Human Pose Estimation from Video and Inertial Sensors

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

Doktor-Ingenieur

(abgekürzt: Dr.-Ing.)
genehmigte

Dissertation

von

Gerard Pons Moll

geboren am 25. Oktober 1984 in Barcelona.

2014

Doktorvater/Supervisor:
Prof. Dr.-Ing. Bodo Rosenhahn
Gottfried Wilhelm Leibniz Universität Hannover, Germany

Gutachter/Reviewers:
Prof. Dr.-Ing. Bodo Rosenhahn
Gottfried Wilhelm Leibniz Universität Hannover, Germany

Prof. Dr. Michael Black
Max Planck for Intelligent Systems, Tübingen, Germany

Prof. Dr.-Ing. Jörn Ostermann
Gottfried Wilhelm Leibniz Universität Hannover, Germany

Datum des Kolloquiums / Date of Defense:
25.02.2014

Autor/Author:
Gerard Pons-Moll

Acknowledgments

I would like to give special thanks to my doctoral advisor Prof. Dr.-Ing Bodo Rosenhahn for many interesting discussions and for his efforts to make me grow as a researcher. I would also like to thank Prof. Michael Black and Prof. Dr.-Ing Jörn Ostermann for being part of the defense tribunal.
I am very grateful to Prof. David Fleet for many interesting discussions and guidance during my stay at University of Toronto. I am also thankful to all the people with whom I collaborated during my Ph.D: Laura Leal-Taixe, Andreas Baak, Thomas Helten Meinard Müller, Hans-Peter Seidel, Jürgen Gall, Alina Kuznetsova, Jonathan Taylor, Jamie Shotton, Aaron Hertzmann and Andrew Fitzgibbon. I would also like to thank all my colleagues at the TNT group for all the help and support I received and special thanks to Stephan Preihs for being such a great office mate.
Finally, I would like to thank my *family* for the unconditional support and for teaching me the most valuable thing I have ever learned: *to enjoy learning*.

To my family

Contents

1 Introduction **1**
1.1 Problem Statement 1
1.2 Motivation 1
1.3 Motion Capture from Image and Inertial Sensors 2
1.4 Challenges 3
1.5 Generative versus Discriminative 5
1.6 Related Work 5
  1.6.1 Image based Pose Estimation 5
  1.6.2 Inertial Sensor Trackers 6
  1.6.3 Representations of Pose 6
1.7 Contributions 7
1.8 Overview 11
1.9 List of Published Papers 11

2 Generative Pose Estimation **15**
2.1 Kinematic Parametrization 15
  2.1.1 Rotation Matrices 15
  2.1.2 Euler Angles 17
  2.1.3 Quaternions 17
  2.1.4 Axis-angle 18
  2.1.5 Rotation Derivatives and Rigid Body Velocity 21
  2.1.6 A Note on Linearization of Rotational Motion 22
  2.1.7 Metrics on SO(3) 23
  2.1.8 Kinematic Chains 24
  2.1.9 Human Pose Parametrization 25
2.2 Other Pose Representations 26
  2.2.1 Part based 27
  2.2.2 Joint Positions 27
  2.2.3 Posebits 27
2.3 Model Creation 28
  2.3.1 Geometric Primitives 28
  2.3.2 Detailed Body Scans 28
  2.3.3 Detailed Shape from Images 30
2.4 Likelihood function 30
  2.4.1 Image based cues 31
  2.4.2 Inertial Sensor based cues 36
2.5 Multiple cues 36
2.6 Optimization 37
  2.6.1 Local Optimization 37
  2.6.2 Particle-based Optimization and Filtering 43
2.7 Prior Knowledge 48
  2.7.1 Joint Limits 48
  2.7.2 Collision Avoidance 48
  2.7.3 Motion Models 48

2.8 Limitations of Pure Image Based Methods 49

3 Inertial Sensors and Camera Calibration 51
3.1 Camera Model 51
3.1.1 Internal Camera Matrix 51
3.1.2 Extrinsics Camera Matrix 52
3.1.3 Camera Calibration 52
3.2 IMUs 53
3.2.1 Sensor Data 53
3.2.2 Limitations of IMUs 53
3.3 Calibration and Synchronization with the Video Cameras 54

4 Local Hybrid Tracker 57
4.1 Introduction 57
4.1.1 Pose Parameterization 58
4.2 Video-based Tracker 59
4.3 Hybrid Tracker 60
4.4 Integration of Sensor Data 61
4.4.1 Sensor Data 61
4.4.2 Integration of Orientation Data into the Video-based Tracker 61
4.4.3 A Geometric Derivation 63
4.5 Experiments 65
4.6 Summary 66

5 Inverse Kinematics Sampler 71
5.1 Introduction 71
5.2 Related Work 73
5.3 Global Optimization with Sensors 73
5.3.1 Calibration and Synchronization 73
5.3.2 Human Motion Model 74
5.3.3 Optimization Procedure 75
5.4 Manifold Sampling 75
5.4.1 Inverse Kinematics using Inertial Sensors 76
5.4.2 Paden-Kahan subproblems 78
5.4.3 Sensor Noise Model 81
5.4.4 Implementation Details 82
5.5 Experiments 85
5.6 Discussion and limitations 90
5.7 Summary 91

6 Posebits for Monocular Pose Estimation 93
6.1 Introduction 93
6.2 Related Work 95
6.3 Posebits 95
6.3.1 Automatic Posebit Generation 96
6.3.2 Learning 96
6.3.3 Single frame proposals 97
6.3.4 Posebits selection 98
6.4 3D Pose Estimation 102
6.5 Posebits database 103

6.6 Annotation with Amazon Mechanical Turk 104
6.7 Setup and Parameter Settings 104
    6.7.1 Learning of Posebits Classifier 104
    6.7.2 Posebit Selection 106
    6.7.3 3D Pose Estimation 106
6.8 Experiments 107
6.9 Discussion 110
6.10 Summary and Future Work 111

7 Conclusions 115
7.1 Local Optimization 115
7.2 Inverse Kinematics Sampling Tracker 116
7.3 Posebits Pose Estimator 117
7.4 Future Work 118
    7.4.1 Multiple Cues 118
    7.4.2 More Detail 118
    7.4.3 Multiple People and the Environment 118
    7.4.4 Monocular Pose Estimation 119
    7.4.5 Putting the Pieces Together 119

Bibliography 129

Abstract

The analysis and understanding of human movement is central to many applications such as sports science, medical diagnosis and movie production. The ability to automatically monitor human activity in security sensitive areas such as airports, lobbies or borders is of great practical importance. Furthermore, automatic pose estimation from images leverages the processing and understanding of massive digital libraries available on the Internet. We build upon a model based approach where the human shape is modeled with a surface mesh and the motion is parameterized by a kinematic chain. We then seek for the pose of the model that best explains the available observations coming from different sensors.

In a first scenario, we consider a calibrated multiview setup in an indoor studio. To obtain very accurate results, we propose a novel tracker that combines information coming from video and a small set of Inertial Measurement Units (IMUs). We do so by locally optimizing a joint energy consisting of a term that measures the likelihood of the video data and a term for the IMU data. This is the first work to successfully combine video and IMUs information for full body pose estimation. When compared to commercial marker based systems the proposed solution is more cost efficient and less intrusive for the user.

In a second scenario, we relax the assumption of an indoor studio and we tackle outdoor scenes with background clutter, illumination changes, large recording volumes and difficult motions of people interacting with objects. Again, we combine information from video and IMUs. Here we employ a particle based optimization approach that allows us to be more robust to tracking failures. To satisfy the orientation constraints imposed by the IMUs, we derive an analytic Inverse Kinematics (IK) procedure to sample from the manifold of valid poses. The generated hypothesis come from a lower dimensional manifold and therefore the computational cost can be reduced. Experiments on challenging sequences suggest the proposed tracker can be applied to capture in outdoor scenarios. Furthermore, the proposed IK sampling procedure can be used to integrate any kind of constraints derived from the environment.

Finally, we consider the most challenging possible scenario: pose estimation of monocular images. Here, we argue that estimating the pose to the degree of accuracy as in an engineered environment is too ambitious with the current technology. Therefore, we propose to extract meaningful semantic information about the pose directly from image features in a discriminative fashion. In particular, we introduce posebits which are semantic pose descriptors about the geometric relationships between parts in the body. The experiments show that the intermediate step of inferring posebits from images can improve pose estimation from monocular imagery. Furthermore, posebits can be very useful as input feature for many computer vision algorithms.

Keywords: human pose estimation, model based pose estimation, inertial sensors, posebits.

Zusammenfassung

Die Analyse und Interpretation menschlicher Bewegungsabläufe ist ein wichtiger Bestandteil vieler Anwendungen zum Beispiel in der Sportwissenschaft, der medizinischer Diagnosestellung und der Filmindustrie. Die Überwachung von sicherheitskritischen Bereichen wie Flughäfen, Lobbys oder Grenzübergängen ist ebenfalls von großer praktischer Relevanz. Auch macht es die automatische Bewegungsschätzung möglich riesige (personenbezogene) Datenmengen des Internets zu verarbeiten und zu verstehen. Die vorgestellten Verfahren wählen einen Modell-gestützten Zugang, bei dem die menschliche Gestalt mit einem Polygonnetz als Oberfläche und die Bewegung über eine kinematische Kette parametrisiert wird. Es wird diejenige Konfiguration gewählt, die am besten mit den beobachteten Sensordaten übereinstimmt.

Zu Beginn wird ein kalibriertes Mehrkamerasystem in einer Studioumgebung betrachtet. Hierfür wird ein Nachführsystem vorgestellt, welches auf hohe Präzision abzielt und die Informationen aus Video und Orientierungs- und Beschleunigungssensoren (Inertial Measurement Units, IMUs) zusammenführt. Die verwendete, lokal optimierte, Energiefunktion setzt sich jeweils aus einem Term für die Wahrscheinlichkeit einer Pose, passend zu den Videodaten, und einen Zweiten, für die IMU Daten, zusammen. Dies führt zur erfolgreichen Anwendung, die Video- und IMU-Daten für die Lageschätzung des gesamte Körpers kombiniert. Im Vergleich zu kommerziellen Marker-basierten Systemen ist das vorgestellte Verfahren kostengünstiger und weniger invasiv.

Im Anschluss werden Außenaufnahmen mit komplexen Hintergrund, Beleuchtungsveränderungen und anspruchsvollen menschlichen Bewegungen mit Objektinteraktion betrachtet und damit die Anforderung an die Umgebung des Systems gelockert. Wieder werden Videodaten und IMU-Daten gemeinsam genutzt. Ein Partikel-basiertes Optimierungsverfahren hilft hier die Robustheit gegenüber Nachführfehlern zu erhöhen. Die von den IMUs gestellten Orientierungsnebenbedingungen werden durch ein analytisches inverses Kinematik-Verfahren (IK) geloest, das zufällig Köperkonfigurationen aus der zugrundeliegenden Mannigfaltigkeit der gültigen Konfigurationen zieht. Die generierten Hyphothesen sind Elemente eines niedrigdimensionalen Hypothesenraums und reduzieren damit den notwendigen tatsächlichen Suchraum. Mehrere Experimente zeigen, dass das vorgestellte Nachführsystem zuverlässig und robust funktioniert. Desweiteren kann die vorgestellte IK-Methode dazu genutzt werden, jede Art von Nebenbedingungen der Umgebung zu berücksichtigen.

Schlussendlich betrachten wir den anspruchsvollen Fall der menschlichen Lageschätzung aus monokularen Bildern. Es ist anzunehmen, dass eine Genauigkeit, die mit einer Qualität in einer kontrollierten Umgebung vergleichbar wäre, mit den momentanen technischen Möglichkeiten ein zu ambitioniertes Ziel darstellt. Deshalb werden weitere aussagekräftige semantische Informationen über die Lage direkt aus dem Bild extrahiert. Es werden Posebits eingeführt, semantische Gestaltdeskriptoren, die das geometrische Verhältnis der Körperteile zueinander beschreiben. Die Experimente belegen, dass der vorgelagerte Schritt der Posebit-Schätzung aus Bildern die menschliche Lageschätzung von monokularen Bildern verbessert. Desweiteren stellen sich Posebits auch als ein sehr nützliches Eingabemerkmal für andere Bilderkennungsalgorithmen heraus.

Schlagwörter: Menschlichen Bewegungen, Model-Basiertes Pose-Schätzung, Beschleunigungssensoren, Posebits.

1 Introduction

1.1 Problem Statement

In this thesis we address the problem of estimating the *pose* of humans from images and inertial sensors. The pose of a human is determined by the location of the joints and bones in the body. The level of detail of the pose is application dependent: for applications such as sports science and medical studies a sub-centimeter accuracy is necessary but for applications such as scene understanding or action recognition a coarse representation of the pose might be enough. In this thesis, we introduce two solutions for pose estimation at different levels of detail. The first uses a multicamera setup and Inertial Measurement Units (IMUs) to obtain a very high degree of accuracy. The second solution extracts qualitative pose information from a single image. This qualitative representation of pose consists of bits with information about relative position of body parts, *e.g.*, the left leg is in front of the right leg, the left hand is above the head etc.

Commercial systems such as *Vicon*, *Simi*, *Qualisys* for (MoCap) are marker-based, which limit their applicability in outdoor scenarios where engineered rooms are not an option. Hence, many of approaches have been proposed to estimate the pose from images. Unfortunately, the accuracy of such approaches is still far from the accuracy achieved by commercial systems, specially in uncontrolled outdoor scenarios. This is mostly due to the image ambiguities present in outdoor environments. By contrast, we propose a hybrid tracking system that combines information from video and inertial sensors to obtain much more accurate results. To make the system as un-invasive as possible, we limit the number of IMU's to be used to five which are attached at the back and the lower extremities. We formulate several solutions to fuse image and orientation cues efficiently exploiting the advantages of each sensor type. The accuracy of our hybrid system is competitive with marker based systems even in outdoor environments where extracting reliable image cues is very challenging.
Unfortunately, for certain applications neither IMUs nor multiview images are available, for example in video surveillance, motion capture of live events or motion capture of video archives in video collections. In such scenarios the only available information comes from a single monocular image. However, in many cases the artificial intelligence system might to perceive only an abstract representation of the human pose to understand and interact with the physical world. Hence, we also propose an inference system that extracts semantic pose information directly from monocular images.

1.2 Motivation

Recognizing and understanding the pose of humans has many potential applications Fig. 1.1. For an artificial system interacting with humans it is essential to perceive and understand the human pose. Indeed, several studies show that at least 60% of communication comes from body language. A system capable to understand the human pose in real time enables man-machine interaction, *i.e.*, controlling and interacting with the artificial intelligence systems through gestures. In the movie and gaming industry Motion Capture (MoCap) data is used for character animation, see Fig. 1.2. Notable movies using cutting edge performance capture technology are *Avatar*, *The Adventures of*

---

[1] Image sources: (a) *Credit: James Martin/CNET Networks*, (b) *www.qualysis.com*, (c) *Microsoft Gaming Studios*

[2] Image sources: (a)(b) http://www.lordoftherings.com/

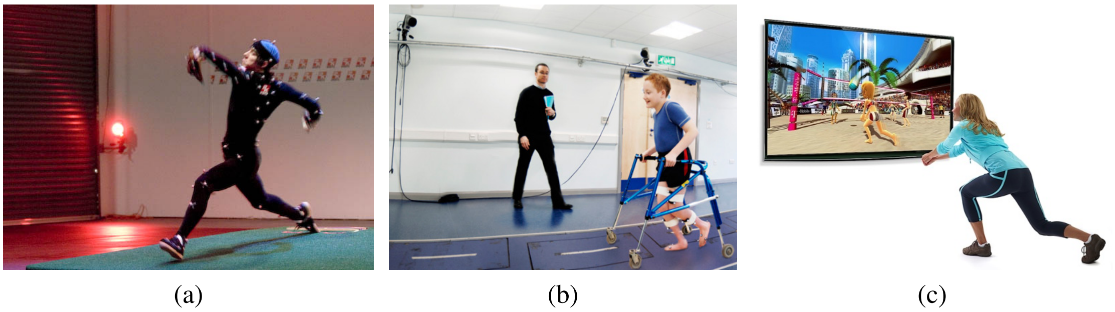

(a) (b) (c)

Figure 1.1: Applications: human motion capture has many potential applications in different fields such as (a) sports science, (b) medical diagnosis and rehabilitation, (c) gaming and man-machine interaction [1].

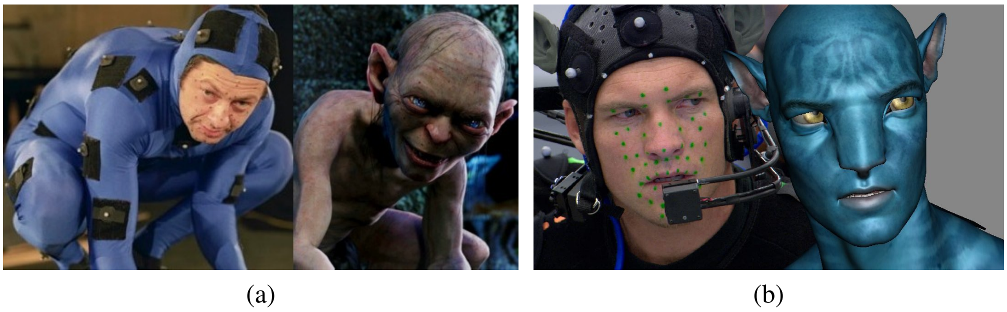

(a) (b)

Figure 1.2: Marker based Motion Capture: a series of markers are tracked by several cameras to reconstruct the 3D motion. We show making-off of the movies (a) The Lord of the Rings, and (b) Avatar. [2]

Tintin, or video games such as *L.A. Noir*. Commercial systems typically use marker-based systems. Human motion analysis also has numerous applications in the fields of sport science and medicine, see Fig. 1.1. The reconstruction of accurate 3D human motion helps clinicians diagnose potential injuries and helps them monitor recovery. Athletes also benefit from MoCap analysis to improve economy and performance.

1.3 Motion Capture from Image and Inertial Sensors

Current commercial MoCap systems are marker based such as *Vicon* [3], *Simi* [4] and *Qualisys* [5]. Typically, the subject has to wear a tight suit with retro-reflective markers that are tracked by several cameras, see Fig. 1.2. Given the 3D location of the markers, the human motion can be reconstructed. The placement of markers is tedious and time consuming and requires a long setup time. Furthermore, optical marker based systems are very expensive. Perhaps, the most limiting factor is that marker based systems usually require an engineered controlled indoor studio. MoCap outdoors is appealing because the human motion of certain activities can be captured in its natural environment, *i.e.*, one can track a tennis player in the court itself. Furthermore, it is unpractical for man-machine interaction to be limited to engineered intelligent rooms. MoCap systems based only on IMUs such as *XSens* [6] also exist and are very appealing because they are portable outdoors and are more cost

---

[3] www.vicon.com
[4] www.simi.com
[5] www.qualisys.com
[6] www.xsens.com

efficient. However, the subject has to wear a suit with at least 17 sensors which hampers the range of motion. The main limitation though is that IMUs are sensitive to magnetic disturbances and suffer from drift. This results in tracking failure during continuous operation. Nonetheless, IMUs are appealing because they are inexpensive and they are integrated in most mobile devices. Therefore, inexpensive solutions that allow MoCap in uncontrolled environments are needed.

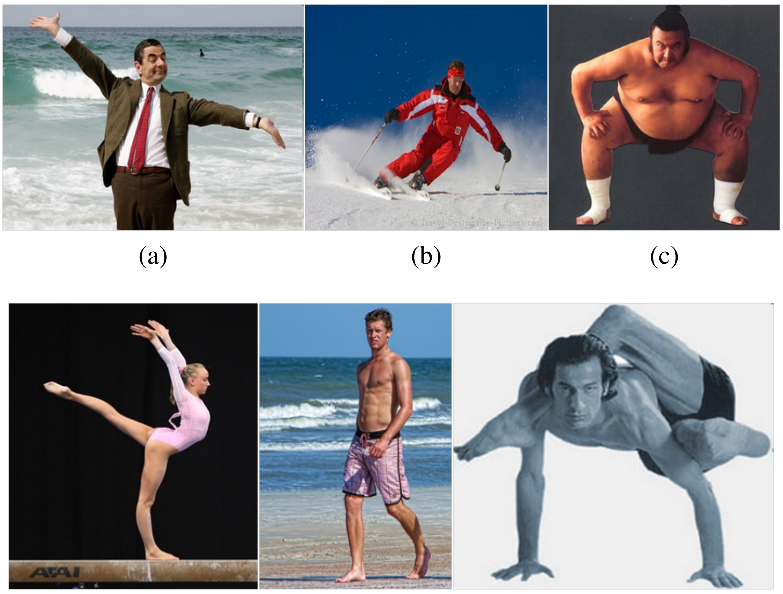
Hence, our first main contribution consists of a hybrid system that integrates information from a *small set of IMUs and at least 2 consumer cameras* which enables accurate pose reconstruction outdoors. The main motivation is that 3D orientation is difficult to reconstruct from images due to rotational ambiguities and is directly measurable by a IMU device. On the other hand, accurate joint locations are difficult to recover from IMUs and relatively easier to obtain from images. Therefore, an intelligent system combining both sensor types can recover human motion by compensating the drawbacks of each sensor type. For applications such as character animation or medical image analysis accurate pose is necessary and engineering the system is reasonable as long as it does not become uncomfortable for the subject and it permits capture outside of the lab. Therefore, our solution is a good trade-off between accuracy and engineering.

In some applications however, it is necessary to understand pose from a single image. This is the case if we want to interact through the camera mounted in mobile devices, or if we want to understand scenes in video collections from the Internet. Unfortunately, accurate 3D pose recovery from a single image is an extremely difficult problem because some degrees of freedom are not observable due to the projection into the image plane. Fortunately, accurate estimation of the pose might not be necessary for many of these applications. This motivates our second major contribution. Recent studies show that humans do not perceive absolute joint locations very accurately. However, humans are extremely good at extracting semantic information, *i.e.*, they can distinguish whether the arms are crossed, whether one hand is over the head or whether the head is tilted. This is probably due to the fact that body language is more closely related to relative location between body parts rather than absolute positions. Therefore, we propose a discriminative inference algorithm that extracts this *semantic attributes* directly from the images. This inferred bits of information can be used directly as input for the understanding of the physical world or as a prior for a more fine detailed accurate pose estimator.

1.4 Challenges

The problem of estimating pose from video images and inertial sensors is a very challenging one. We briefly summarize the main technical difficulties below:

- **Image**: An image is the projection of the 3D world onto a 2D image plane. Hence, the depth information is lost during the process.
  - *Depth and Orientation Ambiguities*: The depth information is lost during the image formation process. Recovering it implies hallucinating some degrees of freedom. The fact that multiple 3D hypothesis explain the image well makes the energy minimization formulations multi-modal. In particular, axial rotations are very difficult to observe from images since they project to almost the same image.
  - *Occlusions*: Parts of the body can be partially occluded by other objects in the world or some parts might be occluded by the body itself (self-occlusions).
  - *Appearance Changes*: People appear in images in different clothing, different illuminations, shapes and backgrounds. A true generative method should model all the intricacies of the image formation process. Obviously the high appearance variability makes it difficult to model. On the one hand, a very detailed model might work best for a specific



(a) (b) (c)

(d) (e) (f)

Figure 1.3: Challenges in image based pose estimation: The high variability in human shape, clothing, appearance, illumination, background and the high dimensionality of the pose space make the problem extremely challenging.

instance but might not generalize well to other scenes. On the other hand, a very coarse model will not truthfully synthesize the image.

- **IMUs**: an inertial measurement unit is a device that measures orientation and acceleration of the local coordinate system w.r.t. a global coordinate system.
  - *Drifting*: The orientation measurements are obtained by merging the data coming from a gyroscope a compass and an accelerometer in a Kalman filter. To obtain orientation the gyroscope data needs to be integrated which can cause drift in continuous operation.
  - *Magnetic Disturbances*: the data can corrupted by magnetic disturbances produced by electronic equipment in the surroundings.
  - *Latency* : Since the orientation data comes as the output of a Kalman filter, there exists a non negligible latency. This is specially noticeable during fast motions.

- *Synchronization*: One of the main problems with sensor fusion is proper synchronization. The inertial sensors and the cameras operate at different frame rates. Furhtermore, we do not assume any kind of communication between sensors. Synchronization between the cameras and the IMUs is specially an isssue during fast motions.

- *High-dimensional search space*: there are more than 206 bones in the human body. Usually, the search space is constrained to 20 to 60 joint angles that determine the pose configuration through a skeletton model. Even with this rather coarse representation the search space is very high dimensional. Therefore, efficient optimization and search strategies are needed. A further complication is that only a few regions of parameter space correspond to valid human poses.

Some of these difficulties are also illustrated in Fig. 1.3.

1.5 Generative versus Discriminative

Most approaches to pose estimation from images can be classified as generative or discriminative. Generative approaches aim at modeling all the intricacies of the image formation process. A truly generative method would model human pose, shape, clothing, illumination and even background. In practice, generative methods only model image features that are easy and efficient to extract and to synthesize. By focusing on features that are relatively invariant to clothing and illumination, researchers circumvent the burden of building very truthful models. As such, samples from such models are hardly indicative of what one can observe in a real image. Nonetheless, such models are useful for inference and are known to generalize well to different motion patterns and different appearance conditions. Generative methods usually define an energy function that has to be optimized w.r.t. to the pose parameters. The energy measures the consistency between the model and the observations. The hope is that the global optimum corresponds to the true pose. Since evaluating this energy is typically expensive and the search space is huge, generative methods are only feasible when some initialization is provided (*e.g.*, the pose in the previous frame in a video sequence).

Discriminative models learn a direct mapping from image features to the pose space using training data. In contrast to generative methods, discriminative methods are more direct and are therefore easier to implement. When the distribution of the training is similar to the distribution of the testing such methods perform very well. However, generalization to instances unseen in the training data is often a challenge. Furthermore, obtaining training pairs of 3D poses and images in different backgrounds and for different motions is not a trivial task. There is no consensus in which of the two approaches performs best because each method has its advantages and shortcomings. However, although these two lines of approaches have been mostly investigated separately, they can complement each other. Discriminative methods can provide rough idea about the pose and generative methods can provide the fine detail. It is reasonable to believe that there is some discriminative and some generative process in the visual system. While the first two tracking systems described in Chap. 4 and Chap. 5 are purely generative, the method presented in Chap. 6 can be seen as a hybrid combining ideas from discriminative and generative methods.

1.6 Related Work

Given the wide range of applications of Motion Capture, it is not surprising that it has received a lot of attention in the computer vision community during the last two decades. While a vast number of papers have been published in the area of pose estimation from images, much fewer papers can found on human pose from IMUs. The work presented in this thesis one of the first to combine images and IMUs for full body pose estimation. The literature on pose from images is vast and an extensive survey is out of the scope of this thesis. For a more thorough review paper, we refer the reader to [1]. We give here a brief review of recent related papers on the subject.

We divide the related work in three subgroups: image based pose estimation methods, inertial sensor trackers and finally, methods that, like our semantic attribute description, propose alternative representation of pose.

1.6.1 Image based Pose Estimation

Generative methods consist of a likelihood model, a search space strategy [2], and a prior [3]. Since sampling directly from the posterior is difficult, the problem is typically inverted to model the likelihood of the image features given a pose configuration. The likelihood models must be easy to compute and are usually based on edges, histograms or silhouette overlap terms [4, 5, 6]. The difficulties arise from multi-modality and ambiguous data association. Some works deal with the data association by introducing more robust matching costs [7, 8]. More sophisticated likelihoods that include illumination and shadows [9, 10, 11] and color appearance [12, 13].

Multi-modality occurs when the likelihood model scores well different pose configurations in 3D. This has been addressed in the past by building activity specific priors [14, 15, 12, 16, 17, 18]. By sampling from an activity specific manifold ambiguities are reduced at the cost of limited applicability to track general human motions. While approaches exist to account for transitions between different types of motion [19, 20, 21, 22], general human motion is highly unpredictable and difficult to be modeled by pre-specified action classes. Our work is nonetheless related to [19, 21, 22] in that we also define an intermediate pose representation. However, posebits are more flexible than action classes because they are compositional and do not suffer from big intra-class variability. Some works have shown that under certain features, it is possible to sample directly from the posterior, or at least to sample in regions of high likelihood probability [23, 24]. In [25] we propose to sample directly from the manifold of poses that satisfy a set of kinematic constraints derived from measurements. In [26] they showed that given the 2D joint locations and known limb proportions, the pose can be reconstructed up to a flip forward-backward ambiguity for every limb. The works of [23, 27, 24] exploit this property by sampling proposals in 3D that project to a given 2D joint location and propagate this multiple modes during tracking.

Discriminative methods model a mapping from image features to the pose space by using training examples. Approaches are either based on nearest neighbors (KNN) schemes [28, 29] or global parametric predictors [30, 31]. Since it is difficult or even impossible to fit a single mapping across the joint feature and pose space, recent works [32, 33, 34] build online local models from a subset of training exemplars found using *e.g.*, KNN or decision trees [35]. Another way to correlate the input features with the output poses is through shared latent variables [17, 21, 34]. Similarly, posebits can be interpreted as a mid-layer latent variable that links the feature and pose spaces with the main difference that posebits have a semantic meaning interpretable by humans. In contrast to fully discriminative methods our proposed algorithm does not require training pairs of images and poses which enables easy annotation and data collection at large in natural scenes.

1.6.2 Inertial Sensor Trackers

Recently, inertial sensors (e.g. gyroscopes and accelerometers) have become popular for human motion analysis. Often, sensors are used for medical applications, see, *e.g.*, [36] where accelerometer and gyroscope data is fused. However, their application concentrates on the estimation of the lower limb orientation in the sagital plane. In [37], a combination of inertial sensors and visual data is restricted to the tracking of a single limb (the arm). Moreover, only a simple red arm band is used as image feature. In [38], data obtained from few accelerometers is used to retrieve and play back human motions from a database. [39] presents a system to capture full-body motion using only inertial and magnetic sensors. While the system in [39] is very appealing because it does not require cameras for tracking, the subject has to wear a suit with at least 17 inertial sensors, which might hamper the movement of the subject. In addition, long preparation time before recording is needed. Moreover, inertial sensors suffer from severe drift problems and cannot provide accurate position information in continuous operation.

