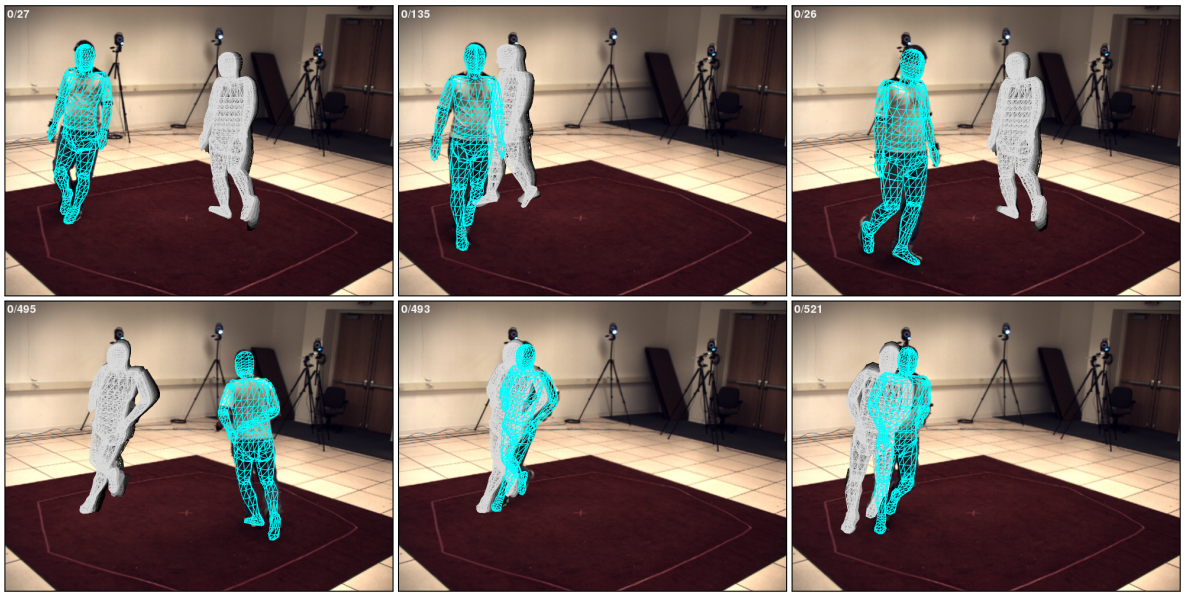


FIGURE 6.5 Screenshots from the HUMANEVAII S4 sequence when using incrementally learned motion models for improved prediction. The detected motion snippet correspondences (greyscale) have been trained earlier in the sequence and increase the computational efficiency when dealing with previously observed motions. The tracker first learns a sequence of motion snippets corresponding to *walking*, and later a sequence corresponding to *running* (see also Figure 6.6). As can be seen, motion snippets are detected independently of their absolute position, which increases detection rates and improves motion prediction.

Figure 6.5 shows some examples from the processed sequence where the self-trained motion model was used to predict the motion for the next timestep. The final pose estimate is shown in blue, whereas the learned motion snippet that was used for prediction is shown in greyscale, with the temporally preceding poses fading to black. Due to the incremental learning of our motion model, the detected motion snippets correspond to motion patterns that occurred earlier in the same sequence. As can be seen, the global occurrence of the pattern is not encoded in our motion snippet representation. This allows us to detect matching patterns independent of the

location they have been observed at, which increases the detection rates for correspondences and reduces the redundancy of motion snippets stored in the graph.

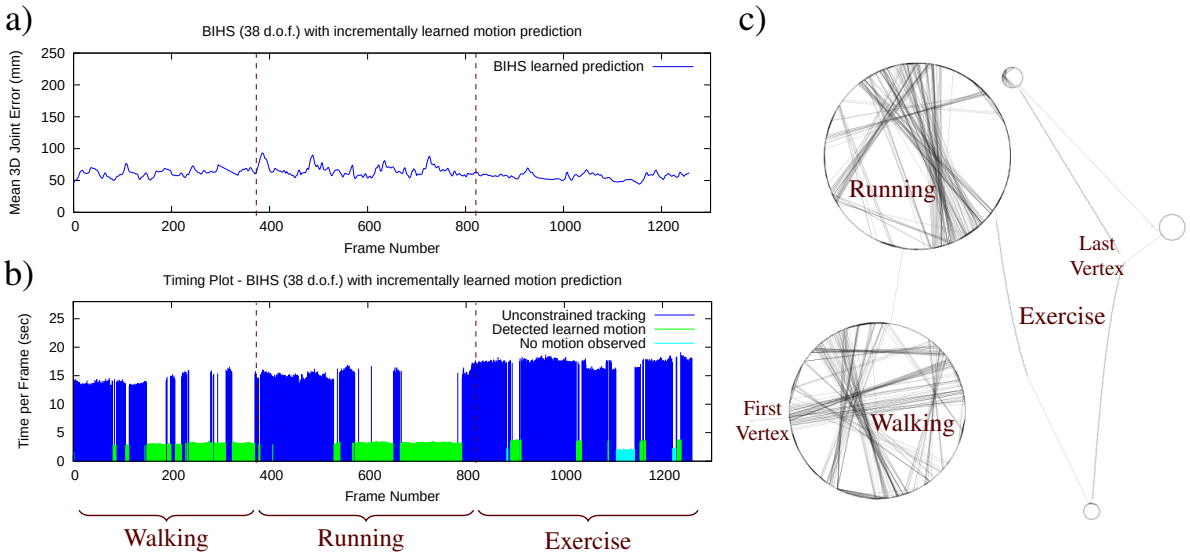


FIGURE 6.6 Errorplot, timingplot and learned motion graph for the HUMANEVAII S4 sequence when using incrementally learned motion models for improved prediction: a) the errorplot shows constant accuracy over the whole sequence b) the timing plot shows the improved efficiency of our tracker once models of the walking and running motion have been learned c) a 2D visualization [171] of the learned motion graph (715 vertices, 1617 edges) shows the clear separation between the walking, running and exercise motions, indicating its potential use for motion segmentation and action recognition. Note that the graph has been self-trained without any prior information on the motion.

We have analyzed the processed sequence in detail to see how our new motion model influences the quality of the pose estimates and the computational efficiency of our tracker. Figure 6.6a shows the errorplot with the mean Euclidean joint errors for our pose estimates that has been computed from the ground-truth data. The quality is comparable to that of our previous experiments in Section 4.6.2 and shows a constantly low mean error of slightly above 5 cm (this includes a systematic error due to different relative joint positions between the ground-truth and our model). We did not expect the quality to improve when using the learned motion models, as the parameters of our tracker are set such that they are able to retrieve accurate estimates even in the uninformed case. Instead, we expect that the computational efficiency of our tracker improves once it can access the learned motion patterns. As shown in Figure 6.6b, this is exactly what happens once motion snippets for the walking motion and for the running motion have been self-trained by the model. When no correspondences are found, the estimation time per frame is around 15 sec. This changes once motion correspondences are detected and the tracker is able to switch to low-variance mode. In these cases, estimation

time drops to around 2.5 sec per frame. When transitioning from walking to running, and later from running to the exercise motion, the new motion patterns force the tracker to switch to the general motion model until the new motions have been trained and added to the graph. Due to the random nature of the exercise at the end of the sequence, only a few correspondences are found and most of this segment has to be processed using the general motion model. Overall, 40.1% of the frames in the sequence have been processed at less computational expense due to the self-training of the repetitive motion patterns.

The immediate loss of motion snippet correspondences when transitioning from walking to running and from running to exercise indicates that these motions show strong stylistic differences. In Figure 6.6c we have plotted the spatio-temporal neighborhood graph in two dimensions to see if these differences are reflected in the structure of the graph. As can be seen, both the walking and the running segments are directly recognizable when looking at the visualization. All edges that connect CTNs are restricted to the respective subgraphs that contain all vertices for either the walking or the running motion. These two subgraphs are connected by just a bridge. The vertices corresponding to the exercise motion are less distinctive due to the non-repetitive pattern of the exercise. The visualization shows that spatio-temporal neighborhood graphs have the potential to be used for motion segmentation or activity recognition, as structural similarities are reflected well in the graph.

### 6.2.2 Improved Prediction on the TUM KITCHEN Data Set

In a next step, we have evaluated our strategy of learning environment-specific motion models to improve the prediction and thus the computational efficiency of our tracker in a series of experiments on the TUM KITCHEN Data Set. This data set has been introduced in Section 5.3.3 and consists of several instances of table setting tasks performed by different subjects. The activities are more complex than in the HUMANEVAII Data Set and feature a variety of manipulation actions that involve objects and dynamic parts of the environment such as cupboards or drawers. We have trained the motion model using the motion capture data collected with our BIHS tracker as described in Section 5.3.3.