1.6.3 Representations of Pose

The pose is typically represented by a set of joint angles or joint positions that have to be inferred from observations. Very few works have explored alternative representations of pose. Recent works have used attribute representations for object categorization [40, 41], and human action recognition [42, 43] with a special emphasis on transfer learning between classes. Similar semantic attributes as the ones we introduce in this thesis have been used for content based retrieval of motion capture data [44], where they noted that similar motions may be numerically not similar. In [19], they also used attributes for retrieving motion priors to stabilize tracking. They do it in an iterative scheme in which they start tracking with no prior knowledge and use noisy tracking estimates to retrieve action specific priors. Our work is also inspired by [45] where they introduce poselets. A poselet is a new notion of part, they noted that parts need not to correspond to physical body parts as in [46, 47, 48]. Instead, they argue that detecting subgroups of body parts in a certain configuration is easier.

1.7 Contributions

In this thesis we have developed several of techniques to recover the pose from images and IMUs at different levels of pose detail in both indoor and outdoor scenarios. In particular, we try to answer the following two research questions:

1. *What cues are necessary for accurate pose estimation in uncontrolled environments?*
2. *What is a suitable representation of the pose?*

The answer to these questions is application dependent as we will see. Bearing the application requirements in mind, the main novel techniques developed can be summarized as:

1. **Local Hybrid Tracker (LHT) :** We have developed a system to recover the highly accurate pose from multiview images and IMUs in indoor scenarios, see Chap. 4. This is the first work on pose estimation to combine information from images with information from IMU for full body pose estimation. The method integrates image cues such as silhouettes and edges with orientation cues coming from 5 inertial sensors attached at the body extremities, see Fig. 1.4. In indoor scenarios relatively clean silhouettes can be obtained using background subtraction techniques. Therefore, the approach relies on *local optimization* to integrate the cues which is fast and efficient. In contrast to existing approaches we can not only recover accurate body part locations but also accurate body part orientations which are very difficult to estimate from only image cues. Body part orientation is equally important for applications such as sports science, medicine or character animation.

2. **Inverse Kinematics Sampling Tracker (IKST) :** We have developed a method to recover the pose in outdoor scenarios from multiview images and IMUs see Chap. 5. Here the main challenge are the limited noisy and ambiguous image cues that can be extracted in outdoor scenarios. Background clutter, inaccurate segmentation, occlusions and fast motions make the problem even more challenging. To this end, we propose and efficient sampling procedure that can be interred in *particle based optimization* methods. The main idea is a strategy to sample pose hypothesis that satisfy the set of orientation constraints imposed by the sensors. This in turn, reduces the dimension of the state space. That is achieved using an efficient sampling procedure that uses *inverse kinematics* to satisfy a set of constraints, Fig. 1.5. The method also incorporates a novel model to cope with sensor noise. The method enables high accurate pose estimation in outdoor scenarios using only 4 consumer cameras in combination with 5 IMUs.

3. **Posebits Pose Estimator (PPE):** We have developed a novel approach to estimate the pose from monocular images, see Chap. 6. Here we rely on the idea that human perception of poses is more semantic than numeric. Ours is one of the first works that proposes a more qualitative description of pose as opposed to common numeric representations seen in the literature. We introduce the concept of *posebits*. Posebits are units of information that resolve typical ambiguities in monocular imagery. They are boolean geometric relationships between body

parts designed to mimic human perception of poses (*e.g.* left-leg in front of right-leg or hands close to each other). In this way, the pose is described by a string of posebits. A classifier is trained using *structural* SVM (Support Vector Machines) to infer these bits of information directly form image features, see Fig. 1.6. We further propose a method to generate posebits automatically and a selection algorithm inspired by *decision trees* to retain a good set. We further show how to use these bits as a powerful feature for applications such as content based retrieval. Furthermore, we show that once these bits of information are known, the problem of monocular pose estimation becomes easier and less ambiguous.

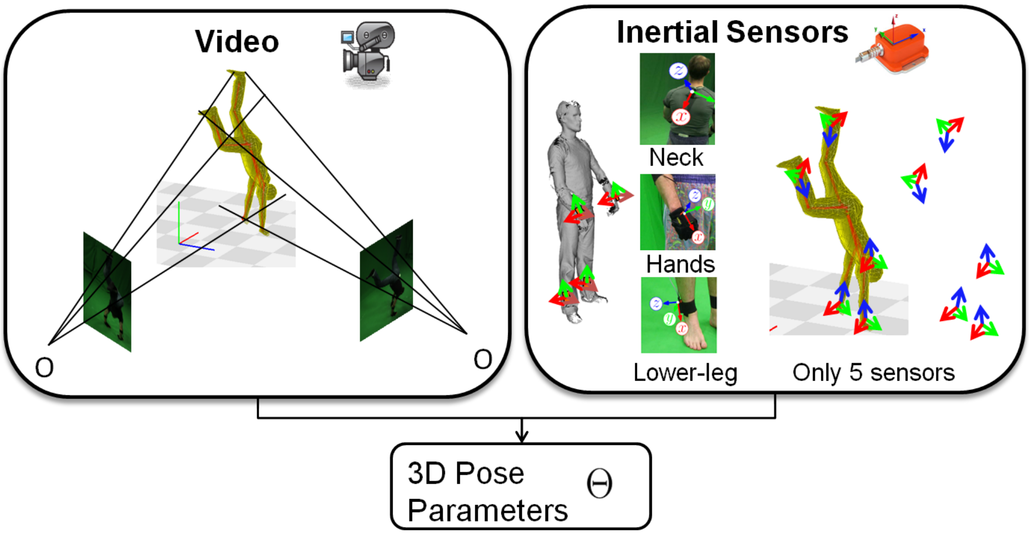


Figure 1.4: Hybrid local tracker: Information coming from video cameras and the orientation data coming from 5 IMU sensors attached at the body extremities is combined to obtain high accurate pose estimation results. We formulate an energy that measures the pose consistency with the image observations as well as the IMUs orientations. Since in indoor scenarios good quality segmentation is possible we opt here to fuse the information using a local optimization method which is very efficient.

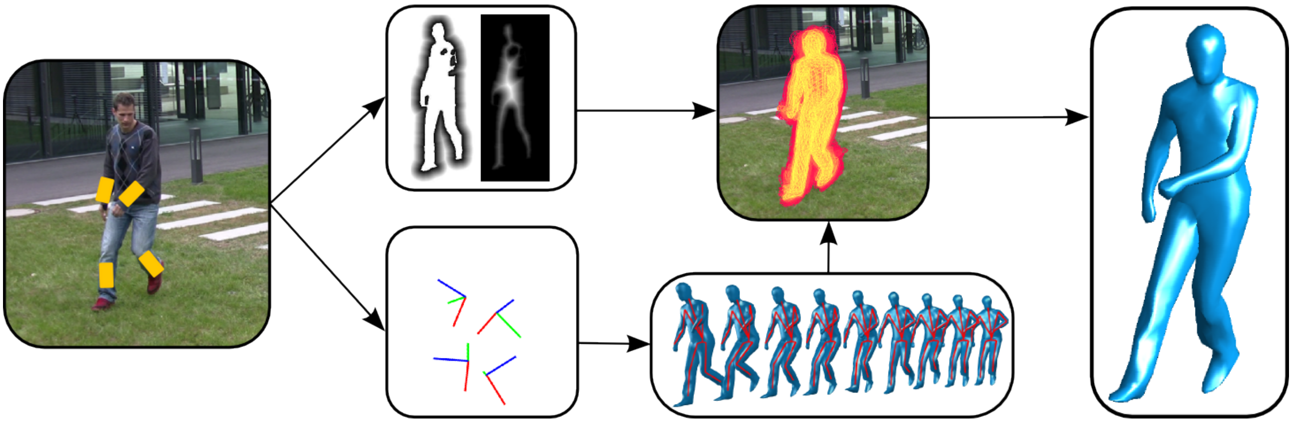
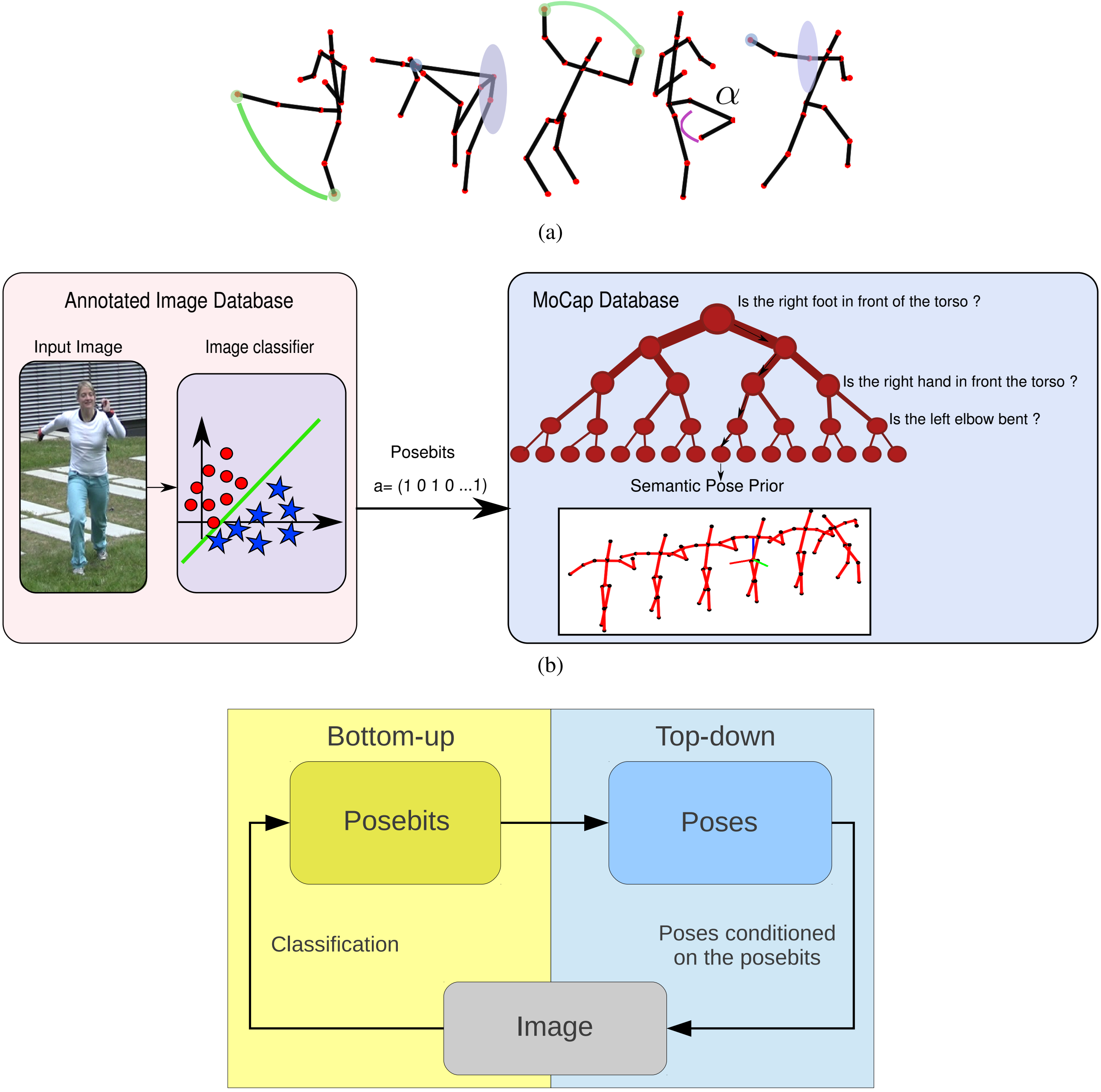


Figure 1.5: Inverse Kinematics sampler: to deal with background clutter and image ambiguities in outdoors scenarios we build upon a particle-based optimization method. The key idea is an efficient sampling procedure based on inverse kinematics to generate pose hypothesis that satisfy the set of orientation constraints imposed by the sensors.



(a)

(b)

(c)

Figure 1.6: Posebits Pose Estimator: (a) Posebits are bits of information that disambiguate poses. Examples of posebits are: *Is the left foot from the right foot,is right knee bent* or *is the right arm extended*. The advantage to represent the pose as set of posebits is two fold: first, collecting training data simplifies to answering a set of yes/no questions and second inference should be significantly easier than estimating exact 3D joint locations. In (b) we show an approach to generate a pose prior given a set of inferred posebits. This pose prior can be used in a standard generative tracker to reduce uncertainty. Given training set of annotated images, image classifiers are trained to infer posebits. Posebits can also be used to cluster MoCap data in semantically meaningful classes (right of (b). The pose prior is obtained by querying the MoCap database with the inferred image posebits. In (c) we show a diagram of the proposed algorithm: posebits are inferred bottom-up. These posebits are leveraged to draw likely pose proposals that are evaluated against the image top-down.

1.8 Overview

- **Chapter 1. Introduction:** Problem Statement, motivation challenges and contributions.
- **Chapter 2. Generative Pose Estimation:** the foundations to build a generative pose tracking system are explained in this chapter. It includes dedicated sections on model generation, parametrization, likelihood modeling and optimization strategies.
- **Chapter 3. Inertial Sensors:** Brief description of hardware specifications and IMU data acquisition and synchronization.
- **Chapter 4. Indoor Motion Capture from Video and Inertial Sensors:** A hybrid tracking system that integrates information from cameras and IMUs is presented. The main applicability is for accurate pose estimation in indoor scenarios is described. The tracker is based on local optimization and is therefore very efficient. The system allows to capture accurate joint locations and to estimate accurate limb orientation for a wide range of complex motions.
- **Chapter 5. Outdoor Motion Capture from Video and Inertial Sensors:** A hybrid tracking system that integrates information from cameras and IMUs is presented. The main features of the tracker are its applicability in outdoor scenarios for a wide range of complex motions, including interaction with objects. To deal with the uncertainties in the data in outdoors this tracker is based on particle based optimization. The main novelty is an efficient sampling procedure to generate hypothesis consistent with the orientation constraints imposed by the IMUs. A novel noise model based on the von Mises-Fisher distribution is also introduced. The main applicability is MoCap of activities that are best performed outdoors.
- **Chapter 6. Posebits for Monocular Pose Estimation:** A method to infer useful semantic pose information (posebits) from single monocular images is described here. The model infers semantic pose information discriminatively directly from image features. We show how posebits can be used as input for a generative tracker. We note that posebits are a powerful source of information that can be useful for many vision based intelligence tasks, such as man-machine interaction, action and gesture recognition or scene understanding.
- **Chapter 7. Conclusions:** Summary the contributions of this work and proposed extensions and future directions of research.

1.9 List of Published Papers

The following papers have been published during the time of this dissertation. Papers are listed in the order of relevance to the topic of this thesis.

G. Pons-Moll, B. Rosenhahn
Model Based Pose Estimation
in Visual Analysis of Humans: Looking at People, (**Book chapter**), Springer, June 2011 edited by T.Moeslund, A.Hilton, V. Krueger and L. Sigal, 2011 [2]

G. Pons-Moll, A. Baak, J. Gall, L. Leal-Taixé, M. Mueller, H.-P. Seidel and B. Rosenhahn *Outdoor human motion capture using inverse kinematics and von Mises-Fisher sampling*
IEEE International Conference on Computer Vision (**ICCV**) 2011 [25]

G. Pons-Moll, A. Baak, T. Helten, M. Mueller, H.-P. Seidel and B. Rosenhahn
Multisensor-Fusion for 3D Full-Body Human Motion Capture
IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**) 2010 [49]

G. Pons-Moll, D. Fleet and B. Rosenhahn
Posebits for Monocular Human Pose Estimation
IEEE International Conference on Computer Vision and Pattern Recognition (**CVPR**) 2014 [50]

G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, A.Fitzgibbon
Metric Regression Forests for Human Pose Estimation
British Conference on Machine Vision (**BMVC**), 2013 [51]

G. Pons-Moll, L. Leal-Taixé, T. Truong, B. Rosenhahn
Efficient and robust shape matching for model based human motion capture
33rd Annual Symposium of the German Association for Pattern Recognition (**DAGM**), 2011 [8]

G. Pons-Moll, B. Rosenhahn
Ball Joints for Marker-less Human Motion Capture
IEEE Workshop on Applications of Computer Vision (**WACV**), Snow Bird, Utah, USA, December 2009 [52]

G. Pons-Moll, L. Leal-Taixé, J. Gall, and B. Rosenhahn
Data-driven Manifolds for Outdoor Motion Capture
Theoretic Foundations of Computer Vision: Outdoor and Large-Scale Real-World 2011 [53]

L. Leal-Taixé, **G. Pons-Moll** and B. Rosenhahn
Branch-and-price global optimization for multi-view multi-object tracking
IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**) 2012 [54]

L. Leal-Taixé, **G. Pons-Moll**, B. Rosenhahn
Everybody needs somebody: modeling social and grouping behavior on a linear programming multiple people tracker
IEEE International Conference on Computer Vision Workshops (**ICCV-W**). 1st Workshop on Modeling, Simulation and Visual Analysis of Large Crowds, 2011 [55]

L. Leal-Taixé, **G. Pons-Moll** and B. Rosenhahn
Exploiting pedestrian interaction via global optimization and social behaviors
Theoretic Foundations of Computer Vision: Outdoor and Large-Scale Real-World 2011 [56]

A. Kuznetsova, **G. Pons-Moll**, and B. Rosenhahn
PCA-enhanced stochastic optimization methods
34th Annual Symposium of the German Association for Pattern Recognition (**DAGM**) 2012[57]

A. Baak, T. Helten, M. Mueller, **G. Pons-Moll**, B. Rosenhahn, H.-P. Seidel
Analyzing and Evaluating Markerless Motion Tracking Using Inertial Sensors
European Conference on Computer Vision (**ECCV-W**), Third Workshop on Human Motion: Understanding, Modeling, Capture and Animation, September 2010[58]

G. Pons-Moll, Gilead Tadmor, Rob S. MacLeod, Bodo Rosenhahn, Dana H. Brooks
4D Cardiac Segmentation of the Epicardium and Left Ventricle

World Congress of Medical Physics and Biomedical Engineering (**IFMBE**), Munich (Germany), September 2009 [59]

G. Pons-Moll, C. Crosas, G. Tadmor, R. MacLeod, B. Rosenhahn, D. Brooks
Parametric Modeling of the Beating Heart with Respiratory Motion Extracted from Magnetic Resonance Images
IEEE Computers in Cardiology (**CINC**), Park City, Utah, USA, September 2009 [60]

2 Generative Pose Estimation

Model-based pose estimation algorithms aim at recovering human motion from one or more camera views and a 3D model representation of the human body. The model pose is usually parameterized with a kinematic chain and thereby the pose is represented by a vector of joint angles. The majority of algorithms are based on minimizing an error function that measures how well the 3D model fits the image. This category of algorithms usually have two main stages, namely defining the model and fitting the model to image observations. In the first section, the reader is introduced to the different kinematic parametrization of human motion. In the second section, the most commonly used representations of human shape are described. The third section is dedicated to the description of different error functions proposed in the literature and on common optimization techniques used for human pose estimation. Specifically, local optimization and particle based optimization and filtering are discussed and compared. The chapter concludes with a discussion of the state-of-the-art in model-based pose estimation, current limitations and future directions.

2.1 Kinematic Parametrization

In this chapter our main concern will be on estimating the human pose from images. Human motion is mostly articulated, *i.e.*, it can be accurately modeled by a set of connected rigid segments. A *segment* is a set of points that move rigidly together. To determine the pose, we must first find an appropriate parametrization of the human motion. For the task of estimating human motion a *good* parametrization must have the following attributes:

Attributes of a good parametrization for human motion:

- Pose configurations are represented with the minimum number of parameters.
- Human motion constraints, such as articulated motion, are naturally described.
- Singularities can be avoided during optimization.
- Easy computation of derivatives of segment positions and orientations *w.r.t.* the parameters.
- Simple rules for concatenating motions.

A commonly used parametrization that meets most of the above requirements is a kinematic chain, which encodes the motion of a body segment as the motion of the previous segment in the chain and an angular motion about a body joint. For example, the motion of the lower arm is parametrized as the motion of the upper arm and a rotation about the elbow, see Fig. 2.3. The motion of a body segment relative to the previous one is parametrized by a rotation. Parameterizing rotations can be tricky since it is a non-Euclidean group, which means that if we travel any integer number of loops around an axis in space we will end up in the same point. We now briefly review the different parametrization of rotations that have been used for human tracking.

2.1.1 Rotation Matrices

A rotation matrix $\mathbf{R}_{3\times3}$ is an element of $SO(3)$. Elements of $\mathbf{R} \in SO(3)$ are the group of 3×3 orthonormal matrices with $\det(\mathbf{R}) = 1$ that represent rotations [61]. A rotation matrix encodes the orientation of a frame B that we call *body frame* relative to a second one S that we call *spatial frame*. Given a point $\mathbf{p}$ with body coordinates, $\mathbf{p}_b = (\lambda_x, \lambda_y, \lambda_z)^T$, we might write the point $\mathbf{p}$ in spatial coordinates as

$$\mathbf{p}_s = \lambda_x \mathbf{x}_s^B + \lambda_y \mathbf{y}_s^B + \lambda_z \mathbf{z}_s^B, \tag{2.1}$$

where $\mathbf{x}_s^B$,$\mathbf{y}_s^B$,$\mathbf{z}_s^B$ are the principal axis of the body frame B written in spatial coordinates. We may also write the relationship between the spatial and body frame coordinates in matrix form as $\mathbf{p}_s = \mathbf{R}_{sb}\,\mathbf{p}_b$. From this it follows that the rotation matrix is given by

$$\mathbf{R}_{sb} = \begin{bmatrix} \mathbf{x}_s^B & \mathbf{y}_s^B & \mathbf{z}_s^B \end{bmatrix} \tag{2.2}$$

Now consider a frame B whose origin is translated *w.r.t.* frame S by $\mathbf{t}_s$ (the translation vector written in spatial coordinates). In this case, the coordinates of frames S and B are related by a rotation and a translation, $\mathbf{p}_s = \mathbf{R}_{sb}\mathbf{p}_b + \mathbf{t}_s$. Hence, a pair $(\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3)$ determines the configuration of a frame B relative to another S and is the product space of $\mathbb{R}^3$ with $SO(3)$ denoted as $SE(3) = \mathbb{R}^3 \times SO(3)$. Elements of $SE(3)$ are $\mathbf{g} = \{\mathbf{R},\mathbf{t}\}$. Equivalently, writing the point in homogeneous coordinates $\bar{\mathbf{p}}_b = \begin{bmatrix} \mathbf{p}_b \\ 1 \end{bmatrix}$ allows us to use the more compact notation The rigid body motion is then completely represented by the matrix $\mathbf{G}_{sb}$ which is the homogeneous representation of $\mathbf{g}_{sb}$. The reader unfamiliar with rotation matrices might be surprised because the definitions given here for rotation and rigid motion do not represent motion of points in a fixed frame but rather transformations between coordinate systems. This does not correspond to our informal understanding of rotations. Consequently, *do rotations and rigid body motion represent coordinate transformations or motion?*

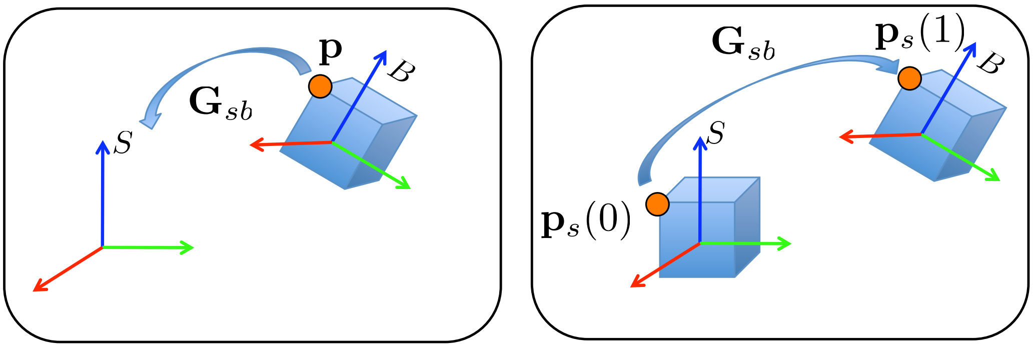


Figure 2.1: Left: rigid body motion seen as a coordinate transformation, Right: rigid body motion seen as a relative motion in time

The answer is both. To see this, consider a point $\mathbf{p}$ in a rigid body, see Figure 2.1, and imagine that the body and spatial frames coincide at $t = 0$ see Figure 2.1 right, consequently $\mathbf{p}_s(0) = \mathbf{p}_b$. At this time we apply the rigid body motion to the point such that the point now moves to a new position $\mathbf{p}_s(1)$. We can write it as,

$$\bar{\mathbf{p}}_s(1) = \mathbf{G}_{sb}\,\bar{\mathbf{p}}_s(0) \tag{2.3}$$

where the coordinates of $\mathbf{p}_s(1)$ and $\mathbf{p}_s(0)$ are both relative to the spatial frame. This new interpretation of rigid motion will be very useful when we talk about human motion in the next section. Both interpretations of rigid motion are correct and depending on the context one is preferable over the other, (*e.g.*, to think about world-to-camera mapping it is better to interpret it as a coordinate transformation but when we think of human motion it is most of the times more intuitive to think of rigid motion as the relative motion between temporal instants). Rotations can be combined by simple matrix multiplication. However, representing rotations with rotation matrices is suboptimal for optimization problems. This is because from the 9 numbers composing the matrix, 6 additional constraints must be imposed during optimization in order to ensure that the matrix is orthonormal. Therefore, representing angular motions with rotation matrices is problematic for motion tracking because we need more parameters than strictly needed.

2.1.2 Euler Angles

One method for describing the orientation of a frame B relative to another frame S is as follows: start with frame B coincident with frame S, rotate B α degrees about the x-axis of frame S, then rotate β degrees about the y-axis of frame S and finally rotate γ degrees about the z-axis (of frame S again). This corresponds to the x,y,z Euler angles defined in frame S. There are several conventions on the order in which these rotations are carried out; for example, it is also possible to perform the rotation in the order z,y,z. Therefore, when we talk about Euler angles the order of the rotations must be specified. It is very important to note with respect to which frame the rotations are defined, they can be defined on the fixed reference frame S or alternatively on the moving frame B. Therefore, rotation matrix can always be written as the composition of three rotations around the x,y,z axes Eq. (2.4). Note that had we chosen the rotations to be defined in the moving frame B, the order would be inverted.

$$\mathbf{R} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta)) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{2.4}$$

In this manner, a rotation is completely defined by a triplet of Euler angles (α,β,γ). The derivatives of a rotation with respect to the Euler angles are easy to compute. Additionally, differential equation integration in parameter space is straightforward, for example to update one of the three angles: $\alpha_t = \alpha_{t-1} + \dot{\alpha}$. Unfortunately, Euler angles have a well known problem: when two of the rotation axis align one of the rotations is lost. This well known singularity of Euler parametrization is called gimbal lock. For more details we refer to [61].

2.1.3 Quaternions

Quaternions generalize complex numbers and can be used to represent 3D rotations the same way as complex numbers can be used to represent planar rotations. Formally, a quaternion is a vector quantity of the form $\mathbf{q} = q_w + q_x \cdot \mathbf{i} + q_y \cdot \mathbf{j} + q_z \cdot \ \mathbf{k}$ with $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{i} \cdot \mathbf{j} \cdot \mathbf{k} = -1$, see [62]. They can also be interpreted as a scalar q_w plus a 3-vector $(q_w,\mathbf{v})$. The square root of the product of the quaternion and its conjugate is the norm

$$\|q\| = \sqrt{\mathbf{q}\bar{\mathbf{q}}} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \tag{2.5}$$

where $\bar{\mathbf{q}} = (q_w, -\ \mathbf{v})$ is the complex conjugate of $\mathbf{q}$. Unit length quaternions, *i.e.*, $\|q\| = 1$ form a set called S^3 which can be used to carry out rotations [62]. One nice property about quaternions is that rotations can be carried out in parameter space via quaternion product. Given two quaternions $\mathbf{q}_1 = (q_{w,1},\mathbf{v}_1)$ and $\mathbf{q}_2 = (q_{w,2},\mathbf{v}_2)$ the quaternion product dennoted by $(\circ)$ is defined as

$$\mathbf{q}_1 \circ \mathbf{q}_2 = (q_{w,1}q_{w,2} - \mathbf{v}_1 \cdot \mathbf{v}_2 \ , \ q_{w,1}\mathbf{v}_2 + q_{w,2}\mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2). \tag{2.6}$$

If we want to rotate a vector $\mathbf{a}$ we can simply use the quaternion product. Thereby, a rotation by an angle θ about an axis ω is represented by the quaternion:

$$\mathbf{q} = [q_w,q_x,q_y,q_z]^T = \left(\cos\left(\frac{\theta}{2}\right),\omega \sin\left(\frac{\theta}{2}\right)\right) \tag{2.7}$$

and the vector $\mathbf{a}$ is rotated with

$$\mathbf{a}' = Rotate(\mathbf{a}) = \mathbf{q} \circ \tilde{\mathbf{a}} \circ \bar{\mathbf{q}} \tag{2.8}$$

where $\circ$ denotes quaternion product, $\tilde{\mathbf{a}} = [0, \mathbf{a}]$ is a zero scalar component appended with the original vector $\mathbf{a}$ and $\bar{\mathbf{q}} = (q_w, -\mathbf{v})$ is the complex conjugate of $\mathbf{q}$. Additionally, there exist simple formulae for computing the rotation matrix from a quaternion and vice versa [62]

$$\mathbf{R} = \begin{bmatrix} 1 - 2q_y^2 - 2z^2 & 2q_xq_y + 2q_wq_z & 2q_xq_z - 2q_wq_y \\ 2q_xq_y - 2q_wq_z & 1 - 2q_x^2 - 2q_z^2 & 2q_yq_z + 2q_wq_x \\ 2q_xq_z & 2q_yq_z - 2q_wq_x & 1 - 2q_x^2 - 2q_y^2 \end{bmatrix} \tag{2.9}$$

.