In Figure 6.7 we show some examples of typical matches retrieved during tracking. In this case we have trained our model with sequence 1-5 which corresponds to a long sequence (6505 frames) of repetitive pick and place actions in the kitchen. The learned motion model was then used to test the prediction on sequence 1-0. Again, the original training correspondences are shown in greyscale and the final pose estimates are shown in blue. One striking observation made in this kitchen environment is that a lot of similar motion patterns are repeatedly detected at different spatial locations. As an example, the motion patterns for opening a

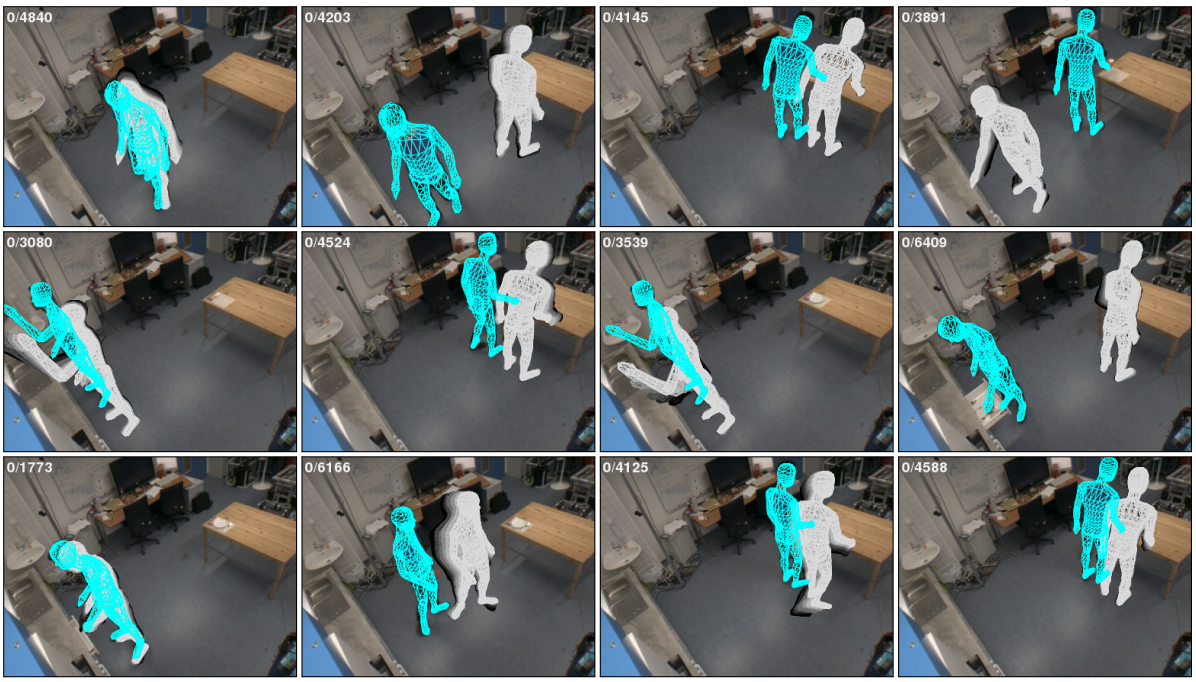


FIGURE 6.7 Screenshots from a kitchen sequence with corresponding motion snippets used for prediction. The trained motion snippets are shown in greyscale, with darker parts corresponding to temporally preceding poses. Motion snippets are detected independently of their absolute position, which increases detection rates and improves motion prediction.

cupboard are exchangeable for most of the overhead cupboards, as only the base position of the person varies. The motion patterns for taking or putting down objects are also remarkably similar regardless of whether they have been observed at the table or at the kitchen.

We also evaluated the influence of the size of the training set on the predictive capabilities of our motion model. For this, we have taken five sequences of the same subject (1-0 to 1-4) and incrementally evaluated the results for tracking sequence 1-1 to 1-4 with motion models learned on all preceding sequences (*i.e.* sequence 1-1 trained with 1-0, 1-2 trained with 1-0 and 1-1 *etc.*). Thus, the size of the training set for evaluating sequence 1-1 is only about a quarter of the size used for evaluating sequence 1-4. All of the sequences consist of table-setting activities, although the order of the subactions is not fixed. Figure 6.8 depicts the timing plots for each of these 4 test runs. As can be seen, the largest part of all frames could be processed with reduced variance and particle count due to detected motion correspondences between the sequences. In numbers, 84.2% of the frames in sequence 1-1, 79.1% in sequence 1-2, 80.8% in sequence 1-3, and 89.0% of the frames in sequence 1-4 could be processed more than 4 times faster. As it seems, the size of the training set does not necessarily lead to better detection rates. Although the best results were achieved with the largest training set, sequence 1-1 with

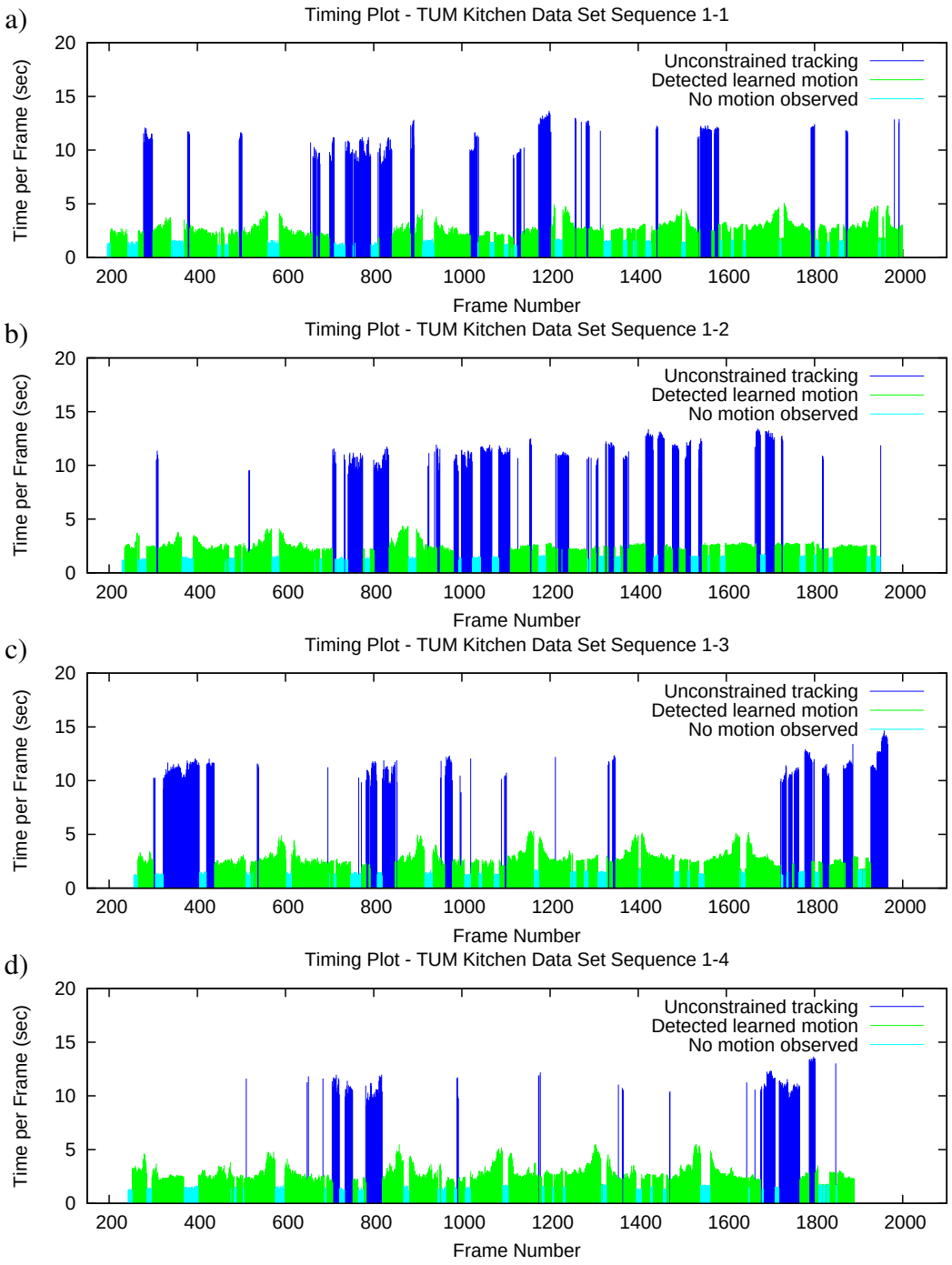


FIGURE 6.8 Processing times per frame for the TUM KITCHEN Data Set when using learned motion models for prediction. Shown are the processing times in seconds per frame for a) sequence 1-1 (trained with sequence 1-0) b) sequence 1-2 (trained with sequences 1-0 and 1-1) c) sequence 1-3 (trained with sequences 1-0, 1-1 and 1-2) and d) sequence 1-4 (trained with sequences 1-0, 1-1, 1-2 and 1-3). Blue values correspond to frames where no reliable prediction could be made and the tracker switched to unconstrained mode.

the smallest training set came of second best. Therefore, even small training sets seem to improve the efficiency of the tracker significantly.

The overall quality of the pose estimates during all our experiments has been on par with our previous method that does not make use of learned motion models. The informed prediction makes it possible to compensate for the reduced particle count and the lower inter-frame variances. Note that when using the same low-variance sampling with reduced particle count but *without* the improved prediction given by the learned motion models, the tracking results are significantly worse and even fail without recovery during fast turning motions of the subjects. Using the learned motion models on the other hand, tracking results are accurate and free of errors except for occasional problems with the arms. As noted in Section 5.3.3, these problems mainly occur when subjects reach into the rightmost cupboard with their right hand, due to a blind spot not covered sufficiently by our camera setup. Unfortunately, the trained models of these actions were unable to correct these problems as one would have expected. This seems to be due to the fact that the low-variance tracking step performed after an informed prediction is sufficient to pull away the final estimate from a correct prediction in cases where the global maximum of our weight function does not correspond to the ground-truth pose, *i.e.* the observations are misleading. Omitting the low-variance tracking step does not help either, as this would result in a reduced overall accuracy of our tracker. We plan to further investigate this issue in our future work so that the learned motion models will not only be used to increase the computational efficiency of our tracker, but also to improve the quality in such difficult cases.

Note that the computational overhead of finding good predictions in the learned graph is negligible when compared to the costs of the particle evaluations. The largest graph structure created during our experiments (trained with sequences 1-0, 1-1, 1-2, 1-3 and 1-5) consisted of more than 13440 vertices, and the (exact) correspondence search took around 70 ms per frame. This is by far outweighed by the computational savings of the improved prediction.

### 6.2.3 Activity Recognition on the TUM KITCHEN Data Set

As discussed in Section 6.1.3, spatio-temporal neighborhood graphs are not only useful to improve the prediction of human motions, but they are also well-suited for activity recognition. We will now show some results on activity recognition for the sequences 1-1 to 1-4 of the TUM KITCHEN Data Set. The results have been derived from the same experiments that are described in Figure 6.8.