Furthermore, the four partial derivatives $\frac{\partial \mathbf{R}}{\partial q_w},\frac{\partial \mathbf{R}}{\partial q_x}, \frac{\partial \mathbf{R}}{\partial q_y},\frac{\partial \mathbf{R}}{\partial q_z}$ exist and are linearly independent in S^3 which means there are no singularities. Probably this last property is the most interesting but this comes at the expense of using 4 numbers instead of just 3. This means that during optimization we must impose a quadratic constraint so that the quaternion keeps unit length. Integrating Ordinary differential equations (ODEs) can also be problematic since the quaternion velocity $\dot{\mathbf{q}}$ generally lies in the tangent space [1] of S^3 and any movement in the tangent plane will push the quaternion *off* S^3. Nonetheless, there exist solutions to these limitations [63, 62]. Since unit quaternions directly represent the space of rotations and are free of singularities they provide an efficient representation of rotations. Particularly, quaternions have proven to be very useful for the interpolation of key-frame poses because they respect $SO(3)$ geometry.

2.1.4 Axis-angle

To model human joint motion it is often needed to specify the axis of rotation of the joint. For example we might want to specify the motion of the knee joint as a rotation about an axis perpendicular to the leg and parallel to the hips. Therefore, for our purpose the axis-angle representation is optimal because rotations are described as an angle θ and an axis in space $\omega \in \mathbb{R}^3$ where θ determines the amount of rotation about ω. Unlike quaternions the axis-angle, requires only 3 parameters $\theta\omega$ to describe a rotation. It does not suffer from gimbal lock and their singularities occur in a region of parameter space that can be easily avoided. Since it will be our parametrization of choice to model human joint motion we will give a brief introduction to the formulation of twists and exponential maps. For a more detailed description we refer the reader to [61].

The Exponential Formula

Every rotation $\mathbf{R}$ can be written in exponential form in terms of the axis of rotation $\omega \in \mathbb{R}^3$, s.t. $||\omega|| = 1$ and the angle of rotation θ as

$$\mathbf{R} = \exp(\theta\widehat{\omega}) \tag{2.10}$$

where $\widehat{\omega} \in so(3)$ is the skew symmetric matrix constructed from ω. The elements of $so(3)$ are skew symmetric matrices *i.e.*, matrices that verify $\{\mathbf{A} \in \mathbb{R}^{3 \times 3} | \mathbf{A} = -\mathbf{A}^T\}$. Given the vector $\theta(\omega_1,\omega_2,\omega_3)$ the skew symmetric matrix is constructed with the wedge operator $\wedge$ as follows:

$$\theta\widehat{\omega} = \theta \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{2.11}$$

By definition, the multiplication of the matrix $\widehat{\omega}$ with a point $\mathbf{p}$ is equivalent to the cross-product of the vector ω with the point.

---

[1] This will be described in more detail in Sec. 2.1.4]

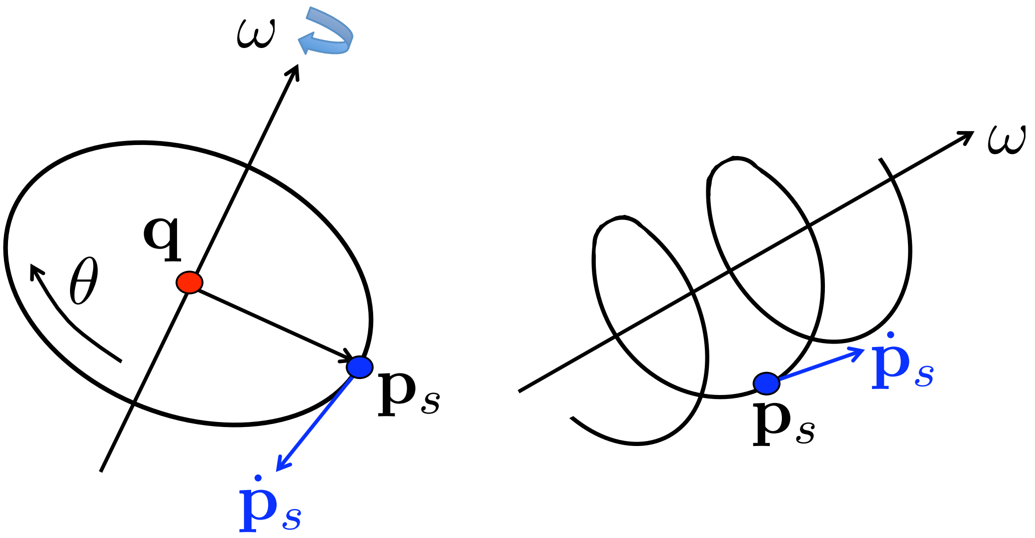


Figure 2.2: Screw motion, left: the cross product of the scaled axis $\theta\omega$ and the vector $(\mathbf{p}_s - \mathbf{q})$ results in the tangential velocity of the point $\dot{\mathbf{p}}_s = \theta\omega \times (\mathbf{p}_s - \mathbf{q})$. Equivalently, the tangential velocity may be written using the twist $\dot{\mathbf{p}}_s = \widehat{\xi}\,\mathbf{p}_s$. Right: generalized screw motion with rotation and translation along the axis

To derive the exponential formula in Eq. (2.10) consider a 3D point $\mathbf{p}$ rotating about an axis ω intersecting the origin at a unit constant angular velocity. Recall from elementary physics that the tangential velocity of the point may be written as

$$\dot{\mathbf{p}}(t) = \omega \times \mathbf{p}(t) = \widehat{\omega}\mathbf{p}(t) \tag{2.12}$$

which is a differential equation that we can integrate to obtain

$$\mathbf{p}(t) = \exp(\widehat{\omega}t)\mathbf{p}(0) \tag{2.13}$$

It follows that if we rotate θ units of time the net rotation is given by:

$$\mathbf{R}(\theta,\omega) = \exp(\theta\widehat{\omega}) \tag{2.14}$$

The exponential map of a matrix $\mathbf{A} \in \mathbb{R}^{3\times 3}$ is analogous to the exponential used for real numbers $a \in \mathbb{R}$. In particular the Taylor expansion of the exponential has the same form:

$$\exp\left(\theta\widehat{\omega}\right) = e^{(\theta\widehat{\omega})} = I + \theta\widehat{\omega} + \frac{\theta^2}{2!}\widehat{\omega}^2 + \frac{\theta^3}{3!}\widehat{\omega}^3 + \ldots \tag{2.15}$$

Exploiting the fact that $(\theta\widehat{\omega})$ is screw symmetric, we can easily compute the exponential of the matrix $\widehat{\omega}$ in closed form using the *Rodriguez formula*:

$$\exp(\theta\widehat{\omega}) = I + \widehat{\omega}\sin(\theta) + \widehat{\omega}^2(1 - \cos(\theta)) \tag{2.16}$$

where only the square of the matrix $\widehat{\omega}$ and sine and cosine of real numbers have to be computed. Note that this formula allows us to reconstruct the rotation matrix from the angle θ and the axis of rotation ω by simple operations and this is probably the main justification of using the axis-angle representation at all.

Exponential Maps for Rigid Body Motions

The exponential map formulation can be extended to represent rigid body motions, namely any motion composed by a rotation $\mathbf{R}$ and a translation $\mathbf{t}$. This is done by extending the parameters $\theta\omega$ with

$\theta v \in \mathbb{R}^3$ which is related to the translation along the axis of rotation and the location of the axis, see Fig. 2.2 This six parameters form the *twist coordinates* $\theta\xi = \theta(v_1,v_2,v_3,\omega_1,\omega_2,\omega_3)$ of a twist. Analogous to Eq. (2.10), any rigid motion $\mathbf{G} \in \mathbb{R}^{4\times 4}$ can be written in exponential form as:

$$\mathbf{G}(\theta,\omega) = \begin{bmatrix} \mathbf{R}_{3\times 3} & \mathbf{t}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} = \exp(\theta\widehat{\xi}) \tag{2.17}$$

where the 4×4 matrix $\theta\widehat{\xi} \in se(3)$ is the *twist action* and is a generalization of the screw symmetric matrix $\theta\widehat{\omega}$ of Eq. (2.11). The twist action is constructed from the twist coordinates $\theta\xi \in \mathbb{R}^6$ using the wedge operator $^\wedge$

$$[\theta\xi]^\wedge = \theta\widehat{\xi} = \theta \begin{bmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.18}$$

and its exponential can be computed using the following formula

$$\exp(\theta\widehat{\xi}) = \begin{bmatrix} \exp(\theta\widehat{\omega}) & (I - \exp(\theta\widehat{\omega}))(\omega \times v + \omega\omega^T v\theta) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.19}$$

with $\exp(\theta\widehat{\omega})$ computed by using the Rodriguez formula Eq. (2.16) as explained before.

The Logarithm

For human tracking it is sometimes needed to obtain the twist parameters given a transformation matrix $\mathbf{G}$. In particular if we want to obtain the resulting twist of two consecutive twists this operation is needed. In [61], a constructive way is given to compute the twist which generates a given transformation matrix $\mathbf{G}$. For the case $\mathbf{R} = \mathbf{I}$, the twist is given by

$$\theta\xi = \theta(0,0,0,\frac{\mathbf{t}}{||\mathbf{t}||}), \qquad \theta = ||\mathbf{t}|| \tag{2.20}$$

For the other cases, the motion velocity θ and the rotation axis ω are given by

$$\theta = \cos^{-1}\left[\tfrac{tr(\mathbf{R})-1}{2}\right], \quad \omega = \frac{1}{2\sin(\theta)} \begin{bmatrix} \mathbf{R}_{32} - \mathbf{R}_{23} \\ \mathbf{R}_{13} - \mathbf{R}_{31} \\ \mathbf{R}_{21} - \mathbf{R}_{12} \end{bmatrix} \tag{2.21}$$

From Eq. (2.19) it follows that to obtain v, the matrix,

$$\mathbf{A} = (I - \exp(\theta\widehat{\omega}))\widehat{\omega} + \omega\omega^T\theta \tag{2.22}$$

needs to be inverted and multiplied with the translation vector $\mathbf{t}$,

$$v = \mathbf{A}^{-1}\mathbf{t} \tag{2.23}$$

We call this transformation from a rigid motion $\mathbf{G} \in SE(3)$ to a twist $\theta\xi \in \mathbb{R}^6$ the logarithm, $\theta\xi = \log(\mathbf{G})$, see [61].

Adjoint Transformation

Given a twist $\xi_b = (v_b,\omega_b) \in \mathbb{R}^6$ with coordinates in the body frame B, we can find the coordinates of the twist in the spatial frame S. Given that the configuration of B relative to frame S is the rigid motion $\mathbf{g} = (\mathbf{R},\mathbf{t}) \in SE(3)$, the twist coordinates in the spatial frame are given by

$$\xi_s = \mathrm{Ad}_{\mathbf{g}}\, \xi_b \qquad \mathrm{Ad}_{\mathbf{g}} = \begin{bmatrix} \mathbf{R} & \mathbf{t}^{\wedge}\mathbf{R} \\ \mathbf{0}_{3\times 3} & \mathbf{R} \end{bmatrix} \tag{2.24}$$

where Ad_g is the *adjoint transformation* associated with g, see [61]. To see this, note that the angular components are related by the rotation $\omega_s = \mathbf{R}\,\omega_b$ (the same way we rotate points we can rotate the axes). From this it follows that $v_s = \mathbf{R}\; v_b \;+\; \mathbf{t}^{\wedge}\,\mathbf{R}\omega_b$. Equivalently, the action $\widehat{\xi}_s$ of a twist with twist coordinates ξ_s is related to the action $\widehat{\xi}_b$ with twist coordinates ξ_b by

$$\widehat{\xi}_s = \mathbf{G}\, \widehat{\xi}_b\, \mathbf{G}^{-1} \tag{2.25}$$

Recall that the product of a twist $\widehat{\xi}$ by a point results in the velocity of the point $\mathbf{v}_{\mathbf{p}}$, see Figure 2.2. Furthermore, *a twist with twist coordinates ξ_a in a given frame A, applies on points $\mathbf{p}_a$ defined in the same frame A and this results in the velocity of the point relative to frame A $\mathbf{v}_{\mathbf{p}_a}$*. Thus, we can interpret Eq. (2.25) the following way: the velocity in spatial coordinates of a point $\mathbf{p}_s$ is obtained by first transforming the point to body coordinates $\mathbf{p}_s \mapsto \mathbf{p}_b$ with $\mathbf{G}^{-1}$, then finding the velocity of the point in body coordinates $\mathbf{v}_{\mathbf{p}_b}$ using the twist action $\widehat{\xi}_b$ and finally transforming the velocity back to spatial coordinates $\mathbf{v}_{\mathbf{p}_s}$ with $\mathbf{G}$. One can prove that indeed this results in the spatial velocity

$$\bar{\mathbf{v}}_{\mathbf{p}_s} = (\mathbf{G}\, \widehat{\xi}_b\, \mathbf{G}^{-1})\; \bar{\mathbf{p}}_s \;=\; (\mathbf{G}\, \widehat{\xi}_b\, \mathbf{G}^{-1})\; \mathbf{G}\; \bar{\mathbf{p}}_b = \mathbf{G}\; \widehat{\xi}_b\; \bar{\mathbf{p}}_b \;=\; \mathbf{G}\; \bar{\mathbf{v}}_{\mathbf{p}_b}, \tag{2.26}$$

where $\bar{\mathbf{v}}_{\mathbf{p}_s} = [\mathbf{v}_{\mathbf{p}_s} 0]$ are the homogeneous coordinates of the vector $\mathbf{v}_{\mathbf{p}_s}$. An interesting property that stems from (2.25) and (2.26) is that $\mathbf{G}$ can be shifted inside the exponential

$$\exp(\widehat{\xi}_s) \;=\; \mathbf{G}\; \exp(\widehat{\xi}_b)\, \mathbf{G}^{-1} \;=\; \exp(\mathbf{G}\; \widehat{\xi}_b\; \mathbf{G}^{-1}) \tag{2.27}$$

which means that to express a rigid body motion $\exp(\xi)$ in another coordinate system we can simply transform the corresponding twist action with $\mathbf{G}$. The same way we can interpret a rigid motion applied to a point as a coordinate transformation or as a relative motion, we can interpret the adjoint transform applied to twists as a transformation that brings a twist from their initial coordinates ξ to their coordinates ξ' (defined in the same frame) after the rigid motion g is applied, see Figure 2.3. Indeed, we will make frequent use of this interpretation in the next sections when we have to keep track of human joints locations and orientations during tracking.

2.1.5 Rotation Derivatives and Rigid Body Velocity

Rotation derivatives do not have a direct physical meaning. Indeed, we will see that it is more meaningful to think about the twist or screw associated with a rigid motion rather than on infinitesimal rotations. Consider a point following a trajectory defined by the rigid motion $\mathbf{G}(\theta,\omega)$

$$\mathbf{p}(\theta,\omega) = \mathbf{G}(\theta,\omega)\bar{\mathbf{p}}(0) \tag{2.28}$$

Given a rigid motion controlled by the parameters θ,ω, it is useful to know the derivatives of a point trajectory w.r.t. the parameters. Suppose, the axis of rotation ω is fixed and therefore the motion is fully described by the parameter θ. In that case, the derivatives would be

$$\frac{\Delta \mathbf{p}(\theta)}{\Delta\theta} = \frac{\Delta \mathbf{G}(\theta)}{\Delta\theta}\bar{\mathbf{p}}(0). \tag{2.29}$$

The quantity $\frac{\Delta \mathbf{G}}{\Delta \theta}$ is a 4×4 matrix and represents an infinitesimal rigid transformation. If the rigid motion is parameterized by more parameters, then the Jacobian would correspond to a tensor of dimensions $4 \times 4 \times D$ where D is the number of parameters. The problem is that the quantity $\frac{\Delta \mathbf{G}}{\Delta \theta}$ does not have any physical meaning. We can obtain a more meaningful derivative by multiplying the right hand side by $\mathbf{G}^{-1}(\theta)\mathbf{G}(\theta)$ since $\mathbf{G}^{-1}(\theta)\mathbf{G}(\theta) = \mathbf{I}$, obtaining

$$\frac{\Delta \bar{\mathbf{p}}(\theta)}{\Delta \theta} = \frac{\Delta \mathbf{G}(\theta)}{\Delta \theta}\mathbf{G}(\theta)^{-1}\mathbf{G}(\theta)\bar{\mathbf{p}}(0) = \frac{\Delta \mathbf{G}(\theta)}{\Delta \theta}\mathbf{G}(\theta)^{-1}\bar{\mathbf{p}}(\theta). \tag{2.30}$$

where the product $\frac{\Delta \mathbf{G}(\theta)}{\Delta \theta}\mathbf{G}(\theta)^{-1} \in se(3)$ is the twist $\hat{\xi}$ with axis of rotation ω.

Given that θ follows a trajectory over time $\theta(t)$, $\frac{\Delta \theta(t)}{\Delta t}\hat{\xi}$ equals the rigid body velocity. However, we will often omit the dependency of time, as for optimization we are generally only interested in knowing how infinitesimal changes in parameter space relate to physical quantities in the world. From Eq. (2.30) it follows that

$$\hat{\xi} = \frac{\Delta \mathbf{G}(\theta)}{\Delta \theta}\mathbf{G}(\theta)^{-1} \tag{2.31}$$

is a matrix that when multiplied by a point in a given configuration θ maps to the tangential space of the point trajectory $\frac{\Delta \mathbf{p}}{\Delta \theta} = \hat{\xi}\Delta\theta\bar{\mathbf{p}}(\theta)$. Since the twist has a physical meaning we will always seek for derivatives that map infinitesimal changes in parameter space to the tangential space of the rigid motion $se(3)$.

2.1.6 A Note on Linearization of Rotational Motion

Linearizing functions that map to $SO(3)$ can often be confusing. Consider the following simple inverse kinematics problem: find the the angle of rotation θ about a fixed axis ω such that the rotating point $\mathbf{p}(\theta)$ is closest to the target point $\mathbf{q}$. Mathematically, this is expressed as

$$\exp(\theta\widehat{\omega})\mathbf{p}(0) = \mathbf{q}. \tag{2.32}$$

This equation can be linearized using a first order Taylor expansion at the current configuration $f(\theta + \Delta\theta) = f(\theta) + \frac{\Delta f(\theta)}{\Delta \theta}\Delta\theta$. Thereafter, we can find the optimal step $\Delta\theta$

$$(\exp(\theta\widehat{\omega}) + \Delta\theta\widehat{\omega}\exp(\theta\widehat{\omega})\,\mathbf{p}(0) = (\mathbf{I} + \Delta\theta\widehat{\omega})\mathbf{p}(\theta) = \mathbf{q} \tag{2.33}$$

An alternative way formulation that leads to exactly the same linear equations is to exploit the fact that $f(\theta + \Delta\theta) = \exp(\Delta\theta\omega)f(\theta) = (\mathbf{I} + \Delta\theta\omega)f(\theta)$. That implies that given an estimate of θ we can rotate the point in a rest configuration by θ degrees obtaining $\mathbf{p}(\theta) = \exp(\theta\omega)\mathbf{p}(0)$. Given this estimate we can iteratively find the infinitesimal rotation that will bring the point in the current configuration $\mathbf{p}(\theta)$ closest to the target. This can be written as

$$\exp(\Delta\theta\widehat{\omega})\exp(\theta\widehat{\omega})\mathbf{p}(0) = \exp(\Delta\theta\widehat{\omega})\mathbf{p}(\theta) = \mathbf{q} \tag{2.34}$$

where the infinitesimal rotation can be approximated with the Taylor expansion around the origin

$$(\mathbf{I} + \Delta\theta\widehat{\omega})\mathbf{p}(\theta) = \mathbf{q}. \tag{2.35}$$

Note that both derivations lead to the same equations in Eq. (2.33),(2.33). This implies that at any given point during optimization we can update points $\mathbf{p} \mapsto \mathbf{p}(\theta)$ in the model and solve by linearizing around the current solution.

2.1.7 Metrics on SO(3)

For the problem of human pose estimation often we need to evaluate the distance between rotation matrices. Since the distance should be invariant to the chosen coordinate frame we are mostly interested in *bi-invariant* distances. A distance on $SO(3)$, $d_{SO(3)} : SO(3) \times SO(3) \mapsto \mathbb{R}^+$ is said to be invariant when $d_{SO(3)}(\mathbf{R}_1,\mathbf{R}_2) = d_{SO(3)}(\mathbf{S}\mathbf{R}_1,\mathbf{S}\mathbf{R}_2)$, where $\mathbf{R}_1$,$\mathbf{R}_2$,$\mathbf{S}$ are elements of $SO(3)$. In the context of rotations in $SO(3)$ the natural geodesic metric is equal to the angle between two rotations, $\mathbf{R}_1$ and $\mathbf{R}_2$. As we have seen already in Sec. 2.1.4 a rotation can be expressed as an axis ω and an angle θ. In this context the *angular distance* is

$$d_{\text{geo}}(\mathbf{R}_1,\mathbf{R}_2) = \| \log(\mathbf{R}_1\mathbf{R}_2^T)\| = \theta \tag{2.36}$$

where θ is the rotation angle between the relative rotation $\mathbf{R}_1\mathbf{R}_2^T$. Another commonly used metric is the *chordal distance*, which consists of the Euclidean distance between them in the embedding space $\mathbb{R}^{3\times3} = \mathbb{R}^9$

$$d_{\text{chord}}(\mathbf{R}_1,\mathbf{R}_2) = \|\mathbf{R}_1 - \mathbf{R}_2\|_F \tag{2.37}$$

where $\|\cdot\|_F$ represents the Frobenius norm of the matrix. This distance can easily be related to the angular distance using Rodriguez formula 2.16. Specifically, let $\mathbf{R}_1\mathbf{R}_2^T = \exp(\theta\hat{\omega})$. Exploiting that $\|\hat{\omega}\|_F^2 = \|\hat{\omega}^2\|_F^2 = 2$ and the fact that $\hat{\omega}$ and $\hat{\omega}^2$ are orthogonal w.r.t to the Frobenius norm we have

$$\begin{aligned}
d^2_{\text{chord}}(\mathbf{R}_1,\mathbf{R}_2) &= \|\mathbf{R}_1 - \mathbf{R}_2\|_F^2 = \|\mathbf{R}_1\mathbf{R}_2^T - \mathbf{I})\|_F^2 && (2.38)\\
&= \|\mathbf{I} + \sin(\theta)\hat{\omega} + (1-\cos(\theta))\hat{\omega}^2 - \mathbf{I}\|_F^2 && (2.39)\\
&= 2(\sin^2(\theta) + (1-\cos(\theta))^2) && (2.40)\\
&= 8\sin^2(\theta) && (2.41)
\end{aligned}$$

from which we get the relationship

$$d_{\text{chord}} = 2\sqrt{2}\sin(\theta/2). \tag{2.42}$$

Another important metric is the *quaternion distance*. Let $\mathbf{q}_1$ and $\mathbf{q}_2$ be the quaternion representation of two rotations, then the distance is defined as the Euclidean distance between them

$$d_{\text{quat}} = \|\mathbf{q}_1 - \mathbf{q}_2\| \tag{2.43}$$

The quaternion distance can also be easily related to the angular distance. Let $\mathbf{q}_{\text{id}} = (1,0,0,0)^T$ be the identity quaternion, and $\mathbf{q}_r = \mathbf{q}_1 \circ \mathbf{q}{-}1_2$ the relative rotation with rotation angle θ. Then we can write

$$\begin{aligned}
d^2_{\text{quat}}(\mathbf{q}_1,\mathbf{q}_2) &= d^2_{\text{quat}}(\mathbf{q}_r,\mathbf{q}_{\text{id}}) && (2.44)\\
&= \|\mathbf{q}_r\|^2 - 2\langle\mathbf{q}_r,\mathbf{q}_{\text{id}}\rangle + \|\mathbf{q}_{\text{id}}\|^2 && (2.45)\\
&= 2(1-\cos(\theta/2)) && (2.46)
\end{aligned}$$

and using the identity $\cos(\theta/2) = 1 - 2\sin^2(\theta/4)$ we obtain the relationship

$$d_{\text{quat}} = 2\sin(\theta/4). \tag{2.47}$$

Another distance often used is the scalar product between quaternions which equals $\cos(\theta/2)$. This can be easily seen since $\langle\mathbf{q}_1,\mathbf{q}_2\rangle = \langle\mathbf{q}_r,\mathbf{q}_{\text{id}}\rangle = \cos(\theta/2)$

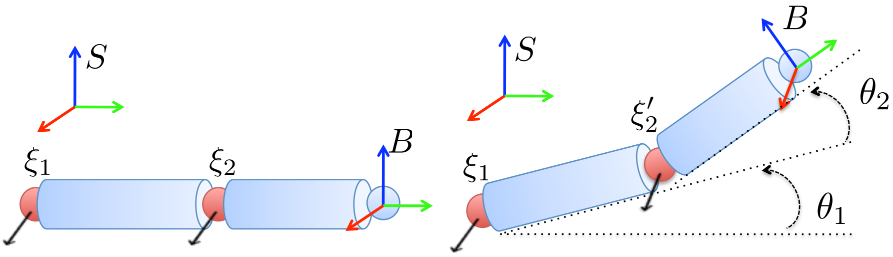


Figure 2.3: Kinematic chain: the motion is given by the concatenation of joint angular motions. Note how the twists ξ are transformed to ξ' when parent articulations move

2.1.8 Kinematic Chains

Human motion is articulated and we want to model the motion taking all the joints into account at the same time. For example, consider the motion of the hand, this motion will be the concatenation of motions of their parent joints, wrist, elbow, shoulder and root. To formulate this we now define two coordinate frames, the spatial frame S, which is usually fixed and the body frame B, which is the coordinate system attached to the segment of interest. Note that the body frame moves along with the segment and therefore a control point in the segment in body coordinates $\mathbf{p}_b$ is always constant. Consider the arm in Figure 2.3 with two segments and only two degrees of freedom. To obtain the coordinates of a control point in the hand in spatial coordinates $\mathbf{p}_s$ from their body coordinates $\mathbf{p}_b$ we can concatenate the rigid motions along the chain:

$$\bar{\mathbf{p}}_s = \mathbf{G}_{sb}\bar{\mathbf{p}}_b = \mathbf{G}_1\mathbf{G}_2\mathbf{G}_{sb}(\mathbf{0})\bar{\mathbf{p}}_b \tag{2.48}$$

where $\mathbf{G}_1$,$\mathbf{G}_2$ are the rigid motion matrices of the upper and lower arm respectively and $\mathbf{G}_{sb}(\mathbf{0})$ is the transformation between B and S at the zero pose. By using the fact that the motion of each individual joint is generated by a twist associated with a joint axis (see Figure 2.3) we can write the spatial coordinates of a point in the body as a function of the joint angles in the chain:

$$\bar{\mathbf{p}}_s = \mathbf{G}_{sb}(\theta_1,\theta_2) = e^{\widehat{\xi}_1\theta_1}e^{\widehat{\xi}_2\theta_2}\mathbf{G}_{sb}(\mathbf{0})\bar{\mathbf{p}}_b \tag{2.49}$$

Any new articulation joint will represent an additional twist in the chain, see Figure 2.3. If we generalize this procedure for any limb of the human body we can define what is known in the robotics literature as the forward kinematics map. The forward kinematics is then defined as the mapping between the vector of joint angles $\Theta = (\theta_1,\theta_2,\ldots,\theta_n)^T$ to the transformation matrix between the spatial and body frames $\mathbf{G}_{sb}$. If we define Q as the space of joint angle vectors, then the forward kinematics $\mathbf{G}_{sb} : Q \to SE(3)$ is given by:

$$\mathbf{G}_{sb}(\Theta) = e^{\widehat{\xi}_1\theta_1}e^{\widehat{\xi}_2\theta_2}\ldots e^{\widehat{\xi}_n\theta_n}\mathbf{G}_{sb}(\mathbf{0}) \tag{2.50}$$

where ξ are constant twists in the reference configuration, *i.e.*, the starting *zero pose*. For human tracking it is usual to take $\mathbf{G}_{sb}(\mathbf{0})$ to be the identity, *i.e.*, the body and spatial frame are coincident at the beginning for every single limb on the human body.

The Articulated Jacobian

The articulated Jacobian[2] is a matrix $\mathbf{J}(\Theta) \in \mathbb{R}^{6\times n}$ that maps joint velocities to a rigid body motion velocity $\mathbf{J}(\Theta) : \mathbb{R}^{\mathbb{D}} \mapsto se(3)$ represented by a twist and it may be written as

$$\mathbf{J}_\Theta = [\xi_1 \quad \xi'_2 \quad \ldots \quad \xi'_n] \tag{2.51}$$

---

[2] We call it articulated Jacobian and not *manipulator Jacobian* as in Murray *et al.* [61] because we find it more appropriate in this context

where $\xi'_i = \mathrm{Ad}_{\left(e^{\widehat{\xi}_1\theta_1}\ldots e^{\widehat{\xi}_{i-1}\theta_{i-1}}\right)}\xi_i$ is the i-th joint twist transformed to the current pose, Figure 2.3. To obtain ξ'_i an option is to update at every time step the twists with the accumulated motion of parent joints in the chain. Note that the form of the Jacobian is different for every limb in the body since different body parts are influenced by different joints. Now given a pose determined by Θ and point in the body in spatial coordinates $\mathbf{p}_s$ we can obtain the increment $\Delta\mathbf{p}_s$ in position as a function of the increment in parameter space $\Delta\Theta$ as

$$\Delta\bar{\mathbf{p}}_s = [\ \mathbf{J}_\Theta\ \cdot \Delta\Theta\ ]^\wedge\ \bar{\mathbf{p}}_s = [\xi_1\Delta\theta_1 + \xi'_2\Delta\theta_2 + \ldots + \xi'_n\Delta\theta_n]^\wedge\bar{\mathbf{p}}_s \tag{2.52}$$

where $^\wedge$ is the wedge operator defined in Eq. (2.18) and we drop the homogeneous component after the multiplication of $[\ \mathbf{J}_\Theta\ \cdot \Delta\Theta\ ]^\wedge\ \bar{\mathbf{p}}_s$. We can interpret the formula as follows: the total displacement of the point $\mathbf{p}_s$ is the sum of individual displacements generated by the angle increments $\Delta\theta_i$ in upper joints keeping the others fixed. It is very important to note that the result of $[\ \mathbf{J}_\Theta\ \cdot \Delta\Theta\ ]$ are the twist coordinates ξ_s of the rotational ω and "linear velocity" v of the body expressed in the spatial frame. Note that the product $\widehat{\xi}_s\bar{\mathbf{p}}_s$ results in the point increment in homogeneous coordinates $\Delta\bar{\mathbf{p}}_s = \begin{bmatrix}\Delta\mathbf{p}_s & 0\end{bmatrix}$. Since we are not interested in the last homogeneous component, in the following we will confuse $\Delta\bar{\mathbf{p}}_s$ with $\Delta\mathbf{p}_s$ by dropping the homogeneous component after the multiplication $\widehat{\xi}_s\mathbf{p}_s$.

2.1.9 Human Pose Parametrization

Now we have the necessary mathematical tools to model all the joints in the human model. We identify three kinds of joints in the human body according the *DoF*(*degrees of freedom*, see Table 2.1. The *root joint* that determines the overall orientation and position of the body has 6 *DoF* and can be modeled as a twist $\theta\xi$ with the six components as free parameters. *Ball joints* capable of any rotation with no translation can be efficiently modeled as twist with known joint location $\mathbf{q}$ and unknown axis of rotation $\theta\omega$ [52]. Finally, simple *revolute joints* are only capable of rotations about a fixed known axis. For revolute joints the twist is constant and is given by

$$\xi = \begin{bmatrix} -\omega \times \mathbf{q} \\ \omega \end{bmatrix} \tag{2.53}$$

and the only unknown is the rotation angle θ, see Figure 2.2 for a geometrical interpretation. This last category is very convenient to constrain the motion of 1 *DoF* joints (*e.g.*, the knee). In the literature the common choice is to model the root joint with six free parameters and to model all the other joints with the concatenation of revolute joints. A saddle joint is modeled as the concatenation of two one *DoF* joints. A ball joint can be modeled by 3 consecutive revolute joints, *i.e.*, 3 consecutive rotations about 3 fixed axes. The free parameters are then the angles about each of the 3 axes θ_1,θ_2,θ_3. This parametrization is very similar to the Euler angles and has the same limitations in terms of singularities, *i.e.*, it is not free of gimbal lock. However, to keep the notation simple, in the following we assume that we parametrize ball joints as 3 consecutive revolute joints. For a description of the parametrization of ball joints using a single free axis of rotation we refer the reader to [52].

Table 2.1: Table of existing joints to model human motion

Joint	DoF	Unknown parameter	Example
Root	6	$\xi = \theta[v\ \omega]^T$	All body
Ball	3	$\theta\omega$	Hips
Saddle	2	θ_1,θ_2	Wrist
Revolute	1	θ	Knee

Therefore, the pose configuration of the human is usually encoded with a scaled twist ξ for the root joint and a vector of n joint angles for the rest of the joints. Let us denote the state vector of pose parameters at time t, as

$$\mathbf{x}_t := (\xi, \Theta) \qquad \Theta := (\theta_1 \, \theta_2 \, \dots \, \theta_n). \tag{2.54}$$

Thereby, a human pose is totally determined by a D-dimensional state vector $\mathbf{x}_t \in \mathbb{R}^D$, with $D = 6 + n$.

The Pose Jacobian

For local optimization it is necessary to know the relationship between increments in the pose parameters and increments in the position of a point in a body segment. This relationship is given by the pose Jacobian $\mathbf{J}_p(\mathbf{x}; \mathbf{p}_s) = \frac{\Delta \mathbf{p}_s}{\Delta \mathbf{x}}$. In this paragraph, we derive the analytical expression for the pose Jacobian. We start our derivation from the expression of the point increment of Eq. (2.52). Let us denote with $\Delta\xi = [\ \Delta v_1 \ \Delta v_2 \ \Delta v_3 \ \Delta\omega_1 \ \Delta\omega_2 \ \Delta\omega_3\ ]$ the relative twist corresponding to the root joint. The six coordinates of the scaled relative twist $\Delta\xi$ are now free parameters we will want to estimate. By using the following identity $[u + w]^\wedge = \widehat{u} + \widehat{w}$ we can rewrite equation Eq. (2.52) as increments in pose parameter space

$$\begin{aligned} \Delta \mathbf{p}_s &= [\Delta\xi + \xi_1' \Delta\theta_1 + \ldots + \xi_n' \Delta\theta_n]^\wedge \bar{\mathbf{p}}_s \\ &= \widehat{\Delta\xi}\, \bar{\mathbf{p}}_s + \widehat{\xi_1'}\, \bar{\mathbf{p}}_s \, \Delta\theta_1 + \ldots + \widehat{\xi_n'}\, \bar{\mathbf{p}}_s \, \Delta\theta_n \end{aligned} \tag{2.55}$$

where we can isolate the parameters of the root joint $\Delta\xi$ rewriting $\widehat{\Delta\xi}\, \bar{\mathbf{p}}_s$

$$\widehat{\Delta\xi}\, \bar{\mathbf{p}}_s = \Delta v + \Delta\omega \times \mathbf{p}_s = \Delta\, v - \mathbf{p}_s^\wedge \, \Delta\, \omega = \left[\mathbf{I}_{[3\times3]} \quad | \quad -\mathbf{p}_s^\wedge \right] \Delta\xi \tag{2.56}$$

and substituting this expression in equation Eq. (2.55) again

$$\begin{aligned} \Delta \mathbf{p}_s &= \left[\mathbf{I}_{[3\times3]} \quad | \quad -\mathbf{p}_s^\wedge \right] \Delta\xi + \widehat{\xi_2'}\, \bar{\mathbf{p}}_s \, \Delta\theta_2 + \ldots + \widehat{\xi_n'}\, \bar{\mathbf{p}}_s \, \Delta\theta_n \\ &= \mathbf{J}_p(\mathbf{x}; \mathbf{p}_s)\, \Delta\mathbf{x}, \end{aligned} \tag{2.57}$$

where $\Delta\mathbf{x} = [\ \Delta\xi \quad \Delta\Theta\ ]$ is the differential vector of pose parameters and

$$\mathbf{J}_p(\mathbf{x}; \mathbf{p}_s) = \left[\begin{matrix} \mathbf{I}_{[3\times3]} & -\mathbf{p}_s^\wedge & \widehat{\xi_1}\, \bar{\mathbf{p}}_s & \widehat{\xi_2'}\, \bar{\mathbf{p}}_s & \ldots & \widehat{\xi_n'}\, \bar{\mathbf{p}}_s \end{matrix} \right] \tag{2.58}$$

is the positional Jacobian $\mathbf{J}_p(\mathbf{x}; \mathbf{p}_s) \in \mathbb{R}^{3\times D}$ of a point $\mathbf{p}_s$ with respect to the pose parameters which we denote as *pose Jacobian*. For a given point in the body $\mathbf{p}_s$ in a configuration $\mathbf{x}$, $\mathbf{J}_p(\mathbf{x}; \mathbf{p}_s) : \mathbb{R}^D \mapsto \mathbb{R}^3$ maps an increment of the pose parameters $\Delta\mathbf{x}$ to a positional increment of the point $\Delta\mathbf{p}_s$. We identify two main blocks in the pose Jacobian: the first 6 columns that correspond to the non constant relative twist $\Delta\xi$ of the root joint, and the rest of the columns (*joint columns*) that correspond to the point velocity contribution of each joint angle. Consequently, the column entries of joints that are not parents of the point are simply zero $\mathbf{0}_{3\times1}$. The analytical pose Jacobian derived here is general and will appear in every local optimization method using the parametrization described in Eq. (2.54).

2.2 Other Pose Representations

Although the kinematic representation and joint angle parametrization are the most common for 3D human pose estimation, other parameterizations have been used in the literature. There is no consensus on what is the best representation of poses since each one comes with advantages and disadvantages.

2.2.1 Part based

Part based parameterizations are a collection of *parts* $\mathbf{l}_1$ that are loosely connected. The pose is thus described by a collection of parts $\mathbf{x}_{\text{parts}} := (\mathbf{l}_1 \dots \mathbf{l}_n)$. Each part $\mathbf{l}_i$ is parameterized independently usually by position, scale and rotation parameters. In this way, part is defined as $\mathbf{l}_i = [x_i,y_i,s_i,\theta_i]^T$. The kinematic constraints are enforced using soft constraints or spring forces between connected parts. Typically deformation potentials are used to penalize strong deviations in position and scale between connected parts. Typically an energy function such as

$$p(\mathbf{x}_{\text{parts}}) \propto \sum_{i=1}^{L} E_{\text{image}}(\mathbf{I},\mathbf{l}_i) + \sum_{\forall i,j \in E}^{L} E_{\text{deformation}}(\mathbf{l}_i,\mathbf{l}_j) \tag{2.59}$$

is minimized, where E_{image} is the likelihood term, $E_{\text{deformation}}$ is the deformation potential, and E is the set of edges in the kinematic tree. Note that this allows one to evaluate the likelihood term of each part independently of the others, thus reducing the expensive combinatorial search. The main advantage of this formulation is that inference can be performed in closed form efficiently using DP. Furthermore, computing marginals probabilities on any node in the graph conditioned any observed nodes can be efficiently performed using belief propagation or sum product. This allows to efficiently sample poses conditioned on any observed nodes. For a more detailed description on part based models we refer the reader to [64]. Examples of part based representations of pose can be seen in Fig. 2.4.

2.2.2 Joint Positions

Another obvious representation is given by a collection of joint coordinates,
$\mathbf{x} = [\mathbf{p}_1, \dots \mathbf{p}_n]$, where $\mathbf{p}_j = [x_j,y_j,z_j]^T$ are the coordinates of the j-th joint. Presumably the advantage is that it allows one to work on the Euclidean space as opposed to working on the joint angle space. However, the kinematic constraints such as bone-length preservation and connected parts have to be imposed during optimization. At least two constraints need to be integrated, namely

- *Bone length preservation*: mathematically this is takes the form $\|\mathbf{p}_i - \mathbf{p}_j\| = l_{\text{bone}}$ where l_{bone} denotes bone length, and $\mathbf{p}_i,\mathbf{p}_j$ belong to the same rigid part. At least one of this constraints per rigid bone in the body needs to be included in the optimization problem.
- *Articulated constraints*: these are integrated as a penalty for deviation of adjacent joints, *i.e.*, $E_{\text{penalty}} = \|\mathbf{p}_i - \mathbf{p}_j\|_p, \forall i,j \in E$, where E is the connectivity graph which is typically a tree.

Both types of constraints are more commonly included as soft constraints to allow for non-rigid motion. This has the advantage of allowing limbs to be loosely connected. The main disadvantage is that a strategy is needed to satisfy the constraints during optimization which might be difficult in practice. By contrast, using a kinematic chain representation such constraints are naturally fulfilled. A joint position representation has been commonly used in discriminative methods because presumably prediction does not involve so many non-linearities.

2.2.3 Posebits

Humans rarely describe the pose of a person in terms of joint angles or absolute joint positions. Typically, we describe a pose in terms of relative positions between parts. For example we might say (*e.g.* left-leg in front of right-leg or hands close to each other, hands above the head). We argue that the pose might be well represented by a set of such descriptions. Such representation will be explained in detail in Chap. 6 where we introduce the concept of posebit. In contrast to numerical representations of pose, the posebit representation is semantic, which makes it more intuitive for

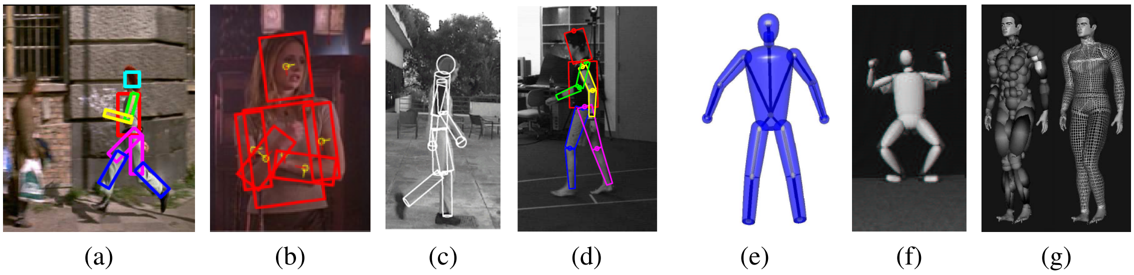


Figure 2.4: Models for human pose estimation can be classified according to the level of detail: in 2D people usually use a set of connected rectangles (a)(b), in 3D a variety of geometric primitives have been used, from cylinders (c)(d) to ellipsoids (e)(f) or Gaussian Blobs (g). From left to right images courtesy of [65][46][66][12][67][68][69]

humans. Indeed, some poses might be perceived as similar by humans while having very distinct numerical representations. Some of the advantages of such description are: annotation of training data is easier and allows to cluster poses in semantic classes.

2.3 Model Creation

To track and estimate the pose of humans in images, one needs a model of the person shape. The goal is then to fit the model of shape to the image observations, typically by maximizing the likelihood of the image observations given a pose. The representations of pose explained in the previous section determine how the model can move. It is obvious that having a realistic representation of the human shape is a crucial step in the pose estimation pipeline. This modeling involves the initialization of the 3D surface mesh and the skeletal kinematic structure. We can roughly classify the approaches for shape initialization according to the level of detail. We find three main classes, methods that approximate the human body using *geometric primitives*, methods that use a subject specific *body scan* to build a 3D mesh model and finally methods that estimate *detailed shape from images* without a body scan of the subject.

2.3.1 Geometric Primitives

A wide variety of geometric primitives have been used to approximate the body shape. Early works used a simplified body model-based on a collection of articulated planar rectangles [70]. More sophisticated models have used cylinders [12], truncated cones, ellipsoids [68] or Gaussian blobs [71]. These geometric primitives can then be parametrized using very few numbers *e.g.*, the shape of the cylinders is encoded as the height and radius. Thereby, if not initialized manually, the vector of shape parameters ϕ is estimated from images in a calibration step. The parameters include internal proportions, limb lengths and volumes.

2.3.2 Detailed Body Scans

Whole-body 3D scans provide a very accurate measurement of the surface shape. However, the model creation from a 3D scan is more involved than using simple geometric primitives. The output from a 3D scans is usually a dense 3D point cloud and a triangulated mesh. However, the triangulated mesh contains holes due to self occlusions. To initialize the model for tracking three main pre-processing steps are needed, (i) fit a template mesh to the 3D point cloud of the scanner, (ii) create a skeleton and

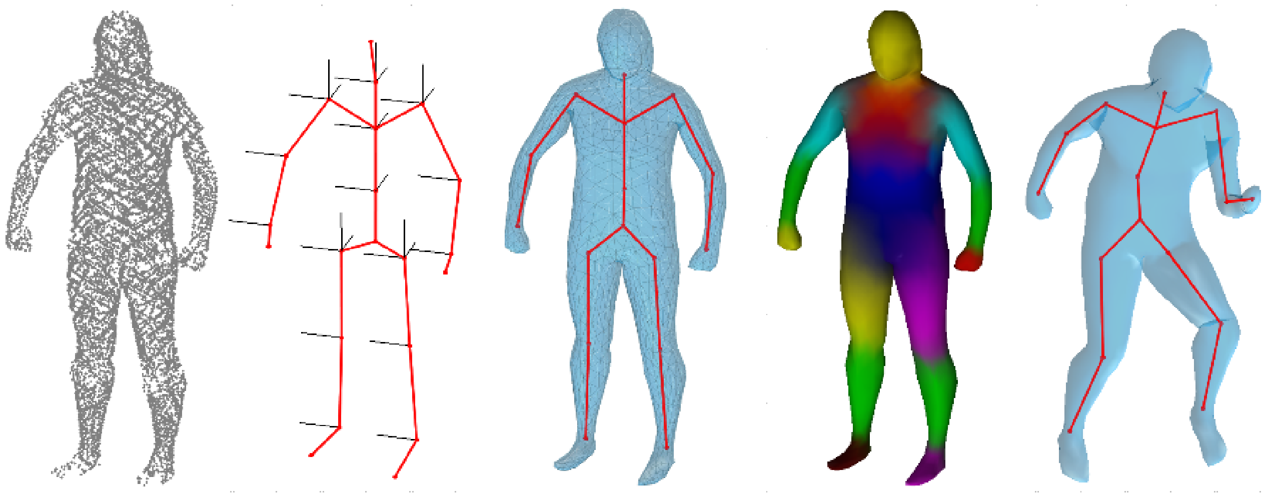


Figure 2.5: Processing pipeline for model rigging from a body scan; from left to right: body scan surface, down-sampled 3D point cloud, skeleton with the twist axis orientations in black, registered template mesh, skinned model and animated model

(iii) bind skin to the skeleton bones. The last is known as skinning and the whole process is known as rigging.

- *Template mesh registration:* Since the triangulated mesh from the laser scan contains holes, a template mesh has to be morphed to fit the point cloud. This can be done with standard non-rigid registration techniques [72, 73]. Current non-rigid registration techniques require a set corresponding control points between the template mesh and the scan. The correspondences can be obtained, for example, with the *Correlated Correspondence* technique which matches similar looking surface regions while minimizing the deformation [74]. Given the correspondences non-rigid registration is used to fit the template to the scan.

- *Skeleton fitting:* The skeleton determines the motion of the model. For the creation of the skeleton we must choose the number of joint articulations and the degrees of freedom for every joint. Rough human models use only 15 degrees of freedom while models used for movie production contain over 60. For human pose estimation from images many researchers use between 20-30 *DoF* [68, 75, 5, 76, 8, 77] which gives a good compromise between degree of realism and robustness. For every joint we must determine two things: the location and the orientation of the axis of rotation ω, see Figure 2.5. The skeleton is usually edited manually before tracking.

- *Skinning:* Given the registered template mesh and the skeleton we have to determine for every vertex in the surface to which body part it belongs, *i.e.*we must assign a joint index to every vertex. For realistic animations, however, the representation of human motion as rigid parts is too simplistic, specially for regions close to the articulations. To obtain a smooth deformation an option is to use *linear blend skinning* [78], which approximates the motion of points close to a joint by a weighted linear combination of neighboring joints. For example the motion of the shoulder vertices would be given by a combination of the torso and arm motions. The motion of a point $\mathbf{p}_s(0)$ in the reference pose is then given by

$$\bar{\mathbf{p}}_s = \sum_{i \in \mathcal{N}} w_i \, \mathbf{G}^i_{sb}(\mathbf{x}_t) \, \bar{\mathbf{p}}_s(0) \tag{2.60}$$

where $i \in \mathcal{N}$ are the indicesinid of their neighboring joints, and w_i are the weights. A simple rule to set the weights is to make them inversely proportional to the distance to neighboring

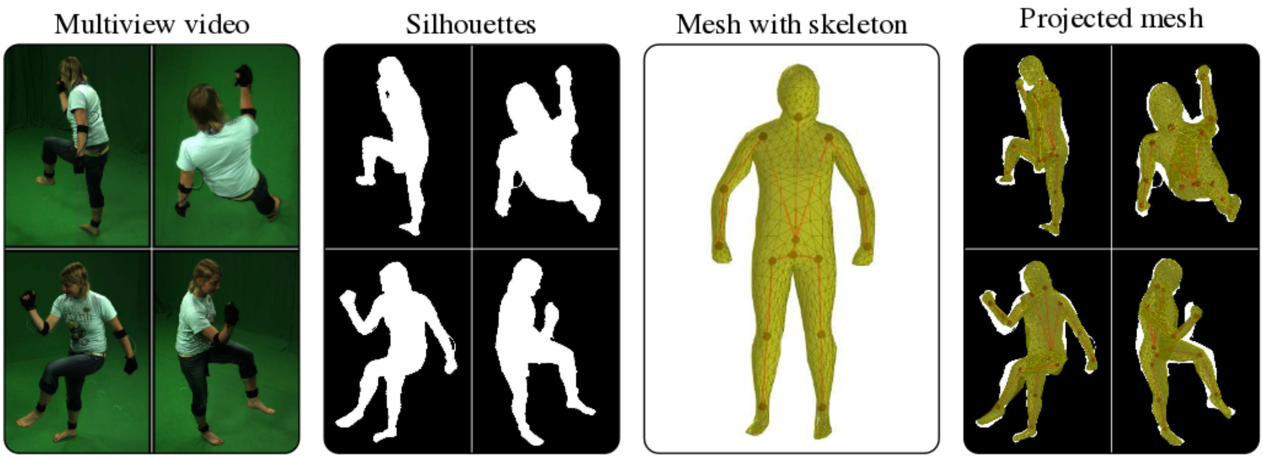


Figure 2.6: Features are extracted from multiview images. In this example we show how silhouettes are obtained using background subtraction. A mesh model is used to synthesize the extracted features. One seeks for the pose parameters that best explain the image evidence.

joint locations $w_i = 1/d_i$. However, this produces severe artifacts. Several algorithms from the graphics community attempt to solve the skinning problem. As a matter of fact, open source software is available to compute the weights given a mesh and a skeleton [79]. Nevertheless, to keep notation simple, throughout this chapter we assume each point is assigned a single joint with weight equal one. We want to emphasize, however, that linear blend skinning *does not change* the formulation on kinematic chains described in the previous section since it is based on linear combinations of rigid motions.

The whole pipeline for mesh registration and rigging is shown in Figure 2.5.

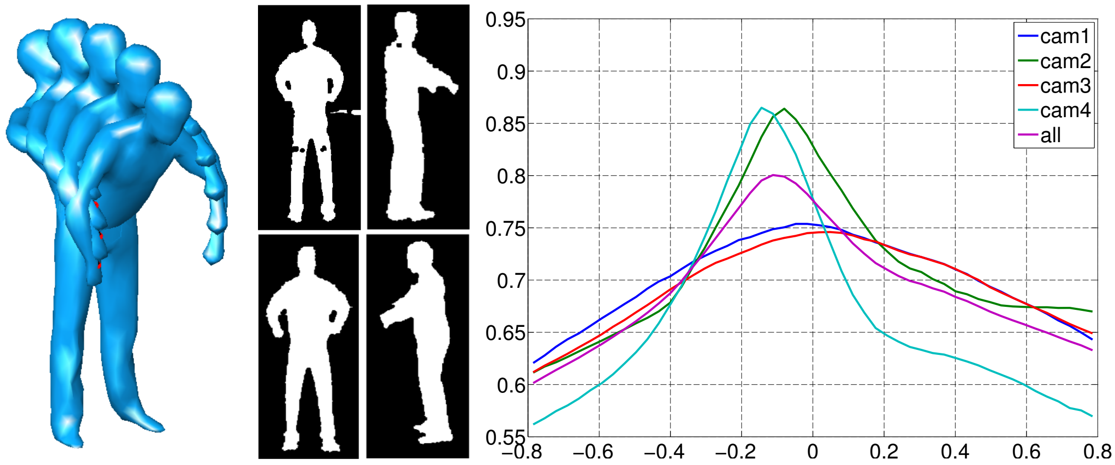
2.3.3 Detailed Shape from Images

Body scan models are limited by the availability of range scanners. To overcome this limitation a recent research direction has focused on the estimation of detailed shape from images [9, 80]. This is achieved by finding parametrization learned from a database of human scans that encodes human shape and pose variation across individuals [81, 82, 83]. All subjects in the database are scanned in different poses to account for both shape and pose deformation. The pose is usually encoded by a combination of rigid and non rigid deformations, and the shape variation is modeled with a set of PCA (Principal component analysis) coefficients learned from the database.

As a final comment, there exist approaches that use neither a skeleton nor shape knowledge from a database [84, 85]. In contrast, such approaches directly deform the mesh geometry by non rigid deformation to fit a set of multiview silhouettes. While impressive results are achieved with such methods, high quality silhouettes are needed and at least 8 cameras are used.

2.4 Likelihood function

One of the key ingredients of a generative tracker is the image likelihood function. Typically, a set of observations will be extracted from the image. Given a predictive model the likelihood measures the consistency between the predictions and the actual observations. Ideally, a truly generative approach should model all the intricacies that generate the image of the moving person: this includes lighting, appearance, shape, pose, shadows etc. Intuitively, the higher the level of detail the higher the fidelity of the likelihood. This should result in less multimodality and a narrower peak at the optimal pose.



(a) (b) (c)

Figure 2.7: Likelihood by varying the torso angle. In (a) we show the sampled poses. For each of this poses we compute the likelihood in each of the four available cameras (b). The landscape of the likelihood can be seen in (c) The vertical axis corresponds to the (unnormalized) likelihood values and the horizontal axis corresponds to the torso angle. Although there exists a clear mode regardless of the camera used one can see that the profile views, (camera 1 and camera 2), results in peakier likelihoods.

Unfortunately, this is both difficult and computationally expensive. For this reason, researchers typically model observations that are both easy to extract from the image and easy to synthesize given a model [12, 65, 8, 49, 25, 86, 18, 87, 88, 24, 66].

2.4.1 Image based cues

Appearance

If the camera is in a fixed location and the scene is relatively static, then it is reasonable to build a background model $\mathbf{B}(x,y)$ of the scene, where (x,y) denote the pixel location in the image. This can then be substracted from the image and thresholded to obtain the foreground region or human silhouette $\mathbf{S}(x,y)$. $\mathbf{S}(x,y) = 1$ if $|\mathbf{I}(x,y) - \mathbf{B}(x,y)| > \epsilon$ and $\mathbf{S}(x,y) = 0$ otherwise. Given a silhouette and a model template $\mathbf{T}(x,y|\mathbf{x})$ generated by projecting the model geometry in a given pose $\mathbf{x}$ to the image (see Fig. 2.6), the likelihood can be formulated as an overlap measure such as

$$p(\mathbf{S}|\mathbf{x}) = \frac{1}{C} \prod_{(x,y)} \exp(-|\mathbf{S}(x,y) - \mathbf{T}(x,y|\mathbf{x})|) \tag{2.61}$$

where C is a normalizing constant. In Figs. 2.7 and 2.8, we show the log-likelihood $p(\mathbf{S}|\mathbf{x})$ by varying the torso angle and the arm angle. We show the likelihood values using four different cameras and all of them. As one can see, the shape of the likelihood strongly depends on the motion relative to the camera. Motions in the direction facing the camera produce little variations in the likelihood. That is due to the intrinsic ambiguities of the images. One can observe for example that the likelihood landscape is much flatter varying the arm Fig. 2.7 than when varying the torso Fig. 2.7. Flatter likelihoods will obviously be more difficult to optimize. In practice, it may be difficult to find a satisfactory threshold ϵ value. To overcome this limitation one can assume background pixels are corrupted with zero-mean additive Gaussian noise. This yields the following likelihood function

$$p(\mathbf{I}|\mathbf{x}) = \prod_{(x,y)} p_B(I(x,y))^{1-\mathbf{T}(x,y|\mathbf{x})}, \tag{2.62}$$

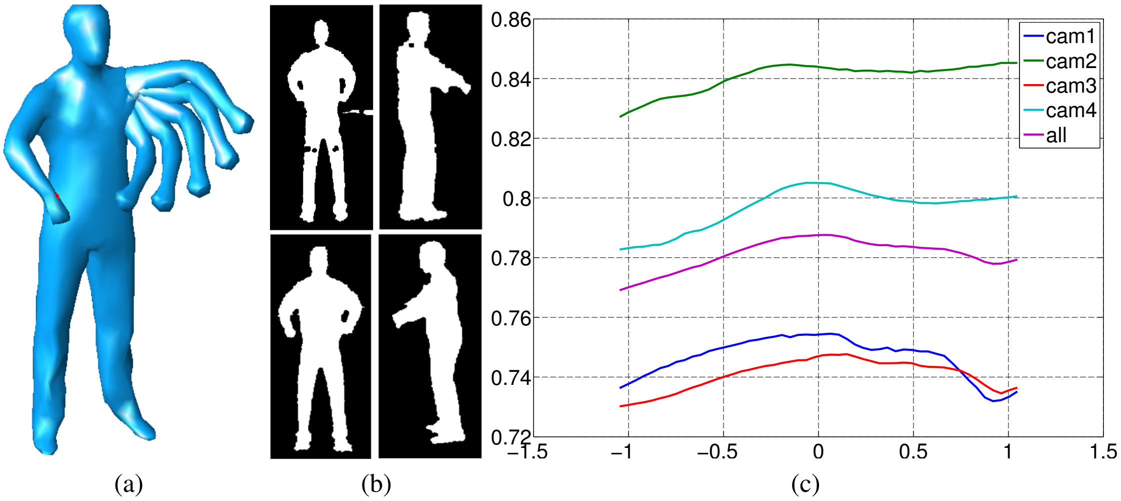


Figure 2.8: Likelihood by varying the arm angle. In (a) we show the sampled poses. For each of this poses we compute the likelihood in each of the four available cameras (b). The landscape of the likelihood can be seen in (c). (c) The vertical axis corresponds to the (unnormalized) likelihood values and the horizontal axis corresponds to the arm angle. This reflects why the arms are usually the most difficult part to estimate, because the likelihood typically does not have clear modes in that direction of parameter space.