Although one could use fully unsupervised methods that segment observed motions into several clusters and then assign labels to each of these clusters, we will assume that the training

sets have been labeled in advance. In the TUM KITCHEN Data Set, we have provided ground-truth labels for the actions of the left hand, the right hand, and the trunk. The labels are provided separately to take into account the high degree of parallelism that is common in the observed activities. Often, one hand is opening a cupboard door, while the other one is reaching towards a cup, and while the body is still moving towards the cupboard. Details about the exact semantics of the labels and the corresponding ontologies are given in [150].

Each of the sequences 1-1 to 1-4 in the following experiments corresponds to one observed instance of a table setting task of the same subject. The table setting activity can be further segmented into several subactions, such as taking something, carrying something, or putting down something. These subactions match the granularity of the semantic labels. Note that we make no assumptions about the frequency or temporal succession of observed actions. The motion models for each of our test sequences have been trained with other instances of table setting activities from the same subject (sequence 1-1 has been trained with sequence 1-0; sequence 1-2 has been trained with sequences 1-0 and 1-1; sequence 1-3 with sequences 1-0, 1-1 and 1-2; sequence 1-4 with sequences 1-0, 1-1, 1-2 and 1-3). Semantic labels for each frame of the test sequence have then been assigned during the pose estimation by retrieving the labels of the best-matching motion snippets, as described in Section 6.1.3.

Figures 6.9, 6.10a and 6.10b show the resulting confusion matrices for the left hand, the right hand, and the trunk labels of the tested sequences. As can be seen, most of the retrieved labels are assigned correctly to the ground-truth labels. Typical confusions for the left hand actions include *reach* that is often misclassified as *idle carry* (which corresponds to a catch-all class), and the *open drawer* and *close drawer* actions that are frequently confused. Furthermore, the *release* action is often falsely classified. For the right hand actions, the most common errors include *take* which is misclassified as *reach*, *release* which is misclassified as *put*, and *put* and *reach* which are misclassified as *idle carry*. We have investigated these confusions by comparing the motion snippets corresponding to the falsely assigned labels, and have found some typical causes for the misclassifications. First, the semantic labels take into account the handling of objects, *e.g.* the *reach*, *take*, *put* and *release* labels implicitly assume that an object is being manipulated. However, as we only compare the motion patterns of a human without taking into account any object detections, these actions cannot always be clearly distinguished. In addition, *take* actions always follow immediately after *reach* actions, and *release* actions follow immediately after *put actions*. Because the transitions are seamless, this is a common cause of confusion. Another problem is that the temporal extent of motion snippets is far shorter than the extent of the labeled actions. Thus, it can happen that similar motion snippets are observed for different actions. As an example, the *open drawer* and *close*

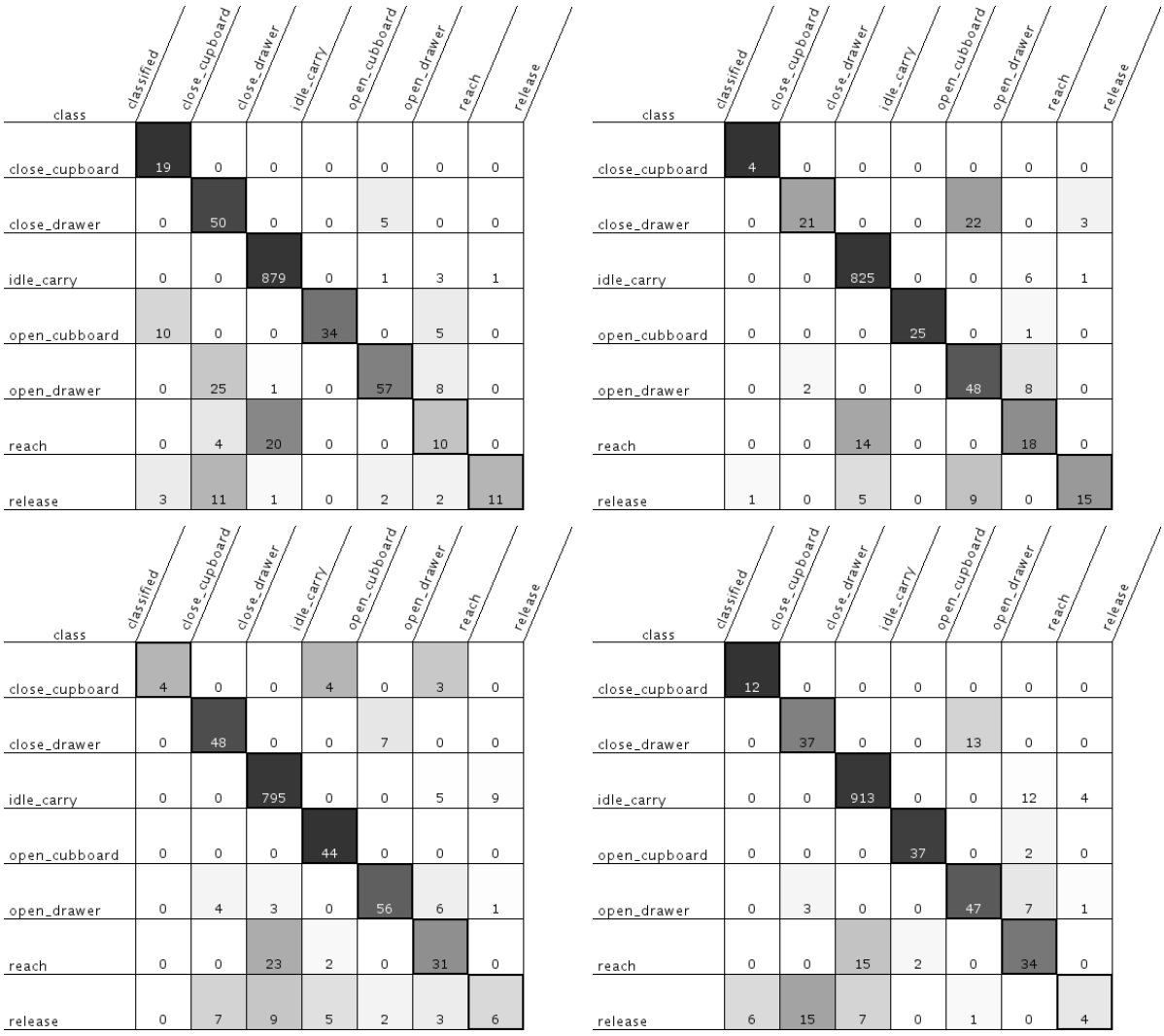


**FIGURE 6.9** Confusion matrices for activity recognition on the TUM KITCHEN Data Set. Shown are the matrices for the left hand labels of sequence 1-1 (trained with sequence 1-0), sequence 1-2 (trained with sequences 1-0 and 1-1), sequence 1-3 (trained with sequences 1-0, 1-1 and 1-2) and sequence 1-4 (trained with sequences 1-0, 1-1, 1-2 and 1-3).

*drawer* actions share similar motion patterns. When opening, the pattern corresponds to reaching for the drawer and pulling it out. When closing, it corresponds to pushing the drawer and retracting the arm. Both actions share the almost identical motion pattern of moving the arm forward then backward again (once with and once without the drawer handle being grasped). To be able to distinguish between these actions, it would be necessary to model the temporal succession of motion snippet detections (*e.g.* using HMMs), or to add complementary sensor input from RFID readers (for object detections) or magnetic sensors (for detecting the states of cupboards or drawers). Note that the observed confusions are irrelevant when using the mo-

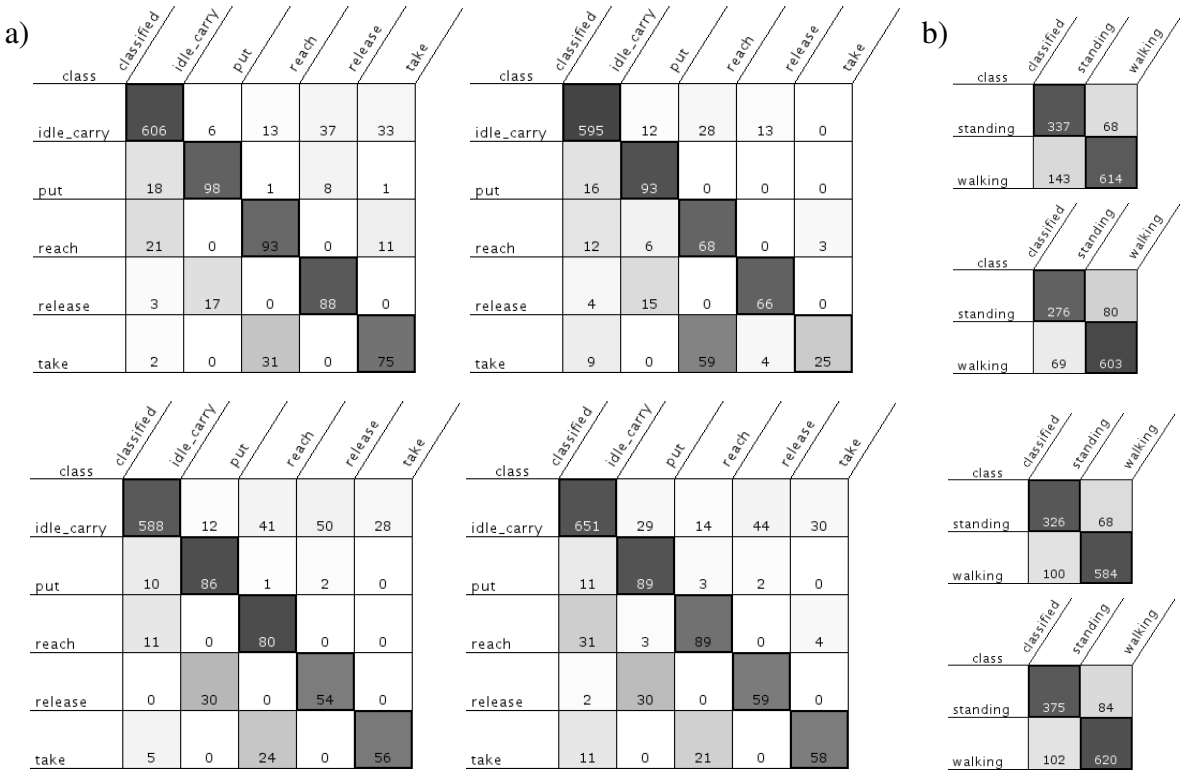


**FIGURE 6.10** Confusion matrices for activity recognition on the TUM KITCHEN Data Set. Shown are the matrices for a) the right hand labels and b) the trunk labels of sequence 1-1 (trained with sequence 1-0), sequence 1-2 (trained with sequences 1-0 and 1-1), sequence 1-3 (trained with sequences 1-0, 1-1 and 1-2) and sequence 1-4 (trained with sequences 1-0, 1-1, 1-2 and 1-3).

tion snippets for prediction, due to the similar motion patterns that result in similar predictions for the tracker.