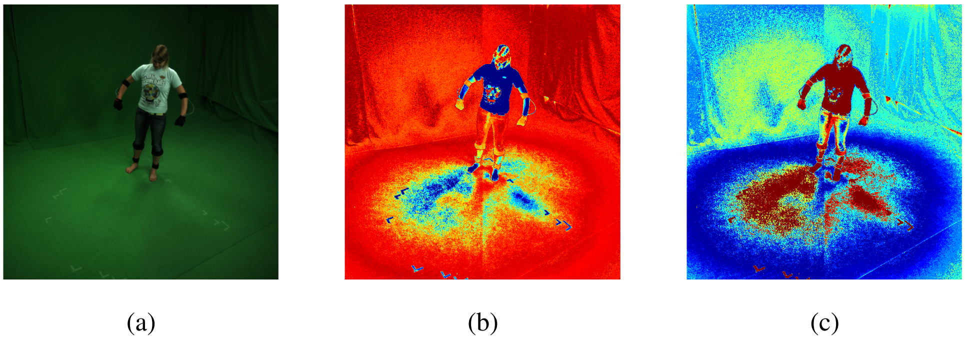


Figure 2.9: Log-likelihood using a background model (b). Hotter colors correspond to more likely values, *i.e.*, in that case more likely to belong to the foreground. In (c) we show the log-likelihood ratio when using a foreground and background model. In that case, hotter colors correspond to pixels where the foreground is significantly better explained by the foreground model than by the background model, *i.e.*, the log-odds. Explicitly modeling of the foreground helps when the foreground and the background have similar appearances. This can be observed in the region of the trousers that are better separated when modeling the foreground as well [89]

where $p_B(I(x,y)$ is the probability of a pixel $\mathbf{I}(x,y)$ belonging to the background. To model p_B a one can model every background pixel independently with a Gaussian. Alternatively, one could use a mixture of Gaussian model or a non parametric histogram to model the background intensities of the background, see Fig 2.9. In our experience, if the background is relatively static, modeling the statistics of each pixel independently works much better than building a single model for all background pixels. For a review of background subtraction methods we refer the interested reader to [90]. When parts of the foreground appearance are similar to the background, then such models penalize severely that part. In such situations it is beneficial to also model the foreground model. This leads to an appearance model for the foreground p_F. If the body figure is homogeneous in appearance one can use a single model for all the body. This yields the following likelihood function

$$p(\mathbf{I}|\mathbf{x}) = \prod_{(x,y)} p_B(\mathbf{I}(x,y))^{1-\mathbf{T}(x,y|\mathbf{x})} p_F(\mathbf{I}(x,y)|\mathbf{x})^{\mathbf{T}(x,y|\mathbf{x})} \tag{2.63}$$

As this is rarely the case it is often more convenient to have a different foreground model for every body part, see Fig. 2.10. The foreground model is typically more difficult than the background due to non rigid deformation of the body and due to clothing. Simple foreground models are either learned offline from pixels where the model projects in an initialization pose [5, 13] or from multiple frames [65]. One major challenge with appearance models is to handle illumination changes and different lighting conditions. A major consideration with appearance based likelihood is computational cost. Note that Eq. 2.63 involves a product over all pixels in the image. Fortunately, likelihoods can be defined up to a normalizing constant. Hence, one can choose to divide the likelihood by the product of all pixels background probabilities. Using this trick many terms cancel out and yields the equivalent likelihood

$$p(\mathbf{I}|\mathbf{x}) \propto \prod_{(x,y)\ \text{s.t.}\ \mathbf{T}(x,y|\mathbf{x})=1} \frac{p_F(\mathbf{I}(x,y)|\mathbf{x})^{\mathbf{T}(x,y|\mathbf{x})}}{p_B(\mathbf{I}(x,y))^{1-\mathbf{T}(x,y|\mathbf{x})}}, \tag{2.64}$$

which is commonly referred to as likelihood ratio. This results in significant computational savings since the product in Eq. (2.64) is only over the projected model geometry. Nonetheless, evaluating appearance based likelihoods is significantly more expensive computationally than evaluating simple likelihoods based on overlap measures, points or edges. To be invariant to illumination changes scores based on HOG features could also be used to model appearance although synthesizing HOG descriptors is not straightforward. Alternatively, shading and illumination could be taken into account as an additional cue to build robust appearance models [10].

Points

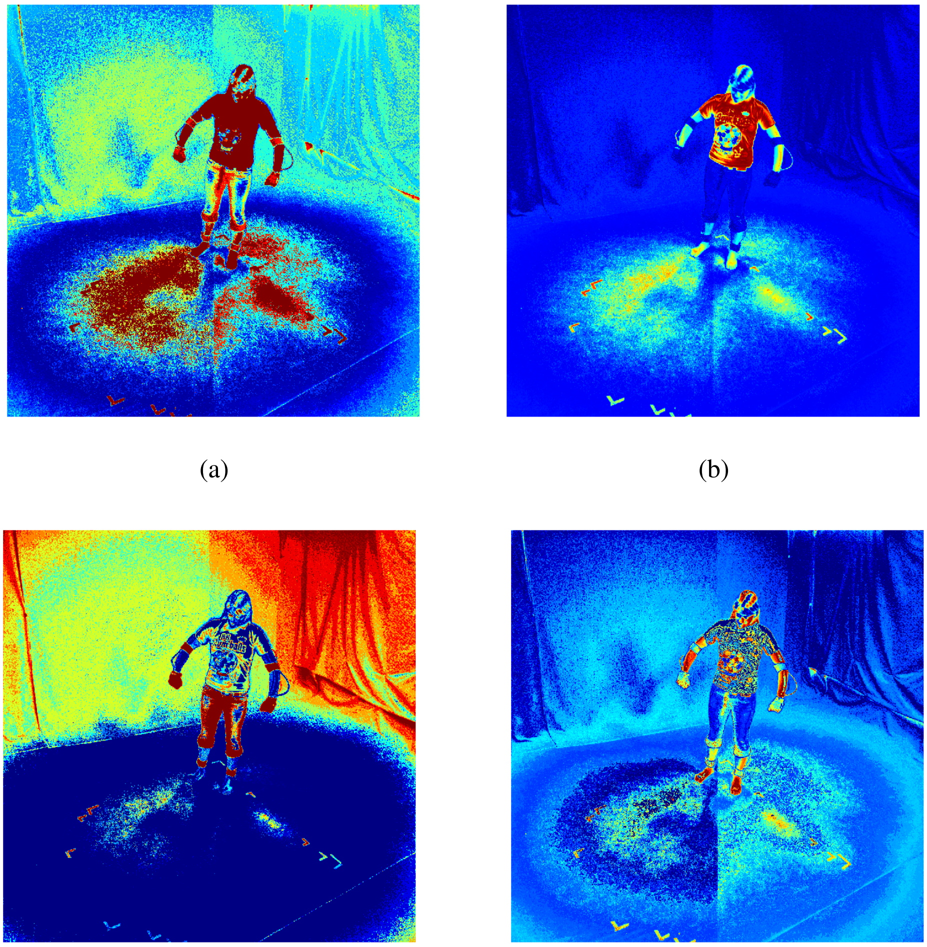
One of the simplest ways to constrain a 3D pose is with a set of image locations that correspond to known $3D$ model points. In this case, the observations are a set of $2D$ points $\mathbf{y}_{\text{points}} = \{\mathbf{r}_i\}_1^N$ that correspond to a set of model points $\mathbf{p}_s^i$. Likelihoods based on points are fairly common because likelihoods based on features, edges or silhouettes boil down to finding a set of model-image point matches. If we assume that observations are conditionally independent and are corrupted by Gaussian noise, we have the following likelihood

$$p(\mathbf{y}_{\text{points}}|\mathbf{x}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\tilde{\mathbf{r}}_i(\mathbf{x}_t) - \mathbf{r}_i\|^2}{2\sigma^2}\right) \tag{2.65}$$

where $\tilde{\mathbf{r}}_i = \mathcal{P}_c(\mathbf{p}_i)$ are the model point projections and $\mathcal{P}_c()$ is the camera projection model.

Edges

Edges together with silhouettes are perhaps the most common features used for pose estimation. They are easy to extract, are fairly invariant to different lighting conditions and clothing deformations. This models assume that a high gradient exists at the boundary of the person body. One way to integrate edges is to compute correspondences $(\tilde{\mathbf{r}}_i, \mathbf{r}_i) \in \mathcal{C}$ between the model occluding contours $\tilde{\mathbf{r}}_i \in \mathcal{O}$ and the image edges $\mathbf{r}_i \in \mathcal{E}$. The likelihood of every correspondence is modeled as a Gaussian centered at the image edge point. Assuming independence among the correspondences the likelihood can be written as a product of Gaussians as in Eq. 2.65. where the parameter σ controls how much penalty we have to pay for deviations between model points and image edges. In practice, recomputing correspondences during optimization can be expensive. One option is to pre-compute an edge. That is, a binary edge image is first obtained by thresholding and then the image is blurred by convolving it



(a) (b)

(c) (d)

Figure 2.10: Appearance likelihood. The log-likelihood ratio $\log\left(\frac{p_F}{p_B}\right)$ is shown for different appearance models of different body parts: (a) all body, (b) torso, (c) upper-legs and (d) lower arms. Notice that modeling the appearance of the foreground makes it easier to segment each body part. For example the lower legs have almost the same likelihood as the background when the foreground model is used using all the body. In contrast, they are better separated when a part specific model is built.

with a Gaussian mask. Alternatively, one can compute a distance transform on the binary edge image, see Fig. 2.11. Given a map $\mathbf{D}(x,y)$ the likelihood can be computed efficiently by simple look ups

$$p(\mathbf{D}|\mathbf{x}) = \prod_{(x,y)\in\mathcal{O}} \frac{1}{2\pi\sigma} \exp\left(-\frac{\|\mathbf{D}(x,y)\|^2}{2\sigma^2}\right). \tag{2.66}$$

Although using a distance map is computationally more efficient, this approach is problematic because it sums over model points only which can lead to degeneracy. A bidirectional likelihood that sums over both image and model points is much more robust. Unfortunately, computing a distance map for the model projection during optimization is more expensive than simply computing correspondences. Therefore, correspondences are a good trade-off between robustness and computational cost.

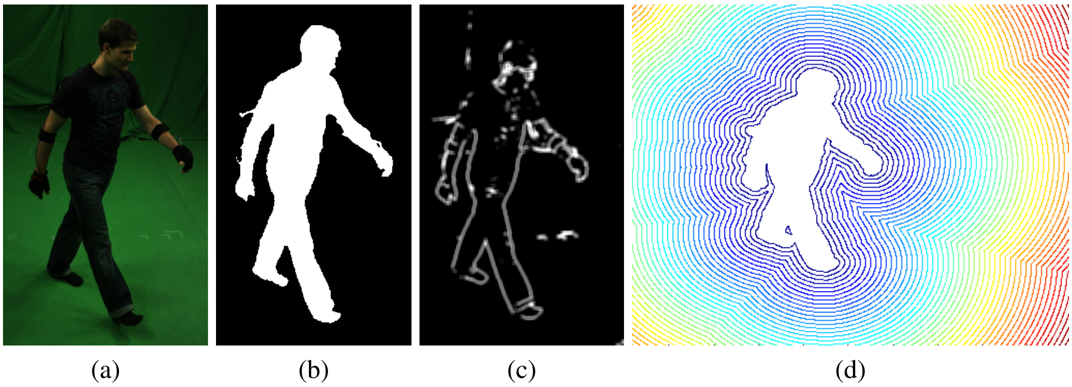

(a) (b) (c) (d)

Figure 2.11: Different features: A likelihood including multiple image features should be more robust. Here we show commonly used features such as silhouettes, edges and distance transforms due to their relative robustness to illumination changes. (a) Original image, (b) silhouette image from background substraction, (c) edge map and (d) distance transform.

Features

A problem with edges and silhouettes is that only correspondences at the boundary are obtained. This can be problematic during strong self occlusions. To overcome this limitation sparse correspondences can be obtained by computing SIFT, [91] features. Correspondences can be obtained by matching features from consecutive frames $(\mathbf{r}(t)_i),\mathbf{r}(t+1)_i) \in \mathcal{F}$. At the current frame every feature location corresponds to a model point $\mathbf{r}(t)_i$, therefore in order to maximize the likelihood the pose has to bring the model points into correspondence with the matches of the next frame $\mathbf{r}(t+1)_i$. However, this approach can lead to drift and tracking failure if errors accumulate. A better option is to collect feature descriptors from image locations where the model points project in an offline stage [92]. Then the descriptors are matched against the extracted features in the new frames.

Optic flow

One way to exploit temporal consistency is optical flow. Optical flow is the apparent motion in the image projection of 3D object in the world. The displacement $[u\ v]$ of every pixel $[x\ y]$ from one frame to the next one is computed assuming that the intensity remains constant. This is known as the *brightness constancy* assumption and it may be written as $\mathbf{I}(x,y,t-1) = \mathbf{I}(x+u,y+v,t)$. Again, we can formulate it in probabilistic terms by modeling brightness variations with a Gaussian density

$$p(\mathbf{U},\mathbf{V}|\mathbf{x}) = \prod_{(x,y)\in\mathbf{T}} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\mathbf{I}(x+u,y+v,t) - \mathbf{I}(x,y,t+1)\|^2}{2\sigma^2}\right) \tag{2.67}$$

where $\mathbf{U},\mathbf{V}$ are the horizontal and vertical flow fields respectively and $\mathbf{T}$ is the region delimited by the projected model geometry as defined earlier. The trick is to parametrize the flow field $\mathbf{U},\mathbf{V}$ using with the pose parameters $\mathbf{x}$. That is, the motion model for the optical flow is the human motion model projected to the image plane; more details are in given in the Sec. 2.6. Unfortunately, approaches relying exclusively on optical flow have two main drawbacks. First, when the motion is large, the Taylor expansion in Eq. (2.84) produces large estimate errors. Second, while image features like edges and silhouettes provide an absolute measurement, relying on optical flow causes error accumulation which results in drift and tracking failure [93]. Nevertheless, [76] was the first work in human pose estimation to use the elegant twists and exponential maps formulation from the robotics community.

2.4.2 Inertial Sensor based cues

A major limitation of image based based cues is that one third of the degrees of freedom are not observable. Particularly, axial rotations of the limbs are practically unobservable in the image. If the application at hand allows the person to be tracked to wear a small set of inertial sensors the accuracy of the pose estimation can be significantly improved as we will show in the next chapter. Inertial sensors are miniature devices that provide orientation and acceleration measurements.

Orientation cues

The orientation data from inertial sensors represent a very strong cue for pose estimation. Specifically, it is a true 3D measurement that can resolve many of the ambiguities present in 2D images. Let us denote the set of sensor orientations as $\mathcal{O}_{\text{sens}} = \{\mathbf{R}_s\}_{s=1}^{N_s}$. Assuming the orientation errors are zero mean and Gaussian, the likelihood can be written as

$$p(\mathcal{O}_{\text{sens}}|\mathbf{x}) = \prod_{s=1}^{N_s} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{d_{SO(3)}(\mathbf{R}_s,\mathbf{R}(\mathbf{x}))}{2\sigma^2}\right) \tag{2.68}$$

where $d_{SO(3)}(;,:) : SO(3) \times SO(3) \mapsto \mathbb{R}^+$ is a distance metric on $S0(3)$, see Sec. 2.1.7.

Acceleration cues

Inertial sensors also provide acceleration measurements. The acceleration data provided by the sensors is given in the coordinates of the local inertial sensor frame. Using the orientation data one can transform the acceleration data to find the coordinates in the spatial coordinate frame. However, we find that the acceleration data is too noisy and too sensitive to disturbances to be useful for pose estimation. Numerical integration to obtain either position or velocity is too unstable. Therefore, we only use it to synchronize the images with the inertial sensors as it will be explained in future sections.

2.5 Multiple cues

There is no consensus on what cues are optimal for pose estimation. If computational time is not a constraint combining cues should lead to more robustness and more accuracy. For simplicity separate measurements are usually assumed to be independent given the pose parameters $\mathbf{x}$. This allows one to factorize the likelihood and write it as a product of individual likelihoods. Given two different sets of measurements $\mathbf{y}_1$,$\mathbf{y}_1$ the likelihood can expressed as the following product

$$p(\mathbf{y}_1,\mathbf{y}_2|\mathbf{x}) = p(\mathbf{y}_1|\mathbf{x})p(\mathbf{y}_2|\mathbf{x}). \tag{2.69}$$

Log-likelihood

Given a likelihood function one seeks for the pose parameters $\mathbf{x}$ that maximize it. Let $\mathbf{y}_t = \{\mathbf{y}_t^i\}_{i=1}^N$ dennote a random vector containing multiple cues at time t (observations can be 2D point locations, lines, 3D points, feature descriptors, appearance, ...) and let $\mathbf{x}_t$ dennote the pose parameters at time t. If likelihood of the errors associated with the observations are independent and have a Gaussian distribution, the likelihood takes the form

$$p(\mathbf{y}_t|\mathbf{x}_t) = \frac{1}{C}\prod_{i=1}^{N} p(\mathbf{y}_t^i|\mathbf{x}_t) = \frac{1}{C}\exp\left(-\sum_{i=1}^{N}\frac{\|\mathbf{e}_i(\mathbf{x})\|^2}{2\sigma_i}\right) \tag{2.70}$$

where $\mathbf{e}_i(\mathbf{x}_t)$ denotes the error associated with observation $\mathbf{y}_t^i$. Taking the negative of the logarithm yields an error function which has better properties for minimization

$$e(\mathbf{x}_t) = -\log\left(p(\mathbf{y}_t|\mathbf{x}_t)\right) = -\log\left(\frac{1}{C}\right) + \sum_{i=1}^{N} \frac{\|\mathbf{e}_i(\mathbf{x}_t)\|^2}{2\sigma_i} \tag{2.71}$$

where the constant $-\log\left(\frac{1}{C}\right)$ does not depend on the pose parameters $\mathbf{x}_t$ and therefore can be disregarded. Although Bayes rule of probability is the most common model to combine measurements, other models exists such as the Dempster-Shafer theory of evidence [94, 95]. Note that the error function in Eq. (2.71) is a sum of squared errors for which efficient optimizers exist as we will see in Sec. 2.6. Ideally, one should seek the maximum a posteriori (MAP) estimate, which includes prior knowledge about the pose

$$\mathbf{x}_{t,\mathrm{MAP}} = \arg\max_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{y}_t) = \arg\max_{\mathbf{x}_t} p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t) \tag{2.72}$$

where $p(\mathbf{y}_t|\mathbf{x}_t)$ is the likelihood of the observations for a given pose $\mathbf{x}_t$ and $p(\mathbf{x}_t)$ is the prior knowledge we have about human motion which will be discussed in Sec. 2.7. Similarly, the MAP estimate can be obtained by taking the negative logarithm which yields an error term $e(\mathbf{x}_t)$ for the likelihood plus and additional prior term $e_p(\mathbf{x}_t)$

$$\mathbf{x}_{\mathrm{MAP}} = \arg\min_{\mathbf{x}_t} \; -\log\left(p(\mathbf{y}_t|\mathbf{x}_t)\right) - \log\left(p(\mathbf{x}_t)\right) = \arg\min_{\mathbf{x}_t} \; e(\mathbf{x}_t) + e_p(\mathbf{x}_t). \tag{2.73}$$

where $\mathbf{e}_i^2(\mathbf{y}_t^i|\mathbf{x}_t)$ is the individual error for a given image observation. Consequently, the error function should be designed to model the cost density associated with the observations [68]. This interpretation will be particularly useful when we see optimization methods based in stochastic search. Given a good likelihood model we need a suited optimization strategy in order to obtain a good pose estimate as we will see in Sec.2.6.

2.6 Optimization

Our aim is to recover this motion form one or multiview images. Given a model and a likelihood function one seeks for the pose parameters that maximize it. Model based algorithms are classified as generative approaches because independently of the optimization scheme used, they all model the likelihood of the observations for a given configuration of pose parameters. The pose that best explains the observations is typically found by minimizing an error function that measures how well the model fits the image data. Even using multiple cameras and relatively simple background settings this poses a hard optimization problem. Difficulties arise from model-image matching ambiguities, depth ambiguities and the high dimensionality of the state space. An additional difficulty is that the space of plausible poses only represents a small region of the full parameter space $\mathbb{R}^D$.

The key components for successful tracking, that we will describe here, are: the design of the cost or likelihood function and the optimization strategy. In this section we describe the different optimization strategies for human pose estimation and the type of error functions used.

2.6.1 Local Optimization

Given an initial estimate, local optimization methods are based on iteratively linearizing the error function to find a descent direction. Usually, these methods converge to a local optimum and consequently their performance strongly depends on the initialization. During tracking, the knowledge of the estimates in previous frames can simplify the task: in the simplest case the initial estimate is given by the pose obtained in the previous frame, or alternatively motion models can be used to make good predictions closer to the true pose. We distinguish three main families of local optimization methods for human pose estimation: methods based on *correspondences*, *optical flow* and *regions*.

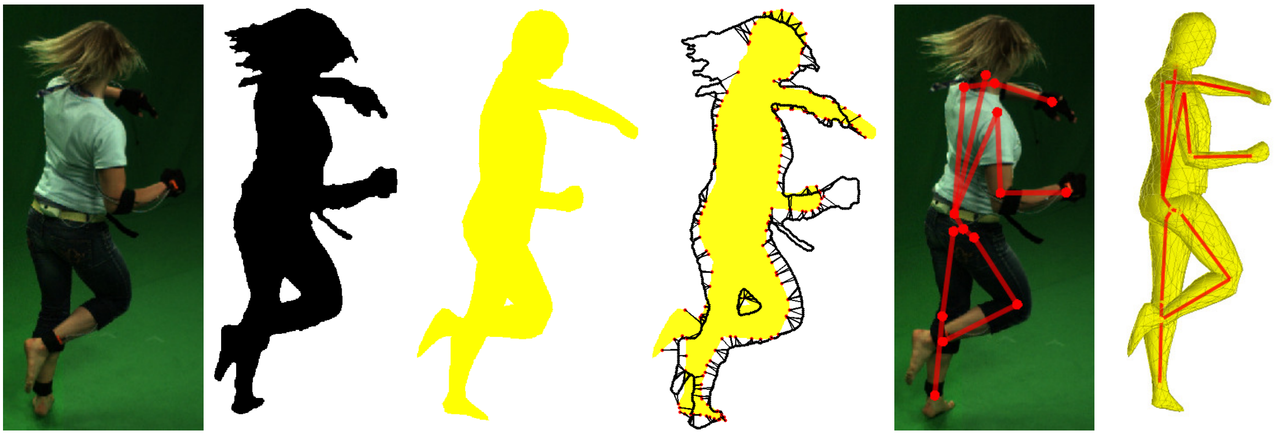


Figure 2.12: From left to right: original image, silhouette from background subtraction, rendering of the projected mesh at the current pose, correspondences by contour matching, animated model seen from a virtual view.

Correspondence based

Almost all early approaches for 3D human pose estimation were *correspondence based* and still it remain one of the most popular strategies. A reason for that is that these approaches are computationally efficient while providing very accurate results in many situations. The key idea is to collect a set of correspondences between 3D points $\mathbf{p}_i$ of the model and the *image observations* $\mathbf{r}_i$. Then, the distance between the projection of the 3D model points $\tilde{\mathbf{r}}_i$ (*predictions*) and the image observations is minimized with respect to the pose parameters $\mathbf{x}_t$. Hence, *correspondence based* algorithms consist of three main stages

1. *Feature extraction*: extract image observations (*e.g.*silhouettes, edges, SIFT features)
2. *Model image association*: match the model 3D points with the image observations and collect this correspondences
3. *Descent strategy*: find the pose parameters that bring the model points into correspondence with the image observations

Feature extraction: Different features like image silhouettes, image edges, and SIFT have been used and combined for human pose estimation. Edge and silhouettes where used in very early works and continue to be dominant for human pose estimation because they are relatively easy to extract and are stable to illumination changes. Therefore, we will explain in detail a motion capture system based on silhouettes and then the integration of additional features like SIFT will become obvious. In the context of human tracking a silhouette is a binary image with white pixels indicating the foreground *i.e.*, the region of the subject we want to track. In indoor environments, silhouettes can be obtained with great accuracy via background subtraction techniques [90]. In outdoor environments, it is considerably more challenging but also possible if background images are available. Once the silhouettes are obtained, the image contour is obtained with an edge detector.

Model image association: For the correspondences, since we want to predict the image contour, only points belonging to the *occluding contour* are considered. A point belongs to the occluding contour $\mathbf{p}_i \in \mathcal{O}$ if its surface normal $\hat{\mathbf{n}}$ is perpendicular to the line $\mathbf{L}$ connecting the camera center $\mathbf{O}$ and the point. In other words, the occluding contour is the set of points in the mesh that project to the silhouette contour of the projected mesh.

To find the points of the occluding contour there are two main strategies, the first and the simplest one is to test for every point in the mesh if the surface normal is perpendicular to the projection ray:

$$\mathbf{p} \in \mathcal{O} \quad if \quad \hat{\mathbf{n}} \cdot (\mathbf{p} - \mathbf{C}) < \epsilon \tag{2.74}$$

where ϵ is a small positive threshold. In practice, however, this approach is problematic since the accuracy of $\hat{\mathbf{n}}$ strongly depends on the mesh resolution (number of vertices of the mesh). One solution is to look for sign changes of the angle between the triangle normal and the projection ray. The second strategy is to first render a binary silhouette projection on the image, project all the vertices of the mesh and retain only those on the silhouette boundary. To render the silhouette image, all the visible surface triangles are projected to the image and filled using typical graphics raster-scan algorithms. Alternatively, the rendering can be very efficiently performed on the GPU using OpenGL[3]. At this point we have two sets of points, the 3D points in the occluding contour $\mathbf{p}_i \in \mathcal{O}$ and the 2D points from the image contour $\mathbf{r}_i \in \mathcal{I}$, The correspondences can be found by finding for every point projection $\tilde{\mathbf{r}}_i = \mathcal{P}_c(\mathbf{p}_i)$ the k-nearest neighbors in the image contour [68, 96, 77, 52, 8, 49]. This will result in a set of 3D-2D correspondences $(\mathbf{p}_i,\mathbf{r}_i)$ or 2D-2D correspondences $(\tilde{\mathbf{r}}_i,\mathbf{r}_i)$. We note that finding correct correspondences is a hard problem with probably many local minima. To leverage this, additional terms based on overlap between the model and image silhouette can be included into the matching cost [7].