In addition to the confusion matrices, we also provide *precision* and *recall* rates for all action labels. These are given in Table 6.1 for the left hand, in Table 6.2 for the right hand, and in Table 6.3 for the trunk labels. Precision and recall rates are defined as follows:

$$precision = \frac{TP}{TP + FP} \quad ; \qquad recall = \frac{TP}{TP + FN} \tag{6.12}$$

The precision rate is a measure of the exactness of the classification and is defined as the ratio of true positives $TP$ to all detections of a class (*i.e.* the sum of true positives $TP$ and false positives $FP$). The recall rate is a measure of completeness that states how many of the possible detections have been correctly classified. This corresponds to the ratio of true positives $TP$ to the number of all frames in the test sequence that belong to the respective class (*i.e.* the sum of true positives $TP$ and false negatives $FN$). The overall precision and recall

| | Sequence 1-1 Precision | Recall | Sequence 1-2 Precision | Recall | Sequence 1-3 Precision | Recall | Sequence 1-4 Precision | Recall |
|---|---|---|---|---|---|---|---|---|
| Close Cupboard | 74.1 % | 55.6 % | 90.0 % | 55.5 % | 91.3 % | 67.1 % | 69.4 % | 54.8 % |
| Close Drawer | 59.8 % | 92.4 % | 85.0 % | 58.6 % | 79.5 % | 89.2 % | 68.3 % | 66.2 % |
| Idle Carry | 96.3 % | 98.7 % | 97.8 % | 98.3 % | 92.9 % | 97.1 % | 97.5 % | 98.4 % |
| Open Cupboard | 82.4 % | 81.1 % | 72.4 % | 95.9 % | 83.5 % | 100.0 % | 74.1 % | 97.7 % |
| Open Drawer | 87.9 % | 71.9 % | 63.2 % | 80.6 % | 83.3 % | 83.3 % | 83.3 % | 85.3 % |
| Reach | 63.3 % | 54.8 % | 66.3 % | 77.2 % | 62.3 % | 71.1 % | 75.3 % | 77.1 % |
| Release | 75.0 % | 65.9 % | 82.9 % | 48.6 % | 85.7 % | 54.5 % | 85.4 % | 43.8 % |
| Overall | 89.6 % | 89.6 % | 90.1 % | 90.1 % | 87.6 % | 87.6 % | 91.1 % | 91.1 % |

TABLE 6.1 Precision and Recall rates for the left hand labels.

rates given in the tables are computed by summing up the true positives, false positives and false negatives over all classes before calculating the overall rates according to Equation 6.12. Note that this results in identical values for the precision and the recall rates, as each $FP$ belonging to one class will always correspond to a $FN$ of another class. Thus, the overall values can be considered a general score for the accuracy of the classification. For the left hand labels, we achieved accuracies of around $90\%$. For the right hand labels, the accuracies have been around $82\%$, and for the trunk labels around $86\%$. Given the complex set of manipulation activities and the subtle differences between the corresponding motion patterns, we believe that these results are quite promising. As discussed, we expect a further boost of classification rates when considering transition probabilities (*e.g.* by using HMMs) or when incorporating complementary sensor data. Note that we considered every frame of the test sequence when calculating the precision and recall rates by selecting the semantic label associated with the best-matching motion snippet. This is in contrast to the calculation of the diffusion matrices, where we only considered frames whose best-matching motion snippet was good enough to be used for prediction (*i.e.* its distance to the current motion history was below $\varepsilon_{sn}$). This is a good indication that activity recognition still works well even when the training data would be unsuited for accurate predictions of human motion.

The experiments presented so far have been trained and tested on the same subject. We also tested how well activity recognition works when the subject differs between the training sequences and the test sequence. To do so, we have performed another evaluation where we used sequence 1-7 of subject S03 from the TUM KITCHEN Data Set as a test sequence. The motion model has been trained using sequences 1-0 and 1-5 of subject S01. It should be noted

| | Sequence 1-1 Precision \| Recall | | Sequence 1-2 Precision \| Recall | | Sequence 1-3 Precision \| Recall | | Sequence 1-4 Precision \| Recall | |
|---|---|---|---|---|---|---|---|---|
| Idle Carry | 91.9 % | 88.3 % | 90.0 % | 90.5 % | 93.8 % | 82.9 % | 92.0 % | 84.0 % |
| Put | 86.7 % | 84.5 % | 81.5 % | 86.8 % | 73.4 % | 90.8 % | 73.6 % | 91.7 % |
| Reach | 61.5 % | 81.3 % | 49.3 % | 77.2 % | 59.0 % | 86.5 % | 67.8 % | 77.1 % |
| Release | 73.2 % | 86.2 % | 77.0 % | 71.9 % | 53.8 % | 64.8 % | 69.9 % | 76.4 % |
| Take | 71.3 % | 86.2 % | 87.4 % | 41.3 % | 67.1 % | 66.3 % | 53.7 % | 50.7 % |
| Overall | 82.8 % | 82.8 % | 82.4 % | 82.4 % | 81.3 % | 81.3 % | 80.5 % | 80.5 % |

TABLE 6.2 Precision and Recall rates for the right hand labels.

| | Sequence 1-1 Precision \| Recall | | Sequence 1-2 Precision \| Recall | | Sequence 1-3 Precision \| Recall | | Sequence 1-4 Precision \| Recall | |
|---|---|---|---|---|---|---|---|---|
| Walking | 88.6 % | 80.6 % | 83.9 % | 89.8 % | 89.4 % | 82.7 % | 87.9 % | 85.3 % |
| Standing | 80.8 % | 88.8 % | 89.7 % | 83.7 % | 83.1 % | 89.6 % | 86.6 % | 89.0 % |
| Overall | 84.5 % | 84.5 % | 86.7 % | 86.7 % | 86.1 % | 86.1 % | 87.2 % | 87.2 % |

TABLE 6.3 Precision and Recall rates for the trunk labels.

that the two subjects differ in body height by about $25$ cm and that their style of motion is very different. To normalize the motion patterns, we have performed all calculations of the body joint locations of each motion snippet based on a mean human model as computed in Section 3.2. Figure 6.11 shows the corresponding confusion matrices where the best-matching label has been assigned to each frame. As expected, the confusions are much more evident than in the single subject experiments. Still, the overall precision and recall rates for this cross-subject evaluation amount to $74.0\%$ for the left hand actions, $78.8\%$ for the right hand actions, and $76.7\%$ for the trunk actions. None of the detections were good enough to be used for improving the prediction of our motion tracker though, *i.e.* the best matches were constantly above our distance threshold $\varepsilon_{sn}$. We take these results as an indication that it is best to learn person-specific motion models and to switch between these models based on an external recognition component. Alternatively, it should be possible to mix motion patterns of different subjects in a single spatio-temporal neighborhood graph, but further investigation is needed to evaluate all implications of this strategy.

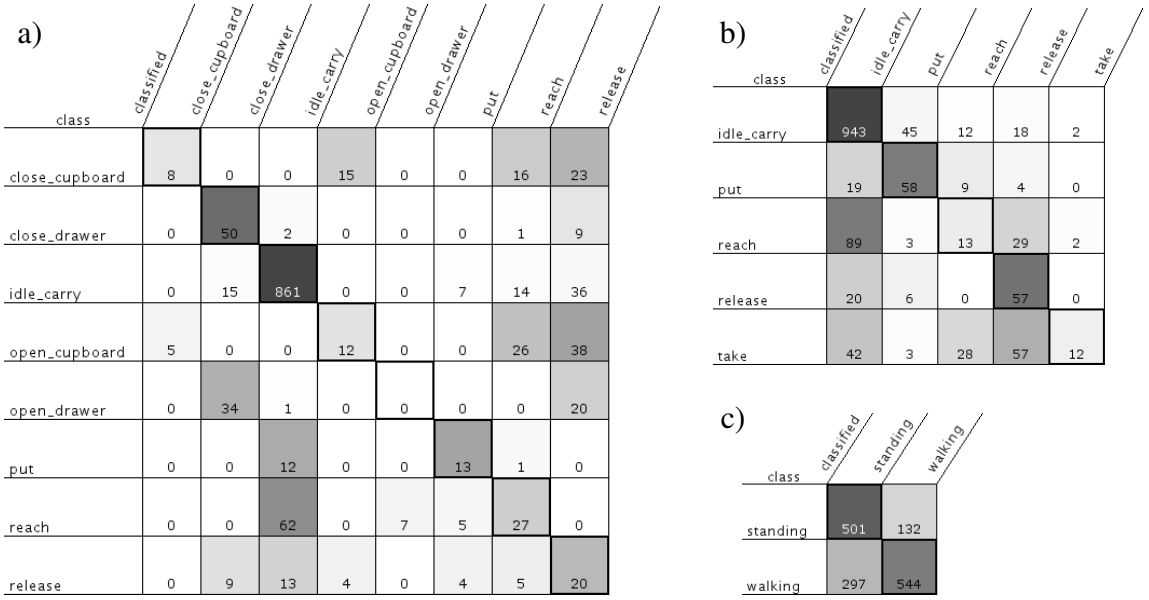


FIGURE 6.11 Confusion matrices for cross subject activity recognition on the TUM KITCHEN Data Set (sequence 1-7 of subject S03). The motion model has been trained with sequences 1-0 and 1-5 of subject S01. Confusion matrices are shown for a) the left hand b) the right hand and c) the trunk labels. Note that the training and test subjects differ in body height by approximately 25 cm and that their motion patterns seem highly dissimilar.