Descent strategies: Collecting many of these correspondences $(\tilde{\mathbf{r}}_i,\mathbf{r}_i)$ one may define a likelihood function based on points as explained in Sec. 2.4.1. Maximizing of the likelihood in Sec. 2.4.1 is equivalent to minimizing the error function $e : \mathbb{R}^D \mapsto \mathbb{R}$ consisting of a sum of squared re-projection errors in the image

$$e(\mathbf{x}_t) = \sum_i^N \mathbf{e}_i^2(\mathbf{x}_t) = \sum_i^N \|\tilde{\mathbf{r}}_i(\mathbf{x}_t) - \mathbf{r}_i\|^2 \tag{2.75}$$

which we want to minimize with respect to the pose parameters $\mathbf{x}_t$. Note that in the case of 2D-2D correspondences the individual residual errors $\mathbf{e}_i \in \mathbb{R}^2$ are 2D error vectors $\mathbf{e}_i = (\Delta r_{i,x},\Delta r_{i,y})$. Eq. (2.75) is a classical non-linear least squares that can be re-written in vector form as

$$e(\mathbf{x}_t) = \mathbf{e}^T\mathbf{e} \tag{2.76}$$

where $\mathbf{e} \in \mathbb{R}^{2N}$ is the total residual error $\mathbf{e} = (\mathbf{e}_1^T,\mathbf{e}_2^T,\ldots,\mathbf{e}_N^T)$. Eq. (2.76) can be efficiently minimized using a *Gauss-Newton* style minimization. The trick is to iteratively linearize the vector function $\mathbf{e} \in \mathbb{R}^{2N}$ around the solution with the Jacobian matrix $\mathbf{J}_t$ to find a descent step. In the literature, the expression for the Jacobian matrix is often omitted due to space limitations. Therefore, we reproduce here how to derive the analytical expression of the Jacobian matrix $\mathbf{J}_t \in \mathbb{R}^{2N \times D}$ of the residual error $\mathbf{e}$. We start by deriving the expression for the Jacobian of the error of a single correspondence $\mathbf{J}_{t,i} = \frac{\mathbf{e}_i}{\Delta\mathbf{x}}$. It is straightforward to see that individual error Jacobian equals the prediction Jacobian $\mathbf{J}_{t,i} = \frac{\mathbf{e}_i}{\Delta\mathbf{x}} = \frac{\tilde{\mathbf{r}}_i}{\Delta\mathbf{x}}$ because only the prediction $\tilde{\mathbf{r}}_i$ depends on the pose parameters. Recall that the matrix $\mathbf{J}_{t,i} \in \mathbb{R}^{2 \times D}$ maps increments in the pose parameters $\Delta\mathbf{x}$ to increments in the predictions $\Delta\tilde{\mathbf{r}}_i$. To compute the Jacobian it is useful to identify the set of transformations applied to a point $\mathbf{p}_s(0)$ in the reference pose to the final projection in the image $\tilde{\mathbf{r}}$, see Sec. 3.1. We can visualize this with the following diagram

$$\begin{tikzcd}
\mathbf{p}_s(0) \arrow[r, "\mathbf{G}_{sb}(\mathbf{x}_t)"] & \mathbf{p}_s \arrow[r, "g_c := \mathbf{M}_{ext}"] & \mathbf{p}_c \arrow[r, "g_p := f(\frac{X}{Z}+o_x,\frac{X}{Z}+o_y)"] & \tilde{\mathbf{r}}
\end{tikzcd}$$

where $\mathbf{G}_{sb}(\mathbf{x}_t)$ is the concatenation of rigid motions in the kinematic chain given by the pose parameters $\mathbf{x}_t$, $g_c(\mathbf{p}) \mapsto \mathbf{M}_{ext}\,\mathbf{p}$ is the extrinsic camera matrix that transforms a point from spatial coordinates

---

[3] www.opengl.org

$\mathbf{p}_s$ to camera coordinates $\mathbf{p}_c$ and $g_p(X,Y,Z) \mapsto (f\,\frac{X}{Z} + o_x, f\,\frac{X}{Z} + o_y)$ is the perspective projection of 3D point in camera coordinates onto the image plane (with f denoting the focal length, (o_x,o_y) the principal point and we assume squared pixels with no skew). Now we can compute the Jacobians $\mathbf{J}_c \in \mathbb{R}^{3\times3}$,$\mathbf{J}_{g_p} \in \mathbb{R}^{2\times3}$ of the functions g_c,g_p separately as

$$g_c : \mathbb{R}^3 \to \mathbb{R}^3 \qquad \mathbf{p}_c^i = \mathbf{M}_{ext}\,\bar{\mathbf{p}}_s^i = \mathbf{R}_{cs}\,\mathbf{p}_s^i + \mathbf{t}_{cs} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad \mathbf{J}_c = \mathbf{R}_{cs}$$

$$g_p : \mathbb{R}^3 \to \mathbb{R}^2 \qquad \tilde{\mathbf{r}}_i = g_p(\mathbf{p}_c^i) = (f\,\frac{X}{Z} + o_x, f\,\frac{Y}{Z} + o_y) \qquad \mathbf{J}_{g_p} = f \begin{bmatrix} \frac{1}{Z} & 0 & -\frac{X}{Z^2} \\ 0 & \frac{1}{Z} & -\frac{Y}{Z^2} \end{bmatrix}$$

where the Jacobian of g_c is directly the rotational component of the extrinsics, $\mathbf{R}_{cs}$ because it is a linear map and the Jacobian of g_p is computed by direct application of the definition of the Jacobian matrix [97]. By applying the chain rule, the Jacobian of the composed mapping $\mathbf{J}_{t,i}$ might be written as

$$\mathbf{J}_{t,i} = \frac{\Delta\tilde{\mathbf{r}}_i}{\Delta\mathbf{x}_t} = \frac{\Delta\tilde{\mathbf{r}}_i}{\Delta\mathbf{p}_c} \cdot \frac{\Delta\mathbf{p}_c}{\Delta\mathbf{p}_s} \cdot \frac{\Delta\mathbf{p}_s}{\Delta\mathbf{x}_t} = \mathbf{J}_{g_p}\,\mathbf{R}_{cs}\,\mathbf{J}_p(\mathbf{x};\mathbf{p}_s^i), \tag{2.77}$$

where $\mathbf{J}_p(\mathbf{x};\mathbf{p}_s^i)$ is the pose Jacobian derived earlier in equation Eq. (2.58). It is straightforward to see that the Jacobian of total residual error $\mathbf{J}_t \in \mathbb{R}^{2N\times D}$ may be written by stacking the individual point Jacobians $\mathbf{J}_{t,i} \in \mathbb{R}^{2\times D}$

$$\mathbf{J}_t = \frac{\Delta\mathbf{e}}{\Delta\mathbf{x}} = \begin{bmatrix} \mathbf{J}_{t,1} \\ \mathbf{J}_{t,2} \\ \vdots \\ \mathbf{J}_{t,N} \end{bmatrix}. \tag{2.78}$$

With the analytical expression of the residual Jacobian the Gauss-Newton method calculates the descent step as follows

$$\begin{aligned} \Delta\mathbf{x} &= \arg\min_{\Delta\mathbf{x}} \frac{1}{2}\,\mathbf{e}^T(\mathbf{x}_t + \Delta\mathbf{x})\;\mathbf{e}(\mathbf{x}_t + \Delta\mathbf{x}) \\ &= \arg\min_{\Delta\mathbf{x}} \frac{1}{2}\,(\mathbf{e} + \mathbf{J}_t\Delta\mathbf{x})^T\,(\mathbf{e} + \mathbf{J}_t\Delta\mathbf{x}) \\ &= \arg\min_{\Delta\mathbf{x}} \frac{1}{2}\mathbf{e}^T\mathbf{e} + \Delta\mathbf{x}^T\mathbf{J}_t^T\mathbf{e} + \frac{1}{2}\Delta\mathbf{x}^T\,\mathbf{J}_t^T\,\mathbf{J}_t\,\Delta\mathbf{x} \end{aligned} \tag{2.79}$$

where $\mathbf{e} = \mathbf{e}(\mathbf{x}_t)^T$ and $\mathbf{J}_t = \mathbf{J}_t(\mathbf{x}_t)$ are evaluated at the current estimation $\mathbf{x}_t$. Finally, derivating with respect to $\Delta\mathbf{x}$ and equating to zero we find that the descent step is:

$$\Delta\mathbf{x} = -(\mathbf{J}_t^T\,\mathbf{J}_t)^{-1}\,\mathbf{J}_t^T\,\mathbf{e} \tag{2.80}$$

At every iteration of the Gauss-Newton algorithm the step is computed using equation Eq. (2.80) and the pose parameters are updated $\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta\mathbf{x}$. This procedure is repeated until the algorithm converges. The step $\Delta\mathbf{x}$ always decreases the error function $e(\mathbf{x}_t)$ as long as the Jacobian matrix $\mathbf{J}_t$ has full rank. In the *Levenberg-Marquadt* algorithm, the Gauss-Newton step is modified

$$\Delta\mathbf{x} = -(\mathbf{J}_t^T\,\mathbf{J}_t + \mu\mathbf{I})^{-1}\,\mathbf{J}_t^T\,\mathbf{e} \tag{2.81}$$

by introducing an additional dynamically chosen parameter $\mu\mathbf{I}$ that improves the performance. If the step decreases the error, the step is accepted and the value of μ is reduced. If the step increases the error, μ is increased and a new step is computed. When μ is large the method performs like

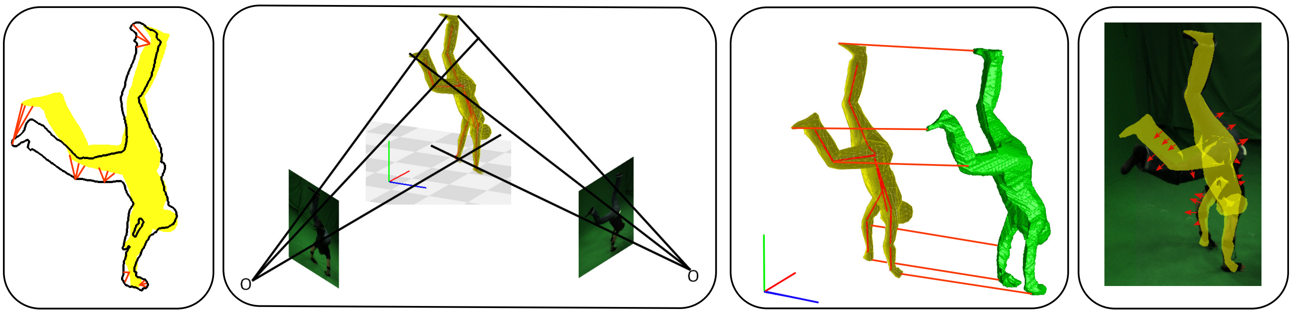


Figure 2.13: Different error functions, from left to right: minimization of reprojection error (2D-2D correspondences), point to line distance error minimization (2D-3D correspondences), point to point distance error minimization (3D-3D correspondences) and region based tracker

standard gradient descent, slow but guaranteed to converge. When μ is small it performs like Gauss-Newton. Once the algorithm has converged the obtained pose estimate is used as initialization for the next frame, new correspondences are found in the new image and the process is repeated. For large motions it is often needed to reproject the model to the image several times to obtain refined correspondences, similar to the standard ICP (*iterative closest point*) registration method [98, 99].

Different error functions: Different error functions have been proposed in the literature. For example, in [77] they directly minimize the sum of squared distances between 3D model points $\mathbf{p}_s^i$ and the projection rays L_i casted by the corresponding 2D image observations $\mathbf{r}_i$. Writing the projection line in Plücker coordinates $L_i = (\mathbf{m}_i,\mathbf{n}_i)$ the error may be written as

$$e(\mathbf{x}_t) = \sum_i^N \mathbf{e}_i^2(\mathbf{x}_t) = \sum_i^N \|\mathbf{p}_s^i(\mathbf{x}_t) \times \mathbf{n}_i - \mathbf{m}_i\|^2 \tag{2.82}$$

where the residuals are 3D distance error vectors $\mathbf{e}_i \in \mathbb{R}^3$. In this case it is straightforward to show that the Jacobian of the error of one correspondence pair is given by $\mathbf{J}_{t,i} = \mathbf{n}_i^{\wedge}\mathbf{J}_{\mathbf{x}}(\mathbf{p}_s^i) \in \mathbb{R}^{3\times D}$.

Another alternative used by several authors [100, 88] is to first reconstruct the visual hull [101] from the multiview images obtaining a rough volume of the human shape. Then the matching is done between the model points and the points from the visual hull resulting in a set of 3D-3D correspondences $(\mathbf{p}_s^i,\mathbf{q}_s^i)$. The error function is then simply the distance between 3D points

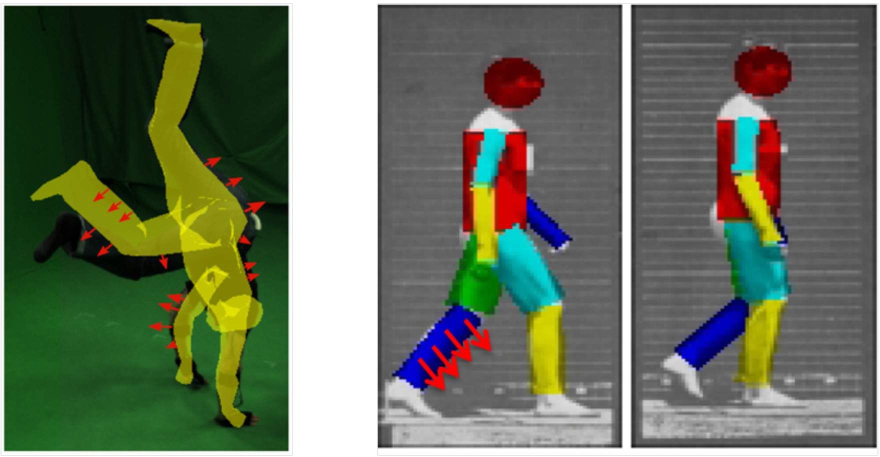
$$e(\mathbf{x}_t) = \sum_i^N \mathbf{e}_i^2(\mathbf{x}_t) = \sum_i^N \|\mathbf{p}_s^i(\mathbf{x}_t) - \mathbf{q}_s^i\|^2 \tag{2.83}$$

with $\mathbf{e}_i \in \mathbb{R}^3$ where in this case the Jacobian is directly $\mathbf{J}_{t,i} = \mathbf{J}_{\mathbf{x}}(\mathbf{p}_s^i)$.
In Figure 2.13 an illustration of common error functions is shown.

Combining features: Other kind of features like SIFT or optical flow can be integrated as additional correspondences into this framework as long as they can be predicted given a pose estimate. The combination of features should robustify the tracker [102].

Optical flow based

Optical flow is the apparent motion in the image projection of 3D object in the world. The displacement $[u\ v]$ of every pixel $[x\ y]$ from one frame to the next one is computed assuming that the intensity



(a) (b)

Figure 2.14: Error functions: (a) Region based error function and (b) optical flow based error function. Region based likelihoods can be optimized locally by taking the gradient direction in pose space that best separates the foreground from background. Similarly, optical flow can be integrated in local methods by parameterizing the flow using the the human motion model.

remains constant. This is known as the *brightness constancy* assumption and it may be written as $I(x,y,t-1) = I(x+u,y+v,t)$. The first order Taylor expansion of the right hand side of the equation leads to the remarkable normal optical flow constraint equation [103],

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} - I_t = 0 \tag{2.84}$$

where I_x,I_y and I_t are the spatial and temporal derivatives of the image. Generally, neighboring pixels are assumed to move together according to a given motion model. Bregler *et al.* [104, 76] used a human motion model to parameterize the motion flow $[u\ v]$. Specifically, he finds for every pixel in the image the corresponding 3D point in the human model $\mathbf{p}_s$. Then the motion $[u\ v]$ is simply given by the projection of the 3D point displacement $\Delta\mathbf{p}_s$ onto the image plane. This can be written as

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \cdot \mathcal{P}_c(\Delta\mathbf{p}_s) - I_t = 0 \tag{2.85}$$

where recall that $\mathcal{P}_c()$ dennotes projection on the image plane (in [104, 76] an orthographic camera projection is assumed), and the point displacement is given by $\Delta\mathbf{p}_s = \mathbf{J}_\mathbf{x}\Delta\mathbf{x}_t$ Eq. (2.57). Note that the error function in Eq. 2.85 derives from the likelihood explained in Sec. 2.4.1. Eq. (2.85) can alternatively be interpreted as a dense correspondence error function where the following error $\mathbf{e}_i(\mathbf{x}_t) = I(x+u,y+v,t) - I(x,y)$ for one correspondence is linearized. Here only $[u\ v]$ depend on the pose parameters. The total error function $e(\mathbf{x}_t)$ can be interpreted then as the sum of squared pixel intensity differences. Unfortunately, approaches relying exclusively on optical flow have two main drawbacks. First, when the motion is large, the Taylor expansion in Eq. (2.84) produces large estimate errors. Second, while image features like edges and silhouettes provide an absolute measurement, relying on optical flow causes error accumulation which results in drift and tracking failure [93]. Nevertheless, [76] was the first work in human pose estimation to use the elegant twists and exponential maps formulation from the robotics community.

Region based

Region based methods are based on separating the foreground figure (the human silhouette) from the background by maximizing the dissimilarity between region density functions (usually the density functions are approximated by simple histograms of pixel intensities and colors). Popular approaches to achieve this are level sets or graph-cuts [105, 106]. This process can be coupled with human pose estimation in an Expectation-Maximization scheme. An initial human pose estimate defines two regions in the image, namely the interior of the projected silhouette and the exterior. This initial boundary is then used as a shape prior for a levelset segmentation. Thereafter, correspondences between the segmentation and the projected silhouette are obtained and the pose is estimated using a correspondence-based method. This process of pose estimation and segmentation is iterated until convergence. Some works have coupled the feature extraction and the descent strategy. The work by [87, 107] skips the segmentation step and directly shifts the points belonging to the occluding contour $\tilde{\mathbf{r}} \in \mathcal{O}$ inwards or outwards (orthogonal to the contour line) according to the posterior probability densities. If the foreground posterior is bigger than the background posterior the point is shifted outwards, otherwise the point is shifted inwards, see Figure 2.13. This implicitly generates correspondence pairs between points and shifted points which feed a correspondence based tracker.

2.6.2 Particle-based Optimization and Filtering

A well known problem of local optimization methods is that since they are based on propagating a single pose hypothesis, when there is a tracking error the system can in general not recover from it. To overcome this limitation, stochastic search techniques have been introduced for human pose estimation. This group of methods are based on approximating the likelihood of the image given the pose parameters by propagating a set of particles from one time step to the next one.

Particle Filter

Problems in human pose estimation arise from kinematic singularities, depth and orientation ambiguities and occlusions. For all this reasons the posterior density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ and the observation process $p(\mathbf{y}_t|\mathbf{x}_t)$ are highly peaked and highly multimodal. The image likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ is the probability of observing certain image features given a pose $\mathbf{x}_t$, and $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is the probability of the pose parameters considering the history of all observations from previous images $1 : t$. To see this multimodality, note that many configurations in pose parameter space $\mathbf{x}$ explain well the observations $\mathbf{y}_{1:t}$ (for example any rotation by an angle α about the axis of a limb will project to almost the same image location). It is well known that in this case a Kalman filter will fail. In these cases the posterior can be approximated by a particle filter (PF). Samples are generated using Markov Chain Monte Carlo. A particle filter approximates the posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ by a set of particles $\{\pi_t^{(i)}, \mathbf{x}_t^{(i)}\}_{i=1}^N$ where the weights are normalized so that $\sum_N \pi_t^{(i)} = 1$. Each particle $\mathbf{x}_t^{(i)}$ corresponds to one configuration of pose parameters Eq. (2.54), and the weights are chosen to be proportional to the likelihood $\pi^{(i)} \propto p(\mathbf{y}_t|\mathbf{x}_t = \mathbf{x}_t^{(i)})$. At each time step the pose parameters can be estimated by the mean of the weighted particles,

$$\mathbf{x}_t^* = \mathbb{E}_{\mathbf{x}}[\mathbf{x}_t] \simeq \sum_N \pi_t^{(i)} \, \mathbf{x}_t^{(i)}. \tag{2.86}$$

or by the mode of the particle set $\mathbf{x}_t^* = \mathbb{M}_{\mathbf{x}}[\mathbf{x}_t] = \mathbf{x}_t^{(i)}$ with $\pi_t^{(i)} = \max\left(\pi_t^{(n)}\right)$.

Assuming a first order Markov process $(p(\mathbf{x}_t|\mathbf{x}_{1:t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}))$ the posterior distribution can be updated recursively

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \tag{2.87}$$

where the integral computes the pose prediction from the previous time step posterior $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ propagated with the dynamical model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The prediction is then weighted by the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$ times the prior $p(\mathbf{x}_t)$ if available. In a particle filter setting, Eq. (2.87) is approximated by *importance sampling*, see for example [108].

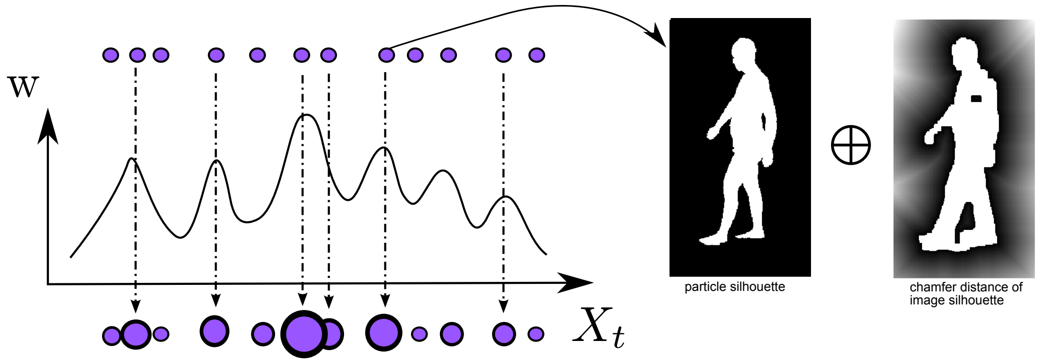


Figure 2.15: *Particle Filter*: on the left the weighting function is shown as a function of the pose parameters. The function is multimodal and it is difficult to tell where the maximum is from the particle distribution. On the right: the weighting function is evaluated for every particle $\mathbf{x}_t^{(i)}$. The weighting function should be fast to evaluate, in this example, it consists of a simple overlap measure between the particle silhouette and the chamfer distance transformed image silhouette.

Given a set of weighted particles approximating the posterior in the previous frame $\mathcal{P}_{t-1}^{+} := \{\pi_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}\}_{i=1}^{N}$, N particles are drawn from $\mathcal{P}_{t-1}$ with replacement and probability proportional to their weights obtaining a new set of unweighted particles $\{\tilde{\mathbf{x}}_{t-1}^{(i)}\}_{i=1}^{N}$. Thereafter, the particles are propagated to the next frame by sampling from the dynamical model $p(\mathbf{x}_t|\tilde{\mathbf{x}}_{t-1}^{(i)})$ producing a new unweighted set of predictions $\mathcal{P}_t^{-} := \{\mathbf{x}_t^{(i)}\}_{i=1}^{N}$. Finally, every particle in $\mathcal{P}_t^{-}$ is weighted according to the likelihood in the new frame $\pi_t^{(i)} = p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$ obtaining the final weighted set $\mathcal{P}_t^{+} := \{\pi_t^{(i)}, \mathbf{x}_t^{(i)}\}_{i=1}^{N}$ that approximates the updated posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

Likelihood functions: Ideally, to model the observation process $p(\mathbf{y}_t|\mathbf{x}_t)$ the complicated image formation process has to be synthesized, *i.e.*, illumination, human appearance rendering on the image, clothing, occlusions, shadows, etc. Since $p(\mathbf{y}_t|\mathbf{x}_t)$ has to be evaluated for every particle in the set $p(\mathbf{y}_t|\mathbf{x}_t = \mathbf{x}_t^{(i)})$, this is obviously computationally infeasible. In practice, to make the problem tractable an intuitive function $w(\mathbf{y}_t, \mathbf{x}_t = \mathbf{x}_t^{(i)})$ is constructed that approximates the probability $p(\mathbf{y}_t|\mathbf{x} = \mathbf{x}_k^{(i)})$. This function takes into account only image observations $\mathbf{y}_t$ that can be modeled easy and efficiently (*e.g.*, edges, coarse foreground appearance or silhouettes). Actually, the error functions $e(\mathbf{x}_t)$ described for local optimization might be used to set the weights w as

$$w(\mathbf{y}_t, \mathbf{x}_t = \mathbf{x}_t^{(i)}) = \exp\left(-e\left(\mathbf{x}_t^{(i)}\right)\right) \tag{2.88}$$

where, as explained before in Section 2.5, we interpret the error as the cost density associated with the observations. To gain in efficiency, a chamfer distance can be pre-computed in the original image silhouette or edge map, see Figure 2.15. Then, simple overlap measures between the synthesized particle silhouette and the chamfer distance image are computed [4, 5, 6]. Another commonly used feature in the weighting function is appearance, whose associated cost can be evaluated with histogram comparison [5, 12]. For a comparative study of the influence of different likelihood/weighting functions we refer the reader to [109]. Nonetheless, the computation of $w(\mathbf{y}_t, \mathbf{x}_t = \mathbf{x}_t^{(i)})$ is a very expensive operation if it has to be evaluated for many particles. In addition, the number of particles required to approximate $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ grows exponentially with the dimensionality of $\mathbf{x}$ which makes the particle filter intractable for realistic human models with more than 30 DoF. Furthermore, even using a large number of particles the search can be misdirected if many local modes are present in $w(\mathbf{y}_t, \mathbf{x}_t = \mathbf{x}_t^{(i)})$, see Figure 2.15.

Annealed Particle Filter

The annealed particle filter (APF)is a modification of the particle filter and it was introduced for human pose estimation by Deutscher *et al.*[4]. The goal here is to modify the particle filter such that the number of needed particles is drastically reduced and the particles do not congregate around local maxima. The APF is motivated from simulated annealing methods designed for global optimization of multimodal functions [110]. The key idea is to gradually introduce the influence of narrow peaks in the weighting function $w(\mathbf{y}_t, \mathbf{x}_t)$. This is achieved by starting a search run in successive layers gradually changing the weighting function as for $\beta_0 > \beta_1 > \ldots > \beta_M$. The run is started at layer M, where w_M is broad and reflects the overall trend of w without the influence of so many local maxima. As we move to the next layer, β increases and therefore the local peaks become more accentuated. For initialization, an initial set of samples is drawn from a proposal distribution q_{M+1}. During tracking we might choose this distribution to be the set of particles from the previous frame or the propagated particles if we use a motion model. For initialization in the first frame the distribution should be spread with a high variance, see for example [111]. Once the algorithm is initialized the optimization consists of the following steps executed at every layer: *weighting*, *resampling* and *diffusion*, see Figure 2.16 and the pseudo code in Algorithm 1.

Algorithm 1 Annealed particle filter

```
Require: number of layers M, number of samples $N$, initial distribution $q_{M+1}$,
  Initialize: Draw $N$ initial samples from $q \to \mathbf{x}_{t,m}^{(i)}$
  for layer $m = M$ to 0 do
    1. WEIGHTING
    start from the set of un-weighted particles of the previous layer
    for $i = 1$ to $N$ do
      compute $w_{t,m}(\mathbf{y}_t,\mathbf{x}_{t,m}^{(i)})^{\beta m}$
      set $\pi_{t,m}^{(i)} \propto w_{t,m}(\mathbf{y}_t,\mathbf{x}_{t,m}^{(i)})^{\beta m}$
    end for
    set $q_m = \{\pi_{t,m}^{(i)}, \mathbf{x}_{t,m}^{(i)}\}_{i=1}^{N}$
    2. RESAMPLING
    draw $N$ samples from $q_{t,m} \to \mathbf{x}_{t,m}^{(i)}$        {Can be done with multinomial sampling}
    3. DIFFUSION
    $\mathbf{x}_{t,m-1}^{(i)} = \mathbf{x}_{t,m}^{(i)} + \mathbf{B}_{t,m}$        {$\mathbf{B}_{t,m}$ is a sample from a multivariate Gaussian with $\mathcal{N}(0,\Sigma)$}
  end for
```

- *Weighting*: The surviving particles of the previous layer are assigned new weights $w_{t,m}(\mathbf{y}_t,\mathbf{x}_{t,m}^{(i)})^{\beta m}$ with the new annealing scheme βm. At this point a new proposal distribution has been formed $q_m = \{\pi_{t,m}^{(i)}, \mathbf{x}_{t,m}^{(i)}\}_{i=1}^{N}$.

- *Resampling*: N new samples are drawn from the distribution q_m with probability equal to the weights, this can be efficiently done with multinomial sampling. Note that particles with high weight (low error) will be selected with higher probability and therefore a higher number of particles concentrate around the maximum of the weighting function. Gall *et al.*[112, 5] proposes a generalization of the resampling strategy that improves the performance of the APF. In this modification, particles with high weight are with high probability retained in the set without replacement.

- *Diffusion*: In order to explore the search space the particles are diffused by some function. A common choice is to shift every particle by adding a sample from a multivariate Gaussian with covariance Σ_m. The covariance Σ_m can be chosen to be proportional to the particle set variance since this provides a measure of uncertainty (The more uncertainty, the farther away we have to explore the search space). The run in the layer terminates with the diffusion step and the particles are used to initialize the weighting step of the next layer.

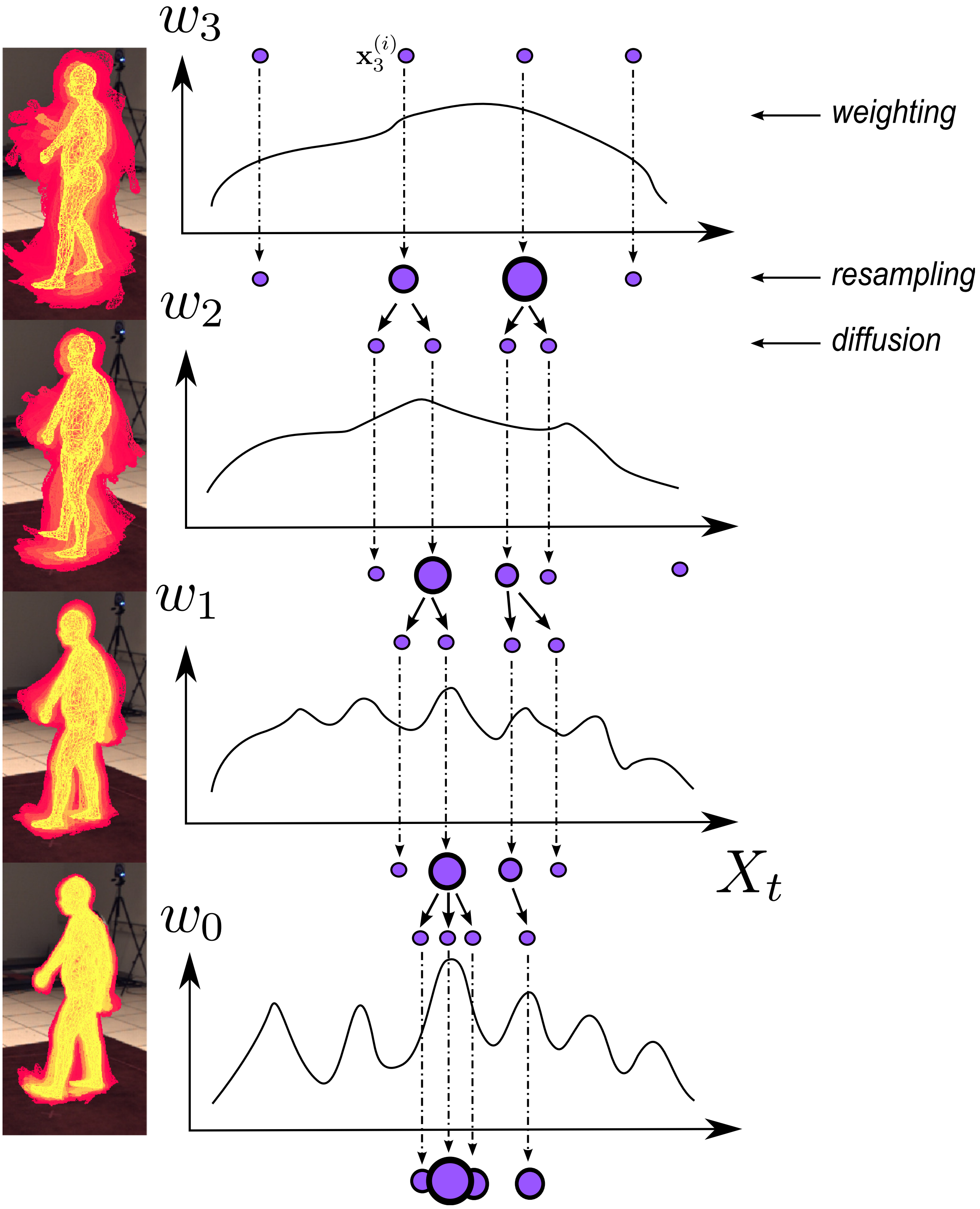


Figure 2.16: *Annealed particle filter*: At each layer the weighting, resampling and diffusion steps are performed. The influence of the peaks is gradually introduced into the weighting function w_m. Consequently, the particles converge at the global maximum in the last layer w_0 and do not get trapped in local maxima peaks as opposed to the standard particle filter. Additionally, on the left column, the distribution of the particles projected to the image is shown, where particles with higher weights are brighter (left column images are courtesy of Gall *et al.*[5]).

$$w_m(\mathbf{y}_t,\mathbf{x}_t) = w(\mathbf{y}_t,\mathbf{x}_t)^{\beta m}, \tag{2.89}$$

Once the algorithm has finished the last annealing run, the pose estimate is obtained by the mean of the final set of particles of the last layer, $\mathbf{x}_t^* = \mathbb{E}_{\mathbf{x}_t}[q_0] = \sum_i \pi_{t,0}^{(i)} \, \mathbf{x}_{t,0}^{(i)}$. Although the APF can be used recursively as the standard PF in Eq. (2.87) using some heuristics, it can be considered a single frame global optimization algorithm. While the APF is computationally more efficient in locating global minima in the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$ than the particle filter, the main disadvantage is not being able to work within a robust Bayesian framework [4]. The reason for that is that the set of particles of q_0 congregate *only* around one maximum of the observation process $p(\mathbf{y}_t|\mathbf{x}_t)$ at the current frame t. Therefore, the particles are not well distributed for the next frame and usually a heuristic has to be used to spread them to search in the next frame. By contrast, in the PF (which needs much more samples) the final particle set represents the total posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ which might contain multiple maxima. Therefore the PF can represent inherent ambiguities by maintaining multiple modes at the expense of a significant loss in accuracy.