## 6.3 Summary

In this chapter we have shown how environment-specific motion models can be used to improve the efficiency of our human motion tracker without loss of accuracy. These motion models are learned incrementally in an unsupervised manner using the general but inefficient motion model from Section 4.2. Over time, the tracker adapts itself to the environment and improves its ability to predict and recognize motions and activities that are typical for this environment.

Observed motions are stored in a spatio-temporal neighborhood graph to create a manifold of poses relevant to a specific environment. This directed graph represents poses as motion snippets that correspond to the pose and its short-term temporal history. Therefore, similar poses can be discriminated by their spatio-temporal context. Learned motion snippets are detected independently of their spatial location, which increases the efficiency of the training data without the need to store redundant information. Spatio-temporal neighborhood graphs are directly related to ST-Isomap [74], which has already been shown to be an effective representation of human activities.

The self-trained motion models are used to improve the prediction of our tracker which

allows us to switch to a low-variance tracking mode that requires less particles to achieve the same high-quality estimates as our uninformed tracker. Whenever unknown motions are observed, we switch back to our general but less efficient tracker. The new estimate is then used to update the motion model to improve the prediction in the case of repeated future observations. The efficiency of this scheme has been evaluated in several experiments on the HUMANEVAII Data Set and the TUM KITCHEN Data Set.

Finally, we have shown that our learned motion models are well-suited for activity recognition. This has been demonstrated on our challenging TUM KITCHEN Data Set where we have achieved recognition rates between $80\%$ and $90\%$ percent ($75\%$ in the case of cross-subject recognition). Activity recognition works well even with comparably small training sets, and we believe that potential for further improvements is given by incorporating complementary sensor data from RFID readers or magnetic sensors. Such data can add valuable information about the state of objects and the environment, which helps to put the observed motion patterns into context. The semantic labels retrieved for each frame during activity recognition finally allow us to not only observe human motions, but to interpret them and to reason about ongoing human activities.

# CHAPTER 7

# Conclusion

In this thesis we have presented our approach on automated observation and interpretation of human activities. We will now conclude with a brief summary of our contributions and a discussion of open problems and promising future work.

## 7.1 Summary

The ability to understand human behavior and to reason about actions and intentions of humans during their everyday work is an important milestone in the creation of truly *intelligent autonomous systems*. In the preceding chapters of this thesis we have presented our contributions towards this ever-challenging goal.

We started in Chapter 2 by giving an overview on the state of research and recent trends in the area of human motion capture and activity recognition. Besides the discussion of related work, we also sketched the basics of our own system to provide the reader with an intuitive understanding of the concepts presented in this work.

In Chapter 3 we introduced the RAMSIS model from the ergonomics community in the context of human motion capture and presented various optimizations to adapt the model to the specific requirements in this field of research. Our modifications include a reduction of the dimensionality of the pose parameters without loss of expressivity, the incorporation of biomechanical angular velocity limits to restrict the search space during consecutive pose estimation, and the caching of body-part dependant pose calculations to boost efficiency during repeated hierarchical computation. In addition, we have learned a reduced set of shape parameters to facilitate model adaptation to different subjects. The resulting digital human model is a detailed yet efficient tool for human motion tracking.

Chapter 4 constitutes the central chapter of this thesis and our core algorithmic contributions to the markerless tracking of human motions. After an introductory discussion of probabilis-

tic methods for state estimation, we motivate our choice of a *particle filter* framework and present corresponding motion and observation models that are well-suited for tracking arbitrary types of human motion. As standard *sampling importance resampling* (SIR) particle filters are computationally intractable due to the high dimensionality of articulated human models, we introduced a novel hierarchical sampling strategy for particle filters termed *branched iterative hierarchical sampling* (BIHS). Our method combines the complementary strengths of two state-of-the-art approaches to articulated pose tracking, namely the hierarchical decomposition of the state space known from *partitioned sampling* [94] and the stochastic search strategy introduced in the *annealed particle filter* [46]. In addition, a parallel evaluation of partitioning schemes increases the diversity during particle evaluations and leads to improved robustness by reducing the risks of getting stuck in local maxima of the weight function. We have validated the improvements over previous approaches to articulated pose tracking and the generality of our method on several challenging sequences with and without ground-truth data.

In Chapter 5 we have shown how to extend the tracking of single subjects in a controlled environment to the tracking of complex manipulation activities and human interactions in everyday environments. For that we introduced color-based appearance models to augment the human model presented in Chapter 3. The color information can be used to complement our shape-based observation model in cases where shape information is ambiguous (*e.g.* for the head). It is also an integral part of *layered observation models* that we use to implicitly model the environment without the need to provide 3D models. We showed in several experiments how this enables us to track subjects (•) that manipulate objects or parts of the environment such as doors, (•) that are being occluded or occlude parts of the environment such as tables, and (•) that interact with other subjects. The proposed methods are accurate and reliable to the extent that we used them to create the publicly available TUM KITCHEN Data Set of complex manipulation activities in a kitchen setting (currently 21 sequences from 4 different subjects).

In our last chapter we have discussed how to learn models of human motion. In this context, we presented a method that is capable to learn environment-specific motion models over time to improve the predictive capabilities of our tracker and thus its efficiency. This unsupervised method makes use of *spatio-temporal neighborhood graphs* as an efficient and easily extendable representation of pose manifolds. In addition, we have shown that the learned motion models are well-suited for activity recognition, which allows us to recognize and interpret observations on-the-fly, and eventually to reason about human activities and intentions.

The methods and components presented in this thesis have been fully implemented and con-

stitute to a running system for markerless observation and interpretation of human activities.

## 7.2 Open Problems and the Challenge Ahead

We believe that the ideas and methods presented in this thesis contribute to the ambitious goal of understanding human actions and intentions by observing their motions and interactions within the environment. Still, work remains to be done until methods and applications will reach a level of maturity that will allow them to become an integral part of our everyday lives. One of the next milestones along this way will be the reliable long-term observation of daily routines of humans.

Among the open problems that need to be addressed is the automatic initialization and also the failure recovery of tracking systems. In our work we have assumed that both the shape parameters of the human model and the initial pose have been provided manually. This is a viable approach when video sequences are first recorded and then processed afterwards. Online approaches that need to observe humans in realtime will require an automated process for the initialization. Unfortunately, shape and pose initialization has received little attention in the scientific community [57]. A solution to this problem could come in two steps. First, a robust classifier could be trained to detect distinct keyposes that are well-suited for initialization. Second, knowledge about the detected keyposes could be used to provide a rough initialization that is then used as a starting point for optimization. When both shape and pose parameters need to be initialized, the estimation could be done in alternating steps, *e.g.* using the *expectation-maximization* (EM) algorithm [45]. Note that we learned a statistical set of shape parameters for our human model in Section 3.2 that is well-suited for automated parameter estimation.

Real-time performance is another important aspect when considering interactive scenarios, *e.g.* in *human-robot-interaction* (HRI). Our tracking system takes about $10$ sec per frame when using the general motion model and $2.5$ sec per frame when the observed motions are known to our self-trained motion model. A speed up of around two orders of magnitude is required to achieve interactive framerates. Given that the aforementioned framerates have been calculated on single-threaded and mostly unoptimized code on a standard consumer PC, we believe that near real-time performance of our approach is within reach. The largest potential for optimization is in the parallelization of the particle evaluations that require more than $95\%$ of the overall processing time. A thorough GPU (*graphics processing unit*) implementation could help to boost performance, as particle filters are almost ideally suited for parallelization.

We also plan to adapt our method to work from a single viewpoint. This is a critical aspect

in many application scenarios, *e.g.* for autonomous robots interacting with humans, or for advanced input methods in *human-computer-interaction* (HCI). To account for the loss of 3D information when omitting the use of multiple calibrated cameras, we will focus on emerging sensor technologies such as *time-of-flight* cameras [54] or *structured light stereo* [35] that are able to provide dense depth maps from a single viewing direction. We believe that these technologies will also help to overcome the limitations of current foreground segmentation methods, *i.e.* the sensitivity to changing lighting conditions and the inability to extract foreground objects that are similar in color to the background.

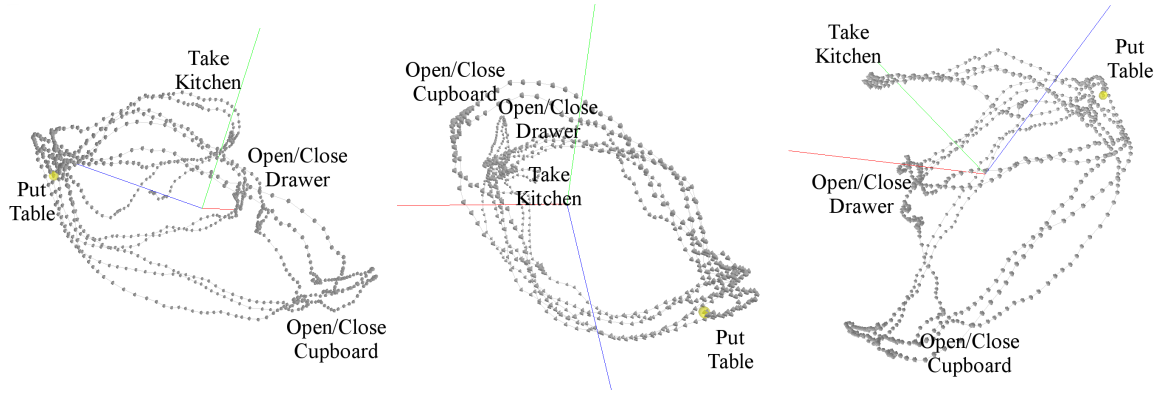


FIGURE 7.1 Three-dimensional embedding of spatio-temporal motion snippets using *Gaussian process dynamical models* (GPDM) [163]. The motion snippets have been calculated from sequence 1-0 of the TUM KITCHEN Data Set according to the formulation in Section 6.1.1, except that absolute world coordinates have been used. The embedding is shown from three different viewpoints. This example shows how low-dimensional embeddings of pose manifolds can be used to reveal the global structure in the data, as indicated by the activity labels. Images courtesy of Dominik Jain, TUM.