Tailored Optimization Methods

Although an exhaustive survey of the proposed sampling methods and likelihood functions used in the literature for human tracking is out of the scope of this chapter, it is worth to highlight some optimization procedures tailored for the problem of human pose estimation. Choo and Fleet [113] use a Markov Chain similar to Eq. (2.87) but use every particle as initialization for local optimization on the likelihood function. Similarly, Sminchisescu and Triggs [68] also combine particle based sampling with local optimization, but additionally sample along the most uncertain directions calculated from the local likelihood Hessian matrix at the fitted optima. Along this lines, importance samplers have been proposed that focus samples on regions likely to contain transitions states leading to nearby minima [7]. Another way to locate multiple optima in monocular images is to exploit the geometry of the problem to deterministically enumerate the set of poses that result in the same projection [26, 23]. To make the sampling tractable in the high dimensional space, state partitions have also been used [114, 115]. These partitions are either specified manually by sequentially tracking different kinematic branches (*e.g.* torso, limbs and head) [114] or selected automatically from the parameter variances in the particle set [115]. Every optimization scheme has its own advantages/disadvantages but a common key component in all of them for successful tracking is a good background subtraction.

2.7 Prior Knowledge

2.7.1 Joint Limits

The kinematic structure of the human body permits a limited range of motion in each joint. Knees and elbows for example can not be hyper-extended. A simple way to integrate such constraints is to include joint limits. This is enforced including a set of linear inequalities for every joint angle $lb < \theta_i < ub$ for $i = 1, \dots, n$, where lb and ub are the lower and upper bound joint limits. In a particle based optimization scheme, satisfying joint limits can be easily achieved by sampling on the valid interval only. In a local optimization scheme, after linearization of the objective function the step $\Delta\mathbf{x}$ is found by solving quadratic programing (QP) problem

$$\begin{array}{lll} \min & \|\mathbf{J}\,\Delta\mathbf{x} + \mathbf{e}\|^2 & \\ \text{subject to} & \mathbf{C}\Delta\mathbf{x} \geqq d_j & \text{for } j = 1, \dots, m \quad \text{Inequality constraints} \end{array}$$

where $\mathbf{J}$ is the Jacobian of the residuals vector $\mathbf{e}$ and $\mathbf{C}$ is a matrix of constraints.

2.7.2 Collision Avoidance

Another desirable restriction is to avoid self-penetration in the human model. Testing for self-penetration during optimization can be too expensive if a fully triangulated mesh is used. A simple but very effective approach is to approximate each body part by a geometric primitive. In that case, intersection can be tested very fast. Typically, when an intersection is detected one includes a penalty in the objective function as in [68, 116, 51].

2.7.3 Motion Models

Another way to constraint the human pose is with motion models. The reasoning is that the poses should vary smoothly over time. Perhaps the most simple prior model of human motion is a smooth, low-order Markov process. Typically, a first order model assumes that the pose at $\mathbf{x}_{t+1}$ equals the pose at the previous frame up to additive Gaussian noise:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{w} \tag{2.90}$$

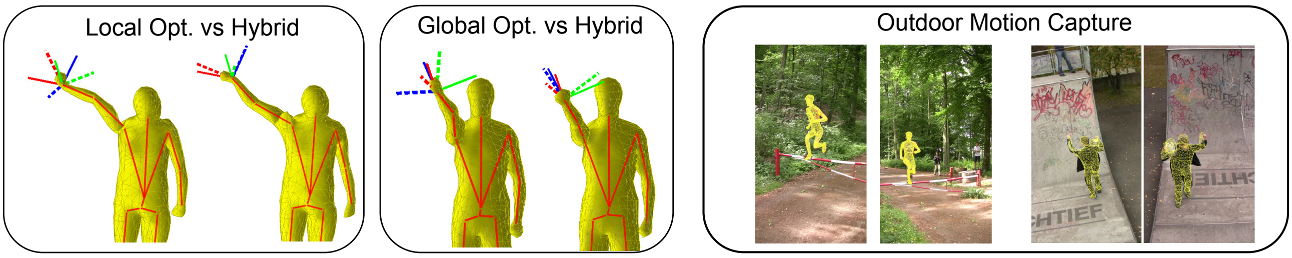


Figure 2.17: left: current approaches, either local or global optimization can not capture axial rotation accurately; the combination of a video based tracker with inertial sensors (hybrid tracker) allows to capture detailed motion. Right: motion capture in uncontrolled scenarios is one of the principal future challenges (outdoor images on the right are courtesy of Hasler *et al.*[118])

The process noise $\mathbf{w}$ is assumed to be zero mean, with covariance Σ, *i.e.*, has a distribution $\mathcal{N}(\mathbf{w}; 0, \Sigma)$. Equivalently, it follows that $p(\mathbf{x}_{t+1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+1}; \mathbf{x}_t, \Sigma)$. More generally one can use an L-th order linear-Gaussian Dynamical system LDS of the form:

$$\mathbf{x}_{t+1} = \sum_{\tau=1}^{L} \mathbf{A}_\tau \mathbf{x}_{t-\tau+1} + \mathbf{w} \tag{2.91}$$

The matrices $\mathbf{A}_\tau$ can be set by hand, *e.g.*, assuming them to be diagonal and fixing the process noise covariance to be also diagonal. Alternatively, one can learn dynamical models from MoCap data [117]. This allows to capture coupling between joints. The problem is that the number of parameters to learn is quadratic with the dimension of the pose parameters which leads to over-fitting. To overcome this, it is also possible to learn a low dimensional latent paramterization using dimensionality reduction techniques such as PCA, GPLVM or GPDM. Then one can learn the dynamics on the latent space [16, 17]. A common problem is that the dynamics can vary a lot depending on the motion pattern. One way around this is to use switching linear dynamical models SLDS.For more details on motion models for people tracking, we refer the reader to [3].

2.8 Limitations of Pure Image Based Methods

The basic mathematical tools necessary for anyone who wants to implement a human pose estimation system have been introduced, namely kinematic structure representation and model creation. A unified formulation has been presented for the most common model-based pose estimation algorithms seen in the literature. We have seen that the model image association problem for pose estimation is usually formulated as the minimization/maximization of an error/likelihood function. The two main strategies have been described, namely local and particle based optimization. Local optimization methods are faster and more accurate but in practice, if there are visual ambiguities, or really fast motions, the tracker might fail catastrophically. To achieve more robustness, particle filters can be used because they can represent uncertainty through a rigorous Bayesian paradigm. The problem here is that the number of particles needed to achieve reasonable results grows exponentially with the dimensionality of the pose parameter space which makes them unpractical for human models with more than 20 DoF. To reduce the number of needed particles, the annealed particle filter can be used at the expense of not being able to work in a fully Bayesian framework. A further problem of global optimization methods is that while robust, they do not provide a single temporally consistent motion but rather a jittery motion which must be post-processed to obtain visually acceptable results. Combinations of global and local optimization have also been proposed to compensate for the drawbacks of each strategy [68, 113, 5]. Nonetheless, current approaches do not capture detailed movement such as hand orientation or axial arm rotation. This stems from depth and orientation ambiguities inherent in the images (*i.e.*, any rotation about a limb axis projects to the same image). To overcome this limitations we propose a hybrid tracker that combines correspondence based local optimization with a small set of inertial sensors placed at body extremities to obtain a much accurate and detailed human tracker, see Figure 2.17. The key idea is to fuse both sensor types in a joint optimization to exploit the advantages of each sensor type (accurate position from images and accurate orientation from inertial sensors). This will be explained in the next Chap. 4 and Chap. 5.

3 Inertial Sensors and Camera Calibration

3.1 Camera Model

We use a perspective camera projection model. A projection model maps points in the 3D world to pixel locations in the image. Mathematically, the image formation can be defined as the projection of the 3D space onto the image plane, see Fig. 3.1.

A point in world coordinates $\bar{\mathbf{p}}$ is projected to a 2D point $\mathbf{r} = [u\ v]^T$ in the image with the following equation

$$\bar{\mathbf{r}} = \frac{1}{Z}\mathbf{M}\bar{\mathbf{p}} \tag{3.1}$$

$$\tag{3.2}$$

where Z is the depth of point, $\bar{\mathbf{r}} = [u,v,1]^T$, and $\bar{\mathbf{p}} = [X,Y,Z,1]^T$ are the homogenous coordinates of the points and $\mathbf{M}$ is the camera projection matrix where

$$\mathbf{M} = \mathbf{K}\left[\mathbf{R}|\mathbf{t}\right]. \tag{3.3}$$

where

- $\mathbf{K}$ is the 3×3 camera calibration matrix that depends on the internal parameters of the camera: focal length and distortion parameters of the lenses.
- $[\mathbf{R}|\mathbf{t}]$ is the 3×4 external camera matrix that contains the rigid transformation between the world coordinate system and the camera coordinate system.

The image formation process is illustrated in Fig. 3.1. Let $\mathbf{m_1}$,$\mathbf{m}_2$,$\mathbf{m}_3$ be the rows of $\mathbf{M}$. Since the depth $Z = \mathbf{m}_3^T\mathbf{p}_s$ we can alternatively write

$$u = \frac{\mathbf{m}_1^T\mathbf{p}_s}{\mathbf{m}_3^T\mathbf{p}_s} \tag{3.4}$$

$$v = \frac{\mathbf{m}_2^T\mathbf{p}_s}{\mathbf{m}_3^T\mathbf{p}_s} \tag{3.5}$$

3.1.1 Internal Camera Matrix

Given a point in camera coordinates, $\mathbf{p}_c = [X,Y,Z]^T$, its projection $\mathbf{r} = [u\,v]^T$ in the image is given by

$$u = k_u f\frac{X}{Z} + u_0 \tag{3.6}$$

$$v = k_v f\frac{Y}{Z} + v_0 \tag{3.7}$$

where

- f is the focal length of the camera.

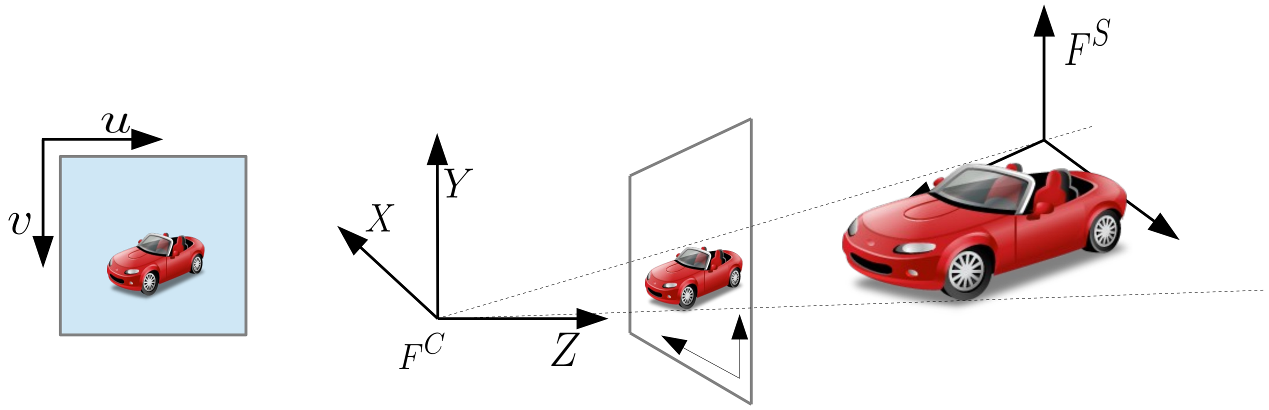


Figure 3.1: Camera perspective model on the right, formed image on the left. The following transformations take place: 1) From world coordinates to camera coordinates, 2) perspective projection onto the image plane, 3) from image plane coordinates to pixels. The scaling factor and the principal point determine how plane coordinates are mapped to the final pixels. The final image in pixels is shown on the left.

- where k_u,k_v are the number of pixels per unit distance in the normalized plane (u,v) in the u- and v- directions respectively.
- $\mathbf{c} = [u_0,v_0]^T$ is called the *principal point* which is the intersection between the optical axis and the image plane expressed in image coordinates.

This can be expressed in matrix form as $\bar{\mathbf{r}} = \frac{1}{Z}\mathbf{K}\mathbf{p}_c$, where

$$\mathbf{K} = \begin{bmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

is the intrinsics camera matrix. We assume the cameras have no skew parameters, additionally if the pixels are assumed to be square we have that k_u and k_v are equal.

3.1.2 Extrinsics Camera Matrix

The 3×4 external parameters simply represents the transformation from camera to world coordinate system. That is, a point in the world coordinate $\mathbf{p}_s$ is mapped to a point in camera coordinates $\mathbf{p}_c$ through

$$\mathbf{p}_s = [\mathbf{R}|\mathbf{t}]\bar{\mathbf{p}}_c = \mathbf{R}\mathbf{p}_c + \mathbf{t} \tag{3.9}$$

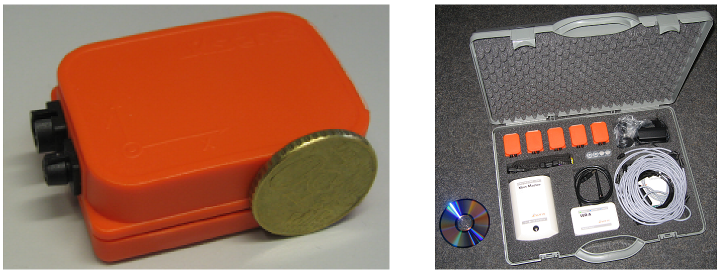
From this relation it is straight-forward to extract the *camera center* $\mathbf{o}_{\text{center}}$ in world coordinates as $\mathbf{o}_{\text{center}} = -\mathbf{R}^T\mathbf{t}$.

3.1.3 Camera Calibration

The parameters of the camera projection M are estimated from point correspondences between known world points $\mathbf{p}_i$ and their corresponding 2d image coordinates $\mathbf{r}_i$. Clearing the denominators in Eq. 3.5, we see that every correspondence leads to the following linear equation

$$(\mathbf{m}_1^T - u_i\,\mathbf{m}_3^T) \cdot \mathbf{p}_i = 0 \tag{3.10}$$
$$(\mathbf{m}_2^T - v_i\,\mathbf{m}_3^T) \cdot \mathbf{p}_i = 0 \tag{3.11}$$

Collecting enough correspondences one can estimate M from the overconstrained linear system. The parameters can then be infered from the projection matrix, see [97].



(a) (b)

Figure 3.2: IMUS: (a) XSens inertial measurement unit and (b) Xbus Kit consisting of Xbus Master and 5 units. The master synchronizes the 5 sensors with a trigger signal, collects the raw data from the units and sends it to a computer.

3.2 IMUs

An Inertial Measurement Unit or IMU is an electronic device that measures and orientation and gravitational forces, using a combination of accelerometers and gyroscopes, sometimes also magnetometers. In all our experiments we use the MTx device provided by XSens [119]. An MTx unit consists of 3 rate gyroscopes, a magnetometer and an accelerometer. This information is fused using a proprietary algorithm to provide 3D acceleration and 3D orientation. The unit's dimensions are 38x53x21 mm and weights 30g. The accuracy of the orientation is lower than 1 deg with an angular resolution of 0.05 deg[1]. The maximum frame rate is 512 Hz provided that all the bandwidth is used by a single sensor. In our experiments, since we are using up to 10 units we record at a frame-rate of 40 Hz.

3.2.1 Sensor Data

In our experiments, we use an orientation estimation device MTx provided by XSens [119]. An Xsens MTx unit provides two different streams of data: three dimensional local linear acceleration $\vec{a}^S$ and local rate of turn or angular velocity ω^S. In Fig. 3.2 we show an MTx unit (a) and the full set including the XBus master (b) which collects data and synchronizes several IMUs. Orientation data can be obtained from the angular velocity $\omega(k)$ provided by the sensor units. Besides angular velocity, the MTx units provide a proprietary algorithm that can accurately calculate absolute orientations relative to a static global frame F^I, which we will refer to as inertial frame. The inertial frame F^I is computed internally in each of the sensor units in an initial static position and is defined as follows: The Z axis is the negative direction of gravity measured by the internal accelerometer. The X axis is the direction of the magnetic north pole measured by a magnetometer. Finally, the Y axis is defined by the cross product $Z \times X$. For each sensor, the absolute orientation data is provided by a stream of quaternions that define, at every frame, the map or coordinate transformation from the local sensor coordinate system to the global one $\mathbf{R}^{IS}(t) : F^S \Rightarrow F^I$. In Fig. 3.3 we show how the 5 sensors are attached to the extremities, we also show the local coordinate frame of each unit that coincides with the box main axes.

3.2.2 Limitations of IMUs

IMUs are very appealing because they provide a direct $3D$ measurement in contrast to images where the $3D$ information needs to be hallucinated. Furthermore, IMUs do not suffer from occlusions/self-occlusions. However, they have some limitations:

---

[1] Specifications provided by the manufacturer

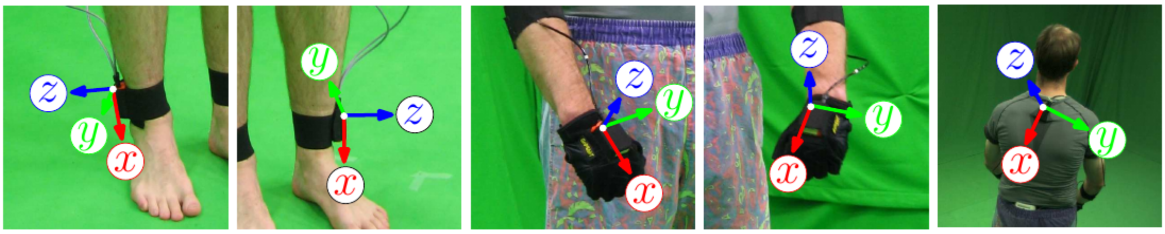


Figure 3.3: Sensor placement: The five sensors are attached at the body extremities. We show the local coordinate frame axes of each of the five sensors used.

- Wearing many IMU is intrusive for the subject.
- IMUs suffer from drift in continous operation. The lack of direct positional information makes IMUs inaccurate. One could in principle derive position from the acceleration measurements but we have found this to be numerically very unstable.
- The orientation data provided by IMUs suffers from lag. That is due to the fact that orientation data is obtained as the output of a Kalman filter that integrates accleration, magnetetometer and gyroscope information together. This is lag is specially problematic during fast motions.

Hence, we introduce in Chap. 4 a hybrid tracker that fuses information coming from a small set of IMUs (we use 5) and information coming from video cameras to compensate for the drawbacks of each sensor type. To this end, we synchronize a set of video cameras with a set of IMUs as explained below.

3.3 Calibration and Synchronization with the Video Cameras

Unfortunately, the world frame defined in our tracking system differs from the global inertial frame. The tracking coordinate frame F^T is defined by a calibration cube placed in the recording volume, in contrast to the inertial coordinate frame which is defined by the gravity and magnetic north directions. Therefore, in order to be able to integrate the orientation data from the inertial sensors into our tracking system, we must know the rotational offset $\mathbf{R}^{TI}$ between both worlds, see Figure 3.4. Since the Y axis of the cube is perpendicular to the ground and so is gravity, the Y axis of the tracking frame and the Z axis of the inertial frame are aligned. Therefore, $\mathbf{R}^{TI}$ is a one parametric planar rotation that can be estimated beforehand using a calibration sequence [58]. Thus, we can easily adapt the transformations so that they define a map from the local sensor frame to the tracking frame F^T:

$$\mathbf{R}^{TS} = \mathbf{R}^{TI}\mathbf{R}^{IS} \tag{3.12}$$

This calibration step can be avoided if the global tracking frame coincides with the inertial global frame. This can be achieved by placing a calibration cube on the ground so that one of the axis is aligned with gravity direction and orientating the cube so that its X-axis aligns with the magnetic north. This can easily be accomplished using a compass. The 3D coordinates of 21 keypoints in the cube are known. By clicking those points in the images we obtain a set of 3D points with its corresponding 2D image projections from which we obtain the camera projection matrices [97]. In Fig. 3.5 we show the calibration cube placement to calibrate four cameras in an outdoor setup.

The MTx units are synchronized with a trigger signal that comes from the master, see Fig. 3.2. The cameras are synchronized with a trigger signal in the indoor setups and synchronized using the audio signals in outdoor setups where we use conventional cameras. To synchronize the cameras with the IMU we start the sequences using a clapping motion. The clapping produces a clear peak in both the camera audio signals and the IMU's acceleration signal which can be easily detected.

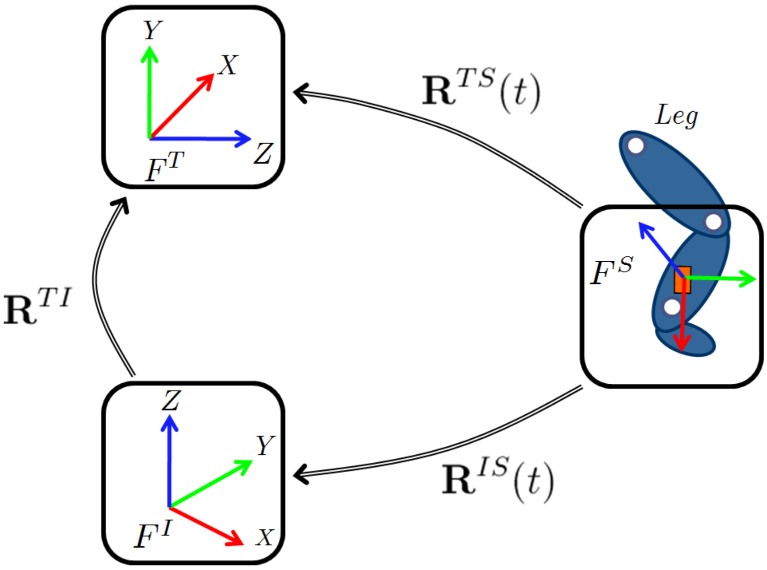
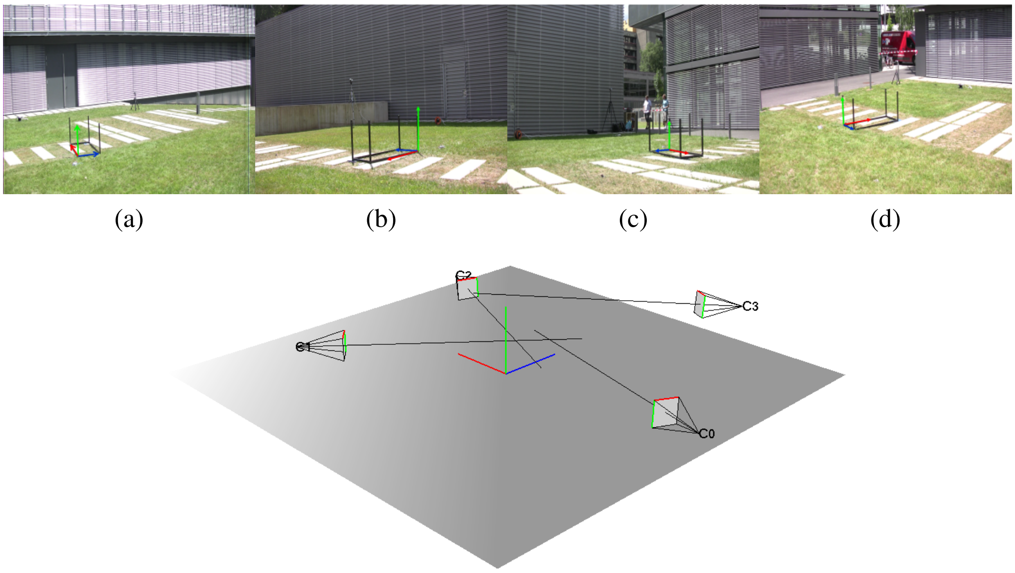


Figure 3.4: Global frames: tracking frame F^T and inertial frame F^I. Local frame: sensor frame F^S.



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

(e)

Figure 3.5: Camera calibration: by placing a calibration cube on the ground and aligning one of its axis to the magnetic north we obtain a tracking coordinate system that coincides with the global coordinate system defined by XSens sensors. In (a),(b),(c),(d) we show each camera view with the projected coordinate system on the image plane. In (e) we show the arrangement of the cameras calculated from calibration.

4 Local Hybrid Tracker

We describe in this chapter an approach to fuse video with orientation data obtained from extended inertial sensors to improve and stabilize full-body human motion capture [49]. Even though the method video data is a strong cue for motion analysis, tracking artifacts occur frequently due to ambiguities in the images, rapid motions, occlusions or noise. As a complementary data source, inertial sensors allow for drift-free estimation of limb orientations even under fast motions. However, accurate position information cannot be obtained in continuous operation. As explained in the previous chapter, we propose a hybrid tracker that combines video with a small number of inertial units, see Fig. 4.1, to compensate for the drawbacks of each sensor type. On the one hand, we obtain drift-free and accurate position information from video data and, on the other hand, we obtain accurate limb orientations and good performance under fast motions from inertial sensors. In several experiments we demonstrate the increased performance and stability of our human motion tracker.

4.1 Introduction

As described in Sec. 2.7 in Chap. 2, one way to overcome image ambiguities is to include prior knowledge, such as human motion constraints. There are several ways to employ such a priori knowledge to human tracking. One option is to learn the space of plausible human poses and motions [19, 9, 120, 121, 12, 28]. Another option is to learn a direct mapping from image features to the pose space [30, 122, 28, 123]. To constrain the high dimensional space of kinematic models, a major theme of recent research on human tracking has been dealing with dimensionality reduction [16, 17]. Here, the idea is that a typical motion pattern like walking should be a rather simple trajectory in a lower dimensional manifold. Therefore, prior distributions are learned in this lower dimensional space. Such methods are believed to generalize well with only little training data. Inspired by the same ideas of dimensionality reduction, physical and illumination models have been recently proposed to constrain and to represent human motion in a more realistic way [18, 9, 10, 124]. A current trend of research tries to estimate shape deformations from images besides the body pose by either directly deforming the mesh geometry [85] or by a combination of skeleton-based pose estimation with surface deformation [125].

Even using learned priors from MoCap data, obtaining limb orientations from video is a difficult problem. Intuitively, because of the cylindrical shape of human limbs, different limb orientations project to very similar silhouettes in the images. These orientation ambiguities can be easily captured by the inertial sensors but accurate positions cannot be obtained. Therefore, we propose to use a small number of sensors (we use only five) fixed at the body extremities (neck, wrists and ankles) as a complementary data source to visual information. One the one hand, we obtain stable and drift-free accurate position information from video data and, on the other hand, we obtain accurate limb orientations from the inertial sensors. We show how to integrate orientation data from sensors in a contour-based video motion capture algorithm. In several experiments, we show the improved performance of tracking with additional small number of sensors.

Recently, inertial sensors (e.g. gyroscopes and accelerometers) have become popular for human motion analysis. Often, sensors are used for medical applications, see, e. g., [36] where accelerometer and gyroscope data is fused. However, their application concentrates on the estimation of the lower limb orientation in the sagittal plane. In [37], a combination of inertial sensors and visual data is restricted to the tracking of a single limb (the arm). Moreover, only a simple red arm band is used as

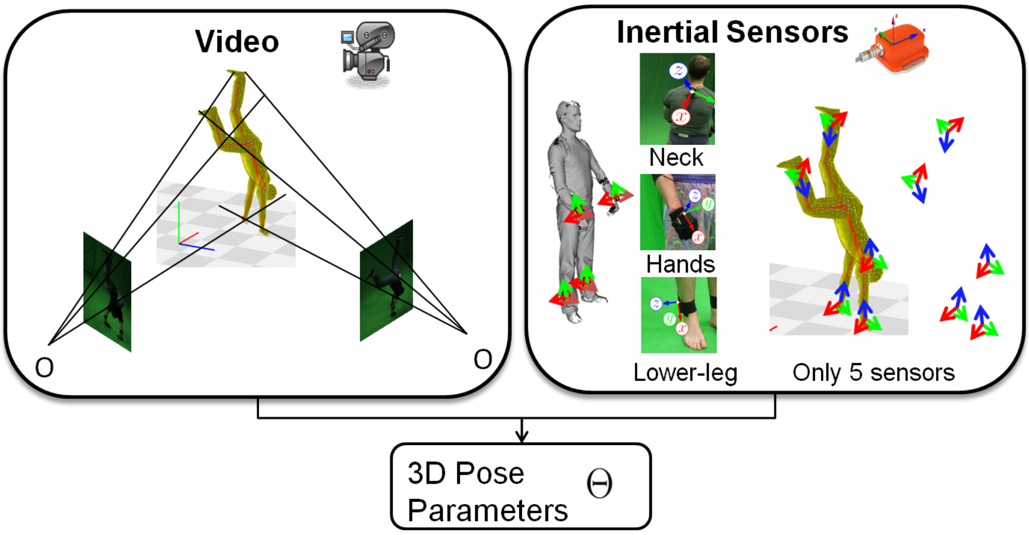


Figure 4.1: Hybrid local tracker: Information coming from video cameras and the orientation data coming from 5 IMU sensors attached at the body extremities is combined to obtain high accurate pose estimation results. We formulate an energy that measures the pose consistency with the image observations as well as the IMUs orientations. Since in indoor scenarios good quality segmentations are possible we opt here to fuse the information using a local optimization method which is very efficient.