Opportunities for further research are also given with respect to the cognitive components of our work. So far we have successfully shown how to perform online activity recognition based purely on observed motions. Incorporating complementary sensor input will help to refine the recognition of actions by putting human motion patterns into context with object detections (*e.g.* through RFID readers) and the state of the environment (*e.g.* by detecting opened cupboards through magnetic sensors). In addition, graphical models such as *hidden markov models* (HMM) of *conditional random fields* (CRF) can be used to find the most probable sequence of semantic labels taking into account the detected motion correspondences. The transition probabilities needed *e.g.* for HMMs can be easily extracted from our *spatio-temporal neighborhood graphs*. Another interesting extension to our work is the structural analysis of human motion data. Revealing the underlying structure of pose manifolds could be used for unsupervised learning of activities. In Figure 7.1 we show a low-dimensional

embedding from one of the table-setting sequences in the TUM KITCHEN Data Set. The embedding was created using *Gaussian process dynamical models* (GPDM) [163] and the motion snippet representation of a temporal sequence of human poses as introduced in Section 6.1.1. Such embeddings provide a global view that can be used to detect structural similarities between activities or to find a generalized and compact representation of the underlying motion capture data.

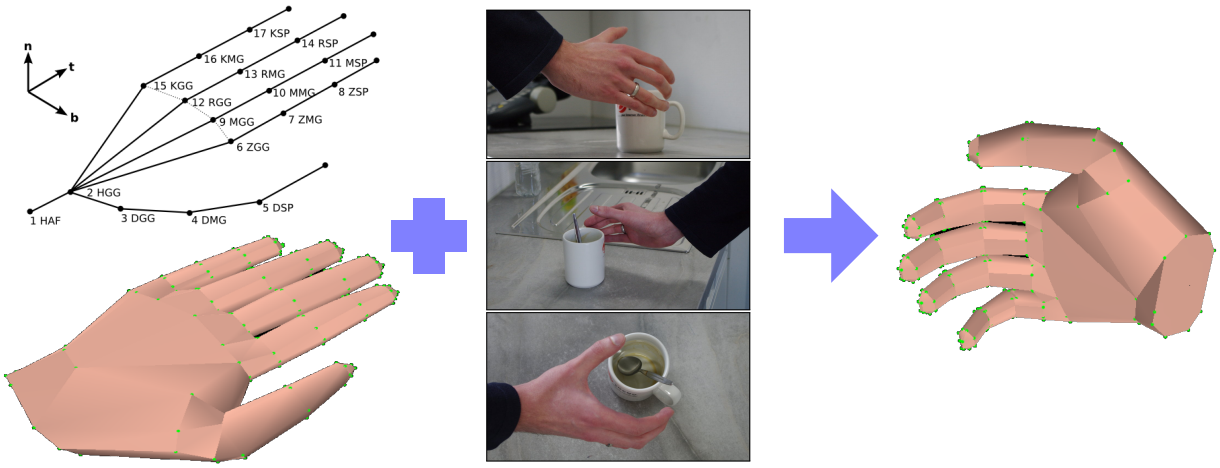


FIGURE 7.2 Extending the application domain to articulated hand tracking. The kinematic structure of human hands (28 d.o.f.) is comparable to that of the human body, and extending the methods presented in this thesis should be straightforward. Hand tracking has promising applications *e.g.* in *humanoid robotics*, where it could be used to attain dexterous manipulation through demonstration. The hand model shown conforms to the specifications of the RAMSIS model as presented in Chapter 3.

Finally, the methods presented in this thesis are applicable in a wide range of related application scenarios. One example is articulated hand tracking [144, 29], which has promising applications *e.g.* for imitation learning in *humanoid robotics* to provide robots with dexterous manipulation skills. As depicted in Figure 7.2, the problem of tracking articulated poses of the human hand resembles the problem of human fullbody motion tracking. The tree-like kinematic structure of human hands is comparable to that of the human body, and we believe that our tracking methods will translate well to this and other related applications.

# Bibliography


[1] Karim Abdel-Malek, Jingzhou Yang, Joo H. Kim, Timothy Marler, Steven Beck, Colby Swan, Laura Frey-Law, Anith Mathai, Chris Murphy, Salam Rahmatallah, and Jasbir Arora. Development of the virtual-human santos$^{\text{tm}}$. In *First International Conference on Digital Human Modeling (ICDHM)*, 2007.

[2] Ankur Agarwal and Bill Triggs. 3d human pose from silhouettes by relevance vector regression. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2004.

[3] Ankur Agarwal and Bill Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1), jan 2006.

[4] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1):5–43, January 2003.

[5] Dragomir Anguelov. *Learning Models of Shape from 3D Range Data*. PhD thesis, Stanford University, California, 2005.

[6] Dragomir Anguelov, Daphne Koller, Hoi-Cheung Pang, Praveen Srinivasan, and Sebastian Thrun. Recovering articulated object models from 3d range data. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 18–26, Arlington, Virginia, United States, 2004. AUAI Press.

[7] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005.

[8] Okan Arikan and D. A. Forsyth. Interactive motion generation from examples. In *29th annual conference on computer graphics and interactive techniques (SIGGRAPH 2002)*, 2002.

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

[10] P. Azad, T. Asfour, and R. Dillmann. Toward an Unified Representation for Imitation of Human Motion on Humanoids. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[11] Pedram Azad, Ales Ude, Tamim Asfour, and Rüdiger Dillmann. Stereo-based markerless human motion capture for humanoid robot systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3951–3956. IEEE, 2007.

[12] Andreas Baak, Meinard Müller, Bodo Rosenhahn, and Hans-Peter Seidel. Stabilizing motion tracking using retrieved motion priors. In Roberto Cipolla, Martial Hebert, Xiaoou Tang, and Naokazu Yokoya, editors, *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, pages 1428–1435, Kyoto, Japan, 2009. IEEE.

[13] A. O. Balan and M. J. Black. The naked truth: Estimating body shape under clothing. In *European Conference on Computer Vision, ECCV*, volume 5303, pages 15–29, 2008.

[14] A. O. Balan, L. Sigal, and M. J. Black. A quantitative evaluation of video-based 3d person tracking. In *ICCCN '05: Proceedings of the 14th International Conference on Computer Communications and Networks*, pages 349–356, Washington, DC, USA, 2005. IEEE Computer Society.

[15] Alexandru O. Balan and Michael J. Black. An adaptive appearance model approach for model-based articulated object tracking. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 758–765, Washington, DC, USA, 2006. IEEE Computer Society.

[16] Jan Bandouch and Michael Beetz. Tracking humans interacting with the environment using efficient hierarchical sampling and layered observation models. In *IEEE Int. Workshop on Human-Computer Interaction (HCI). In conjunction with ICCV2009*, 2009.

[17] Jan Bandouch, Florian Engstler, and Michael Beetz. Accurate human motion capture using an ergonomics-based anthropometric human model. In *Proceedings of the Fifth International Conference on Articulated Motion and Deformable Objects (AMDO)*, 2008.

[18] Jan Bandouch, Florian Engstler, and Michael Beetz. Evaluation of hierarchical sampling strategies in 3d human pose estimation. In *Proceedings of the 19th British Machine Vision Conference (BMVC)*, 2008.

[19] Michael Beetz, Jan Bandouch, Suat Gedikli, Nico von Hoyningen-Huene, Bernhard Kirchlechner, and Alexis Maldonado. Camera-based observation of football games for analyzing multi-agent activities. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2006.

[20] Michael Beetz, Jan Bandouch, Dominik Jain, and Moritz Tenorth. Towards Automated Models of Activities of Daily Life. In *First International Symposium on Quality of Life Technology - Intelligent Systems for Better Living*, Pittsburgh, Pennsylvania USA, 2009.

[21] Michael Beetz, Jan Bandouch, Alexandra Kirsch, Alexis Maldonado, Armin Müller, and Radu Bogdan Rusu. The assistive kitchen — a demonstration scenario for cognitive technical systems. In *Proceedings of the 4th COE Workshop on Human Adaptive Mechatronics (HAM)*, 2007.

[22] Michael Beetz, Suat Gedikli, Jan Bandouch, Bernhard Kirchlechner, Nico von Hoyningen-Huene, and Alexander Perzylo. Visually tracking football games based on tv broadcasts. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[23] Michael Beetz, Freek Stulp, Bernd Radig, Jan Bandouch, Nico Blodow, Mihai Dolha, Andreas Fedrizzi, Dominik Jain, Uli Klank, Ingo Kresse, Alexis Maldonado, Zoltan Marton, Lorenz Mösenlechner, Federico Ruiz, Radu Bogdan Rusu, and Moritz Tenorth. The assistive kitchen — a demonstration scenario for cognitive technical systems. In *IEEE 17th International Symposium on Robot and Human Interactive Communication (RO-MAN), Muenchen, Germany*, 2008. Invited paper.

[24] Michael Beetz, Nicolai von Hoyningen-Huene, Bernhard Kirchlechner, Suat Gedikli, Francisco Siles, Murat Durus, and Martin Lames. ASpoGAMo: Automated Sports Game Analysis Models. *International Journal of Computer Science in Sport*, 8(1), 2009.

[25] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402, 2005.

[26] Liefeng Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas. Fast algorithms for large scale conditional 3d prediction. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[27] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[28] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision*, 2009.

[29] M. Bray, E. Koller-Meier, and L. Van Gool. Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding (CVIU)*, 106(1):116–129, 2007.

[30] Christoph Bregler. Learning and recognizing human dynamics in video sequences. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0, 1997.

[31] Christoph Bregler, Jitendra Malik, and Katherine Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, February 2004.

[32] 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 A. Leonardis, H. Bischof, and A. Pinz, editors, *European Conference on Computer Vision (ECCV)*, volume 3952 of *LNCS*, pages 98–111, Graz, Austria, May 2006. Springer.

[33] H. Bubb. RAMSIS - a Measuring and CAD-Tool, Serving as a Standard for Ergonomic Assessments of Workplaces, Cars and Other Products. In *Proceedings of the 13th Triennial Congress of the International Ergonomics Association*, Helsinki, 1997.

[34] H. Bubb, F. Engstler, F. Fritzsche, Ch. Mergl, O. Sabbah, P. Schaefer, and I. Zacher. The development of RAMSIS in past and future as an example for the cooperation between industry and university. *International Journal of Human Factors Modelling and Simulation*, 1(1):140–157, 2006.

[35] Lam Quang Bui and Sukhan Lee. Light pattern blur estimation for automatic projector focus control of structured light 3d camera. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2009.

[36] Fabrice Caillette, Aphrodite Galata, and Toby Howard. Real-time 3-d human body tracking using learnt models of behaviour. *Computer Vision and Image Understanding*, 109(2):112–125, 2008.

[37] PhaseSpace Motion Capture. http://www.phasespace.com/.

[38] Tat-Jen Cham and James M. Rehg. A multiple hypothesis approach to figure tracking. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1999.

[39] Kong Man Cheung, Simon Baker, and Takeo Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.

[40] Ascension Technology Corporation. http://www.ascension-tech.com.

[41] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[42] CMU Motion Capture Database. http://mocap.cs.cmu.edu.

[43] MPI HDM05 Motion Capture Database. http://www.mpi-inf.mpg.de/resources/HDM05/.

[44] Ankur Datta, Yaser Ajmal Sheikh, and Takeo Kanade. Modeling the product manifold of posture and motion. In *IEEE Int. Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS). In conjunction with ICCV2009*, 2009.

[45] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.

[46] Jonathan Deutscher and Ian Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision (IJCV)*, 61(2):185–205, 2005.

[47] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[48] Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003)*, 2003.

[49] Ahmed M. Elgammal, David Harwood, and Larry S. Davis. Non-parametric model for background subtraction. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 751–767, London, UK, 2000. Springer-Verlag.

[50] Ahmed M. Elgammal and Chan-Su Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[51] Florian Engstler, Jan Bandouch, and Heiner Bubb. Memoman - model based markerless capturing of human motion. In *The 17th World Congress on Ergonomics (International Ergonomics Association, IEA)*, Beijing, China, 2009.

[52] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, 2005.

[53] Dieter Fox. Adapting the sample size in particle filters through kld-sampling. *International Journal of Robotics Research*, 22, 2003.

[54] Stefan Fuchs and Gerd Hirzinger. Extrinsic and depth calibration of tof-cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008.

[55] J. Gall, B. Rosenhahn, and H.P. Seidel. Drift-free tracking of rigid and articulated objects. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[56] Jürgen Gall, Jürgen Potthoff, Christoph Schnörr, Bodo Rosenhahn, and Hans-Peter Seidel. Interacting and annealing particle filters: Mathematics and a recipe for applications. *Journal of Mathematical Imaging and Vision*, 28(1):1–18, 2007.

[57] Jürgen Gall, Bodo Rosenhahn, and Hans-Peter Seidel. Clustered stochastic optimization for object recognition and pose estimation. In Leo Hamprecht, Christoph Schnörr, and Bernd Jähne, editors, *29th Annual Symposium of the German Association for Pattern Recognition (DAGM'07)*, volume 4713 of *Lecture Notes in Computer Science*, pages 32–41, Heidelberg, Germany, 2007. Springer.

[58] Jürgen Gall, Carsten Stoll, Edilson de Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans-Peter Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, pages 1–8, Miami, USA, 2009. IEEE Computer Society.

[59] D. M. Gavrila. The visual analysis of human movement: a survey. *Computer Vision and Image Understanding (CVIU)*, 73(1):82–98, 1999.

[60] Suat Gedikli, Jan Bandouch, Nico von Hoyningen-Huene, Bernhard Kirchlechner, and Michael Beetz. An adaptive vision system for tracking soccer players from variable camera settings. In *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS)*, 2007.

[61] Human Solutions GmbH. http://www.human-solutions.com.

[62] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.

[63] Kristen Grauman, Gregory Shakhnarovich, and Trevor Darrell. Inferring 3d structure with a statistical image-based shape model. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 641, Washington, DC, USA, 2003. IEEE Computer Society.

[64] Daniel Grest and Volker Krüger. Gradient-enhanced particle filter for vision-based motion capture. In Ahmed M. Elgammal, Bodo Rosenhahn, and Reinhard Klette, editors, *Workshop on Human Motion*, volume 4814 of *Lecture Notes in Computer Science*, pages 28–41. Springer, 2007.

[65] Daniel Grest, Jan Woetzel, and Reinhard Koch. Nonlinear body pose estimation from depth images. In *DAGM-Symposium*, 2005.

[66] P. Guan, A. Weiss, A. O. Balan, and M. J. Black. Estimating human shape and pose from a single image. In *IEEE International Conference on Computer Vision, ICCV*, 2009.

[67] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W4s: A real-time system detecting and tracking people in 2 1/2d. In *5th European Conference on Computer Vision (ECCV)*, pages 877–892, London, UK, 1998. Springer-Verlag.

[68] Radu P. Horaud, Matti Niskanen, Guillaume Dewaele, and Edmond Boyer. Human motion tracking by registering an articulated surface to 3-d points and normals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[69] Nicholas R. Howe, Michael E. Leventon, and William T. Freeman. Bayesian reconstruction of 3D human motion from single-camera video. In *Advances in Neural Information Processing Systems (NIPS) 12*, pages 820–826, Denver, CO, November 2000. MIT Press.

[70] W. Hu, T. N. Tan, L. Wang, and S. J. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics - Part C*, 34(3):334–352, August 2004.

[71] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[72] Špela Ivekovič, Emanuele Trucco, and Yvan R. Petillot. Human body pose estimation with particle swarm optimisation. *Evolutionary Computation*, 16(4), 2008.

[73] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970.

[74] Odest Chadwicke Jenkins and Maja J Matarić. A Spatio-temporal Extension to Isomap Nonlinear Dimension Reduction. In *The International Conference on Machine Learning (ICML 2004)*, pages 441–448, Banff, Alberta, Canada, Jul 2004.

[75] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. In *International Conference on Automatic Face and Gesture Recognition*, pages 38–44, Killington, Vermont, 1996.

[76] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.

[77] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[78] Ibrahim A. Kapandji. *Rumpf und Wirbelsäule*, volume 3 of *Funktionelle Anatomie der Gelenke*. Hippokrates, 3rd edition, 1999.

[79] Roland Kehl and Luc Van Gool. Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding (CVIU)*, 104(2):190–209, 2006.

[80] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, and Larry S. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.

[81] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[82] Steffen Knoop, Stefan Vacek, and Rüdiger Dillmann. Fusion of 2d and 3d sensor data for articulated body tracking. *Robotics and Autonomous Systems*, 57(3):321–329, 2009.

[83] David Knossow, Remi Ronfard, and Radu P. Horaud. Human motion tracking with a kinematic parameterization of extremal contours. *International Journal of Computer Vision*, 2008.

[84] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion Graphs. In *29th annual conference on computer graphics and interactive techniques (SIGGRAPH 2002)*, 2002.

[85] Volker Krüger and Daniel Grest. Using hidden markov models for recognizing action primitives in complex actions. In *SCIA*, volume 4522 of *Lecture Notes in Computer Science*, pages 203–212. Springer, 2007.

[86] Volker Krüger, Danica Kragic, Ales Ude, and Christopher Geib. The meaning of action: a review on action recognition and mapping. *Advanced Robotics*, 21(13):1473–1501, 2007.

[87] James J. Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[88] Dana Kulić, Wataru Takano, and Yoshihiko Nakamura. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *International Journal of Robotics Research*, 27(7):761–784, 2008.

[89] A. Ladikos, S. Benhimane, and N. Navab. Efficient visual hull computation for real-time 3d reconstruction using cuda. In *Proceedings of the 2008 Conference on Computer Vision and Pattern Recognition Workshops*, 2008.

[90] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. In *29th annual conference on computer graphics and interactive techniques (SIGGRAPH 2002)*, 2002.

[91] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[92] Fengjun Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[93] John MacCormick and Andrew Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1):57–71, 2000.

[94] John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 3–19, London, UK, 2000. Springer-Verlag.

[95] Autodesk Maya. http://usa.autodesk.com/adsk/servlet/pc/index?id=13577897&siteID=123112.

[96] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.

[97] Ivana Mikic, Mohan Trivedi, Edward Hunter, and Pamela Cosman. Articulated body posture estimation from multi-camera voxel data. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[98] David Minnen, Irfan Essa, and Thad Starner. Expectation grammars: Leveraging high-level expectations for activity recognition. *CVPR*, 02:626, 2003.

[99] J.R. Mitchelson and A. Hilton. Simultaneous pose estimation of multiple people using multiple-view cues with hierarchical sampling. In *British Machine Vision Conference (BMVC)*, 2003.

[100] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding (CVIU)*, 81(3):231–268, 2001.

[101] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding (CVIU)*, 104(2):90–126, 2006.

[102] L. Muendermann, S. Corazza, and T.P. Andriacchi. Accurately measuring human movement using articulated icp with soft-joint constraints and a repository of articulated models. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007.

[103] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pages 331–340. INSTICC Press, 2009.

[104] Microsoft Project Natal. http://www.xbox.com/live/projectnatal.

[105] Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[106] Abhijit Ogale, Alap Karapurkar, and Yiannis Aloimonos. View-invariant modeling and recognition of human actions using grammars. In *IEEE Workshop on Dynamical Vision*, 2005.

[107] A. Oikonomopoulos, P. Ioannis, and M. Pantic. Spatio-Temporal Salient Points for Visual Recognition of Human Actions. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 36(3):710–719, 2006.

[108] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[109] S. Pellegrini, K. Schindler, and D. Nardi. A generalization of the ICP algorithm for articulated bodies. In *British Machine Vision Conference (BMVC)*, 2008.

[110] Patrick Peursum, Svetha Venkatesh, and Geoff A. W. West. Tracking-as-recognition for articulated full-body human motion analysis. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007.

[111] Michael K Pitt and Neil Shephard. Filtering via simulation: auxiliary particle filter. *Journal of the American Statistical Association*, 94:590–599, 1999.

[112] Rolf Plänkers and Pascal Fua. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding (CVIU)*, 81(3):285–302, March 2001.

[113] The Player/Stage/Gazebo project. http://playerstage.sourceforge.net/.

[114] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding (CVIU)*, 108(1-2):4–18, 2007.

[115] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 2010.

[116] Smith Micro Software Poser. http://poser.smithmicro.com.

[117] William H. Press. *Numerical recipes : the art of scientific computing*. Cambridge University Press, 3rd edition, September 2007.

[118] EC Funded CAVIAR project/IST 2001 37540. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[119] Deva Ramanan, , Deva Ramanan, and D. A. Forsyth. Automatic annotation of everyday movements. In *NIPS*. MIT Press, 2003.

[120] Deva Ramanan, David A. Forsyth, and Andrew Zisserman. Tracking people by learning their appearance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):65–81, 2007.

[121] Bodo Rosenhahn, Thomas Brox, Uwe Kersting, Andrew Smith, Jason Gurney, and Reinhard Klette. A system for marker-less motion capture. *Künstliche Intelligenz*, 20(1):45–51, January 2006.

[122] Bodo Rosenhahn, Uwe Kersting, Katie Powell, Reinhard Klette, Gisela Klette, and Hans-Peter Seidel. A system for articulated tracking incorporating a clothing model. *Machine Vision and Applications*, 18(1):25–40, 2007.

[123] Bodo Rosenhahn, Christian Schmaltz, Thomas Brox, Joachim Weickert, and Hans-Peter Seidel. Staying well grounded in markerless motion capture. In *Proceedings of the 30th DAGM Symposium*, München, Germany, 2008.

[124] D.M. Russell and S.G. Gong. Minimum cuts of a time-varying background. In *British Machine Vision Conference (BMVC)*, page II:809, 2006.

[125] Radu Bogdan Rusu, Jan Bandouch, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Action Recognition in Intelligent Environments using Point Cloud Features Extracted from Silhouette Sequences. In *IEEE 17th International Symposium on Robot and Human Interactive Communication (RO-MAN), Muenchen, Germany*, 2008.

[126] Radu Bogdan Rusu, Jan Bandouch, Franziska Meier, Irfan Essa, and Michael Beetz. Human Action Recognition using Global Point Feature Histograms and Action Shapes. *Advanced Robotics journal, Robotics Society of Japan (RSJ)*, 2009.

[127] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Emanuel Dolha, and Michael Beetz. Functional Object Mapping of Kitchen Environments. In *Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, September 22-26*, 2008.

[128] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *17th International Conference on Pattern Recognition (ICPR 2004)*, volume 3, pages 32–36, Aug. 2004.

[129] A. Seidl. *Das Menschmodell RAMSIS - Analyse, Synthese und Simulation dreidimensionaler Körperhaltungen des Menschen*. PhD thesis, Technische Universität München, 1994.

[130] T. Seitz. The optical measurement system PCMAN. In A. Coblentz, editor, *Proceedings of the Workshop on 3D Anthropometry and Products Design*, Paris, June 1998.

[131] T. Seitz. *Videobasierte Messung menschlicher Bewegungen konform zum Menschmodell RAMSIS*. PhD thesis, Technische Universität München, 2003.

[132] CMU Kitchen Data Set. http://kitchen.cs.cmu.edu.

[133] Yaser Sheikh, Mumtaz Sheikh, and Mubarak Shah. Exploring the space of a human action. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 144–149. IEEE Computer Society, 2005.

[134] Ken Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH*, volume 19, 1985.

[135] Hedvig Sidenbladh, Michael J. Black, and Leonid Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *European Conference on Computer Vision (ECCV)*, pages 784–800, 2002.

[136] Leonid Sigal and Michael J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical report, Brown University, 2006.

[137] Leonid Sigal and Michael J. Black. Predicting 3D people from 2D pictures. In *Proceedings of the International Conference on Articulated Motion and Deformable Objects (AMDO'06)*, number 4069 in Lecture Notes in Computer Science, pages 185–195, Port d'Andratx, Spain, July 2006. Springer-Verlag.

[138] Gregory G. Slabaugh. Computing euler angles from a rotation matrix. Technical report, Georgia Institute of Technology, 1999.

[139] C. Sminchisescu and A. Telea. Human pose estimation from silhouettes - a consistent approach using distance level sets. In *In WSCG International Conference on Computer Graphics, Visualization and Computer Vision*, 2002.

[140] Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, and Dimitris Metaxas. Conditional random fields for contextual human motion recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1808–1815. IEEE Computer Society, 2005.

[141] Cristian Sminchisescu and Bill Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotic Research*, 22(6):371–392, June 2003.

[142] Inition Organic Motion Stage. http://www.inition.co.uk.

[143] Chris Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:2246, 1999.

[144] B. Stenger, P. R. S. Mendonca, and R. Cipolla. Model-based hand tracking using an unscented kalman filter. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2001.

[145] Freek Stulp, Andreas Fedrizzi, Franziska Zacharias, Moritz Tenorth, Jan Bandouch, and Michael Beetz. Combining analysis, imitation, and experience-based learning to acquire a concept of reachability. In *9th IEEE-RAS International Conference on Humanoid Robots*, 2009.

[146] Phoenix Technologies Incorporated (PTI) 3D Motion Capture Systems. http://www.ptiphoenix.com.

[147] Vicon Motion Systems. http://www.vicon.com.

[148] Graham W. Taylor, Geoffrey E. Hinton, and Sam T. Roweis. Modeling human motion using binary latent variables. In *Proc. of the 20th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1345–1352. MIT Press, 2006.

[149] J. B. Tenenbaum, V. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[150] Moritz Tenorth, Jan Bandouch, and Michael Beetz. The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In *IEEE Int. Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS). In conjunction with ICCV2009*, 2009.

[151] Moritz Tenorth and Michael Beetz. KnowRob — Knowledge Processing for Autonomous Personal Robots. In *IEEE/RSJ International Conference on Intelligent RObots and Systems.*, 2009.

[152] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, Cambridge, 2005.

[153] Xsens 3D Motion Tracking. http://www.xsens.com.

[154] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrol ogy using off-the-shelf tv cameras and lenses. pages 221–244, 1987.

[155] N.G. Tsakarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, L. Righetti, J. Santos-Victor, A.J. Ijspeert, M.C. Carrozza, and D.G. Caldwell. iCub - The Design and Realization of an Open Humanoid Platform for Cognitive and Neuroscience Research. *Journal of Advanced Robotics, Special Issue on Robotic platforms for Research in Neuroscience*, 21(10):1151–1175, 2007.

[156] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[157] R. Urtasun, D. Fleet, and P. Fua. 3D People Tracking with Gaussian Process Dynamical Models. In *Conference on Computer Vision and Pattern Recognition*, pages 238–245, 2006.

[158] Raquel Urtasun and Pascal Fua. 3d human body tracking using deterministic temporal motion models. In *European Conference on Computer Vision (ECCV)*, pages 92–106, 2004.

[159] Rudolph van der Merwe, Arnaud Doucet, Nando de Freitas, and Eric A. Wan. The unscented particle filter. In *Neural Information Processing Systems (NIPS)*, 2000.

[160] Ashok Veeraraghavan, Amit K. Roy-Chowdhury, and Rama Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1896–1909, 2005.

[161] M. Vondrak, L. Sigal, and O.C. Jenkins. Physical simulation for probabilistic motion tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[162] S. Wachter and H.-H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3):174–192, 1999.

[163] J. Wang, D. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models. *Advances in Neural Information Processing Systems*, 18:1441–1448, 2006.

[164] Liang Wang and David Suter. Informative shape representations for human action recognition. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 1266–1269. IEEE Computer Society, 2006.

[165] Ping Wang and James M. Rehg. A modular approach to the analysis and evaluation of particle filters for figure tracking. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 790–797, Washington, DC, USA, 2006. IEEE Computer Society.

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

[167] Augustus A. White and Manohar M. Panjabi. *Clinical Biomechanics of the Spine*. Lippincott Williams and Wilkins, 2nd edition, 1990.

[168] Christopher R. Wren, Ali J. Azarbayejani, Trevor Darrell, and Alex P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):780–785, July 1997.

[169] Katsu Yamane, Yoshifumi Yamaguchi, and Yoshihiko Nakamura. Human motion database with a binary tree and node transition graphs. In *Robotics: Science and Systems V*, 2009.

[170] A. Yilmaz and Mubarak Shah. Actions sketch: A novel action representation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 984–989, 2005.

[171] yWorks yEd Graph Editor. http://www.yworks.com/en/products_yed_about.html.

[172] G.F. Zhang, J. Jia, W. Xiong, T.T. Wong, P.A. Heng, and H.J. Bao. Moving object extraction with a hand-held camera. In *ICCV*, 2007.

[173] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

[174] Dongfang Zhao and Li Yang. Incremental isometric embedding of high-dimensional data using connected neighborhood graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):86–98, 2009.