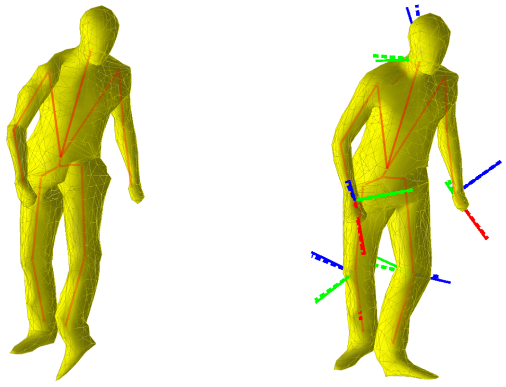


Figure 4.2: Tracking result for one frame using (a) Video-based tracker. (b) Our proposed hybrid tracker.

image feature. In [38], data obtained from few accelerometers is used to retrieve and play back human motions from a database. [126] presents a system to capture full-body motion using only inertial and magnetic sensors. While the system in [126] is very appealing because it does not require cameras for tracking, the subject has to wear a suit with at least 17 inertial sensors, which might hamper the movement of the subject. In addition, long preparation time before recording is needed. Moreover, inertial sensors suffer from severe drift problems and cannot provide accurate position information in continuous operation.

4.1.1 Pose Parameterization

We very breifly recall the basic foundations to model human motion using a kinematic chain, for more details see Sec. 2.1.

The dynamics of the subject are modeled by a *kinematic chain* $\mathcal{F}$, which describes the motion constraints of an articulated rigid body such as the human skeleton. The underlying idea behind a kinematic chain is that the motion of a body segment is given by the motion of the previous body segment in the chain and an angular rotation around a joint axis. Specifically, the kinematic chain is defined with a 6 DoF (degree of freedom) root joint representing the global rigid body motion and a set of 1 DoF revolute joints describing the angular motion of the limbs. Joints with higher degrees of freedom like hips or shoulders are represented by concatenating two or three 1 DoF revolute joints. The root joint is expressed as a twist of the form $\theta\xi$ with the rotation axis orientation, location, and angle as free parameters. Revolute joints are expressed as special twists with no pitch of the from $\theta_j\xi_j$ with known ξ_j (the location and orientation of the rotation axis as part of the model representation). Therefore, the full configuration of the kinematic chain is completely defined by a $(6+n)$ vector of free parameters

$$\mathbf{x} := (\theta\xi, \theta_1, \ldots, \theta_n) \tag{4.1}$$

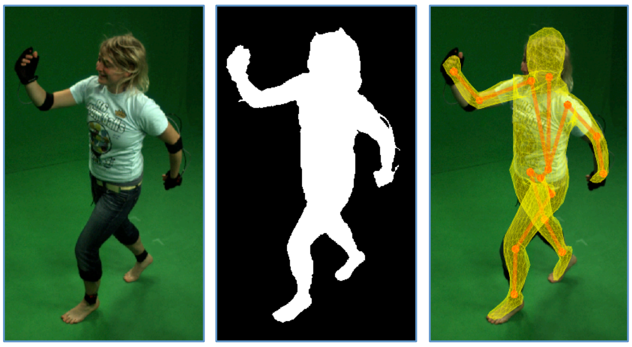
similar to [77]. Now, for a given point $\mathbf{p} \in \mathbb{R}^3$ on the kinematic chain, we define $\mathcal{J}(\mathbf{p}) \subseteq \{1, \ldots, n\}$ to be the ordered set that encodes the joint transformations influencing $\mathbf{p}$. Let $\bar{\mathbf{p}}_s = \frac{\mathbf{p}}{1}$ be the homogeneous coordinate of $\mathbf{p}$ and denote $\mathcal{P}_c()$ as the associated projection with $\mathcal{P}_c(\bar{\mathbf{p}}) = \mathbf{p}$. Then, the transformation of a point $\mathbf{p}$ using the kinematic chain $\mathcal{F}(\mathbf{x};\mathbf{p})$ and a parameter vector $\mathbf{x}$ is defined by

$$\mathcal{F}(\mathbf{x};\mathbf{p}) = \mathcal{P}_h\left(\mathbf{G}^{TB}(\mathbf{x})\bar{\mathbf{p}}_s(0)\right) = \mathcal{P}_h\left(\left(\exp(\theta\hat{\xi}) \prod_{j\in\mathcal{J}(x)} \exp(\theta_j\hat{\xi}_j)\right)\bar{\mathbf{p}}_s(0)\right). \tag{4.2}$$

Here, $\mathcal{F}(\mathbf{x};\mathbf{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is the function representing the total rigid body motion $\mathbf{G}^{TB}(\mathbf{x})$ of the segment in the chain where $\mathbf{p}$ belongs. Equation (4.2) is commonly known as the *product of exponentials formula* [61], denoted as $\mathcal{F}(\mathbf{x};)$.

4.2 Video-based Tracker

The video based tracker is correspondence based as explained in Sec. 2.6.1. In order to relate the surface model to the human's images we find correspondences between the 3D surface vertices and the 2D image contours obtained with background subtraction, see Fig. 4.3.



(a) (b) (c)

Figure 4.3: (a) Original Image, (b) Background subtracted image, (c) Projected surface mesh after convergence.

We first collect 2D-2D correspondences by matching the projected surface silhouette with the background subtracted image contour as described in Sec. 2.4 and Sec. 2.6. Thereby, we obtain a collection of 2D-3D correspondences since we know the 3D counterparts of the projected 2D points of the silhouette. In the presented experiments we only use the silhouettes as image features.We then

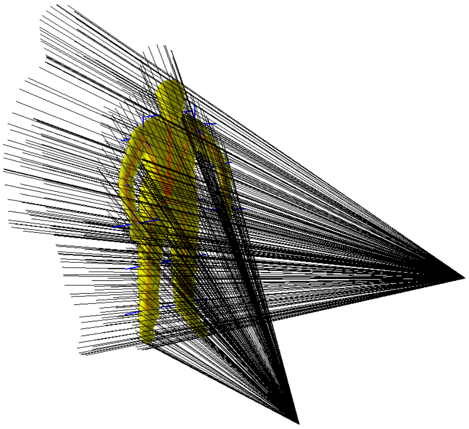


Figure 4.4: Energy corresponding to video data: we minimize the sum of squared distances between model points and the projection rays of their corresponding 2D correspondences.

minimize the distance $\mathbf{e}_i$ between the transformed 3D points $\mathcal{F}(\mathbf{x};\mathbf{p}_i(0))$ and the projection rays defined by the 2D points $\mathbf{p}_i$. This gives us a point-to-line constraint for each correspondence. Defining $L_i = (\mathbf{n}_i,\mathbf{m}_i)$ as the 3D Plücker line with unit direction $\mathbf{n}_i$ and moment $\mathbf{m}_i$ of the corresponding 2D point $\mathbf{r}_i = [x_i,y_i]$, the point to line distance residual $\mathbf{e}_i$ can be expressed as

$$\mathbf{e}_i = \|\mathcal{F}(\mathbf{x};\mathbf{p}_i) \times \mathbf{n}_i - \mathbf{m}_i\| \tag{4.3}$$

Similar to Bregler *et al.*[76] we now linearize the Equation by using $\exp(\theta\widehat{\xi}) = \sum_{k=0}^{\infty} \frac{(\theta\widehat{\xi})^k}{k!}$. With $\mathbf{I}$ as identity matrix, this results in

$$(\mathbf{I} + \Delta\xi + \sum_{j\in\mathcal{J}(x)} \Delta\theta_j \widehat{\xi'_j}))\, \mathbf{p}_i(\mathbf{x})) \times \mathbf{n}_i - \mathbf{m}_i = 0\,. \tag{4.4}$$

Having N correspondences, the energy we minimize E_{video} is the sum of squared point-to-line distances $\mathbf{e}_i$

$$\arg\min_{\mathbf{x}} E_{\text{video}}(\mathbf{x}) = \sum_{i=1}^{N} \|d_i\|^2 = \arg\min_{\mathbf{x}} \sum_{i=1}^{N} \|\mathcal{F}(\mathbf{x};\mathbf{p}_i) \times \mathbf{n}_i - \mathbf{m}_i\|^2 \tag{4.5}$$

which can be locally optimized. After linearization, Eq. (4.5) can be re-ordered into an equation of the form $\mathbf{J}_{\text{video}}(\mathbf{x})\Delta\mathbf{x} = \mathbf{e}_{\text{video}}$ as explained in Sec. 2.6.1, see also Fig. 4.5. Collecting a set of such equations leads to an over-determined system of equations, which can be solved using numerical methods like the Householder algorithm. The pose parameters are then updated as $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}$. The Rodriguez formula can be applied to reconstruct the group action g from the estimated twists $\theta_j\xi_j$. Then, the 3D points can be transformed and the process is iterated until convergence as explained in Sec. 2.6.

4.3 Hybrid Tracker

The input of our tracking system consists of:

- Rigid surface mesh of the actor obtained from a laser scanner (Sec. 2.3).
- Multi-view images obtained by a set of calibrated and synchronized cameras (Sec. 3.3).
- Global orientation data coming from the sensors (Sec. 3.2.1).

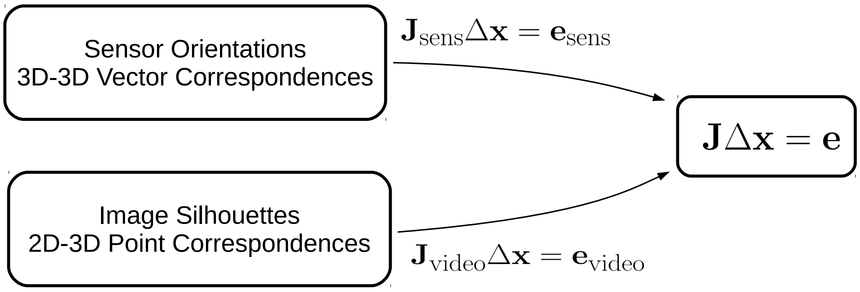

Sensor Orientations
3D-3D Vector Correspondences
$\mathbf{J}_{\text{sens}}\Delta\mathbf{x} = \mathbf{e}_{\text{sens}}$
$\mathbf{J}\Delta\mathbf{x} = \mathbf{e}$
Image Silhouettes
2D-3D Point Correspondences
$\mathbf{J}_{\text{video}}\Delta\mathbf{x} = \mathbf{e}_{\text{video}}$

Figure 4.5: We optimize a joint energy including a term for the video data and a term for the IMU sensor data. The IMU orientation constraints are integrated as 3D-3D vector correspondences and the video data as 2D-3D point correspondences.

We used five inertial sensors fixed at the body extremities (wrists, lower legs, and neck). We define a joint energy E_{hybrid} that measures the consistency between poes estimates with measurements coming from video and inertial sensors:

$$\arg\min_{\mathbf{x}} E_{\text{hybrid}}(\mathbf{x}) = E_{\text{video}}(\mathbf{x}) + \lambda E_{\text{sens}}(\mathbf{x}) \tag{4.6}$$

where $E_{\text{video}}(\mathbf{x})$ is the energy cost corresponding to the video measurements defined in Eq. 4.5 and $\lambda E_{\text{sens}}(\mathbf{x})$ is the cost associated with the IMU orientation measurements. To have a balanced energy we normalize the individual terms in the range of $[0,1]$. In all our experiments we set $\lambda = 0.5$ to have an equal influence of both terms. As we will see in Sec.4.4.2, $E_{\text{sens}}(\mathbf{x})$ can also be expressed as a sum of squared errors. This allows us to use numerical optimization techniques such Newton-Raphson or Lebendberg-Marquadt. Let $\mathbf{e}_{\text{video}} : \mathbb{R}^D \mapsto \mathbb{R}^{3N}$ be the vector valued function of residuals of image correspondences and $\mathbf{e}_{\text{sens}} : \mathbb{R}^D \mapsto \mathbb{R}^{9N_s}$ the function of orientation residuals, where N_s is the number of available sensors. Now, we can express the energy in Eq. (4.6) as

$$\arg\min_{\mathbf{x}} \mathbf{e}^T_{\text{hybrid}}(\mathbf{x})\mathbf{e}_{\text{hybrid}}(\mathbf{x}) = \mathbf{e}_{\text{video}}(\mathbf{x})\mathbf{e}_{\text{video}}(\mathbf{x}) + \sqrt{\lambda}\mathbf{e}^T_{\text{sens}}(\mathbf{x})\sqrt{\lambda}\mathbf{e}_{\text{sens}}(\mathbf{x}). \tag{4.7}$$

Eq. (4.7) is then iteratively linearized and the step $\Delta\mathbf{x}$ is found by solving the following linear system

$$\begin{bmatrix} \mathbf{J}_{\text{video}}(\mathbf{x}) \\ \sqrt{\lambda}\,\mathbf{J}_{\text{sens}}(\mathbf{x}) \end{bmatrix} \Delta\mathbf{x} = \begin{bmatrix} \mathbf{e}_{\text{video}}(\mathbf{x}) \\ \sqrt{\lambda}\,\mathbf{e}_{\text{sens}}(\mathbf{x}) \end{bmatrix} \tag{4.8}$$

The pose parameters are then updated as $\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x}$. The term corresponding to the video data is explained in the previous section. The term for the IMU is explained in Sec. 4.4.2.

4.4 Integration of Sensor Data

4.4.1 Sensor Data

In our experiments, we use an orientation estimation device MTx provided by XSens [119]. As explained in Chap. 3.2 each MTx unit provides the orientation of the local coordinate system w.r.t. a global and is dennoted as $\mathbf{R}^{TS}$.

4.4.2 Integration of Orientation Data into the Video-based Tracker

In this section we explain how to integrate the orientation data from the sensors as additional equations that can be appended into the big linear system, see Figure 4.5. Here we have to be very careful and know, at all times, in which frame the rotation matrices are defined. Three coordinate systems are involved: the global tracking frame F^T, the body frame F^B (the local frame of a segment in the chain, e.g. the leg), and the sensor frame F^S. Recall from Sect. 4.4.1 that the orientation data is given as a rotation matrix $\mathbf{R}^{TS}(t) : F^S \to F^T$ defining the transformation from the local sensor frame F^S to the global tracking frame F^T, which we will refer to as *ground-truth orientation*. In order to relate the orientation data to the differential twist parameters $\mathbf{x}$, we will compare the *ground-truth orientations* $\mathbf{R}^{TS}(t)$ of each of the sensors with the estimated sensor orientations from the tracking procedure $\hat{\mathbf{R}}^{TS}(\mathbf{x})$, which we will denote as *tracking orientation*. Let $\mathbf{R}^{TS}$ be the 3×3 rotation matrix of the *ground-truth orientation* $\mathbf{R}^{TS}$ (quaternions can be easily transformed to rotation matrices [62]). The columns of the rotation matrix $\mathbf{R}^{TS}$ are simply the sensor basis axes in world coordinates. Let us also define $\mathbf{R}^{TB}(\mathbf{x}(t))$ as the total accumulated motion of a body segment at time t, *i.e.*, $\mathbf{R}(\mathbf{x}_t) : F^B \to F^T$. For the sake of clarity we will drop the time subindex $\mathbf{x}_t$ and just write $\mathbf{R}^{TB}(\mathbf{x})$. The transformation from the sensor frame to the body frame $\mathbf{R}^{BS}(t) : F^S \to F^B$ is constant during tracking because the sensor and body frame are rigidly attached to the body segment and move together. Thus, we can compute this rotational displacement $\mathbf{R}^{BS}$ in the first frame

$$\mathbf{R}^{BS} = \mathbf{R}^{TB}(0)^{-1}\mathbf{R}^{TS}(0)\,, \tag{4.9}$$

where $\mathbf{R}^{TB}(0)$ is the configuration of the body part B in the first frame. For every sensor we can minimize the angular distance, see Sec. 2.1.7, between the rotation given by the sensor measurement $\mathbf{R}^{TS}(t)$ and the rotation estimated by our tracking system $\hat{\mathbf{R}}^{TS}(\mathbf{x}) = \mathbf{R}^{TB}(\mathbf{x})\mathbf{R}^{BS}$

$$\arg\min_{\mathbf{x}} \|\mathbf{R}^{TB}(\mathbf{x})\mathbf{R}^{BS} - \mathbf{R}^{TS}(t)\|_{\mathrm{F}}^2 = \|\mathbf{R}^{TB}(\mathbf{x}) - \mathbf{R}^{TS}(t)\mathbf{R}^{SB}\|_{\mathrm{F}}^2 \tag{4.10}$$

where we have used the fact that the Frobenius norm is bi-invariant, see Sec. 2.1.7 . Similarly as we did for point correspondences we can linearize. Here we have to be more careful since we have to compute derivatives of rotation matrices. Fortunately, those can be expressed in terms of twists. Linearizing Eq. (4.10) we can find the optimal step $\Delta\mathbf{x}$

$$\arg\min_{\Delta\mathbf{x}} \|\mathbf{R}^{TB}(\mathbf{x}) + \frac{\Delta\mathbf{R}^{TB}(\mathbf{x})}{\Delta\mathbf{x}}\Delta\mathbf{x} - \mathbf{R}^{TS}(t)\mathbf{R}^{SB}\|_{\mathrm{F}}^2 \tag{4.11}$$

Let $\mathbf{J}_{\text{ori}} : \mathbb{R}^D \mapsto so(3)$ be the Jacobian of the orientation forward kinematics map $\mathcal{F} : \mathbb{R}^D \mapsto SO(3)$. The orientation Jacobian maps an increment in parameter space to the equivalent screw $\omega_{\text{eq}} \in so(3)$ of the rigid motion. Now by expressing the rotation derivatives in terms of the associated screw, we obtain the following expression, see Sec. 2.1.5,

$$\frac{\Delta\mathbf{R}^{TB}}{\Delta\mathbf{x}}\Delta\mathbf{x} = \widehat{\omega}_{\text{eq}}\mathbf{R}(\mathbf{x})^{TB} = [\mathbf{J}_{\text{ori}}\Delta\mathbf{x}]^{\wedge}\,\mathbf{R}^{TB}(\mathbf{x}) \tag{4.12}$$

which we can substitute back in the original objective function Eq. (4.11)

$$\arg\min_{\Delta\mathbf{x}} \|\mathbf{R}^{TB}(\mathbf{x}) + [\mathbf{J}_{\text{ori}}\Delta\mathbf{x}]^{\wedge}\,\mathbf{R}^{TB}(\mathbf{x}) - \mathbf{R}^{TS}(t)\mathbf{R}^{SB}\|_{\mathrm{F}}^2. \tag{4.13}$$

Since this is essentialy a least squares problem, the step can be found by solving the following linear equations

$$[\mathbf{J}_{\text{ori}}\Delta\mathbf{x}]^{\wedge}\,\mathbf{R}^{TB}(\mathbf{x}) = \mathbf{R}^{TS}(t)\mathbf{R}^{SB} - \mathbf{R}^{TB}(\mathbf{x}) \tag{4.14}$$

which can be re-arranged as 9 linear equations of the form

$$\left(\Delta\omega' + \sum_{j \in \mathcal{J}(\mathbf{x})} \Delta\theta_j \widehat{\omega}_j'\right) \mathbf{r}_1^{TB}(\mathbf{x}) = \mathbf{r}_1^{TS}(t)\mathbf{r}_1^{SB} - \mathbf{r}_1^{TB}(\mathbf{x}) \tag{4.15}$$

$$\left(\Delta\omega' + \sum_{j \in \mathcal{J}(\mathbf{x})} \Delta\theta_j \widehat{\omega}_j'\right) \mathbf{r}_2^{TB}(\mathbf{x}) = \mathbf{r}_2^{TS}(t)\mathbf{r}_2^{SB} - \mathbf{r}_2^{TB}(\mathbf{x}) \tag{4.16}$$

$$\left(\Delta\omega' + \sum_{j \in \mathcal{J}(\mathbf{x})} \Delta\theta_j \widehat{\omega}_j'\right) \mathbf{r}_3^{TB}(\mathbf{x}) = \mathbf{r}_3^{TS}(t)\mathbf{r}_3^{SB} - \mathbf{r}_3^{TB}(\mathbf{x}) \tag{4.17}$$

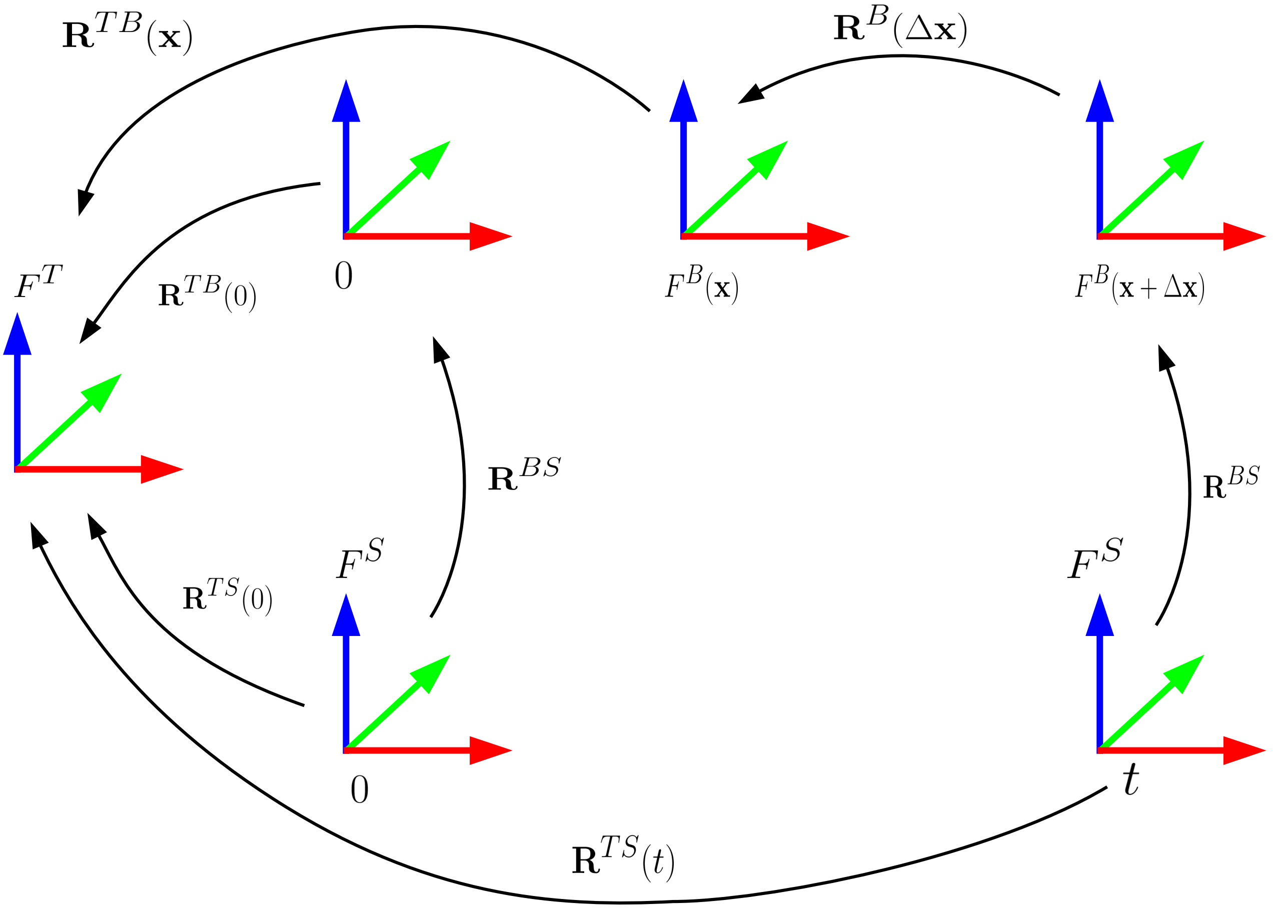


Figure 4.6: Integration of orientation data into the video-based tracker. *Ground-truth orientation*: clockwise down path from F^S at time t to F^T. *Tracking orientation*: anti-clockwise upper path from F^S at time t to F^T.

where $\mathbf{r}_i$ denotes the i-th column of matrix $\mathbf{R}$. This last equations can more conveniently be expressed in matrix form as $\mathbf{J}_{\text{sens}}(\mathbf{x};\mathbf{r}_i^{TB})\Delta\mathbf{x} = \mathbf{e}_{\text{sens,i}}$ for $i = \{1\dots3\}$. Here $\mathbf{J}_{\text{sens}} : \mathbb{R}^D \mapsto \mathbb{R}^3$ has almost the same structure as the positional pose Jacobian except that it does not depend on the translational motion nor the location of the joints. Note also that it takes a vector $\mathbf{r}$ as input as opposed to a point. Also, note the difference between $\mathbf{J}_{\text{sens}}$ which are the derivatives of a rotating vector $\mathbf{r}$ and is therefore local, and $\mathbf{J}_{\text{ori}}$ which maps to the tangential space $so(3)$.

4.4.3 A Geometric Derivation

The previous derivation allows us to integrate the additional equations which are then linearized and appended with the equations corresponding to the cost associated with the image evidence. It is however interesting to derive the same equations form a more geometric point of view. Consider the local infinitesimal rotation $\mathbf{R}^B(\Delta\mathbf{x})$ of frame F^B, see Fig. 4.6. $\mathbf{R}^B(\Delta\mathbf{x})$ is defined in the body frame and represents the transformation from $F^{TB}(\mathbf{x}+\Delta\mathbf{x})$ to $F^{TB}(\mathbf{x})$. Let us also dennote with $\mathbf{R}(\Delta\mathbf{x})$ the instantaneous rotation in the global frame.

We interpret this local rotation $\mathbf{R}(\Delta\mathbf{x})$ as a rigid motion and not as a transformation between coordinate systems, see Sec. 2.1.1. That is, $\mathbf{R}(\mathbf{x})$ defines the motion of a point in the body from configuration $\mathbf{x}$ to configuration $\mathbf{x}+\Delta\mathbf{x}$, $\mathbf{p}_s(\mathbf{x}+\Delta\mathbf{x}) = \mathbf{R}(\Delta\mathbf{x})\mathbf{p}_s(\mathbf{x})$.

The rotation $\mathbf{R}^B(\mathbf{x})$ defined in the body frame is related to the rotation $\mathbf{R}(\mathbf{x})$ defined in the tracking frame by the *adjoint transformation* $Ad_{\mathbf{R}^{-1}(\mathbf{x})}$, 2.1.4.

$$\mathbf{R}^{B}(\Delta\mathbf{x}) = \mathbf{R}^{TB}(\mathbf{x})^{-1}\mathbf{R}(\Delta\mathbf{x})\mathbf{R}^{TB}(\mathbf{x}) \tag{4.18}$$

Thereby, the *tracking orientation* $\hat{\mathbf{R}}^{TS}$ is given by the longer path $F^{S} \overset{\mathbf{R}^{BS}}{\Longrightarrow} F^{B}(\mathbf{x}+\Delta\mathbf{x}) \overset{\mathbf{R}^{B}}{\Longrightarrow} F^{B}_{\mathbf{x}+\Delta\mathbf{x}} \overset{\mathbf{R}^{TB}(\mathbf{x})}{\Longrightarrow} F^{T}$, see Figure 4.6. Now we can compare this transformation matrix to the *ground-truth orientation* given by the sensors $\mathbf{R}^{TS}$

$$\mathbf{R}^{TB}(\mathbf{x})\mathbf{R}^{B}(\Delta\mathbf{x})\mathbf{R}^{BS} = \mathbf{R}^{TS}(t)\,. \tag{4.19}$$

Substituting $\mathbf{R}^{B}(\mathbf{x})$ by its expression in (4.18) it simplifies to

$$\mathbf{R}(\Delta\mathbf{x})\mathbf{R}^{TB}(\mathbf{x})\mathbf{R}^{BS} = \mathbf{R}^{TS}(t)\,. \tag{4.20}$$

Therefore, for each sensor s, we can minimize the squared Frobenius norm of both matrices with respect to $\mathbf{x}$

$$\arg\min_{\Delta\mathbf{x}} \sum_{s=1}^{5} \left\| \mathbf{R}_s(\Delta\mathbf{x})\mathbf{R}_s^{TB}(\mathbf{x})\mathbf{R}_s^{BS} - \mathbf{R}_s^{TS}(\mathbf{x}) \right\|^2 . \tag{4.21}$$

Equation (4.21) can again be linearized using and integrated into the linear system as soft constrains, see Figure 4.5. Nonetheless, it is interesting to take a closer look at equation (4.20). Substituting the rotational displacement $\mathbf{R}^{BS}$ in equation (4.20) by its expression in Eq. (4.9) we obtain

$$\mathbf{R}(\Delta\mathbf{x})\mathbf{R}^{TB}(\mathbf{x})\mathbf{R}^{TB}(0)^{-1}\mathbf{R}^{TS}(0) = \mathbf{R}^{TS}(t)\,. \tag{4.22}$$

Now let $\mathbf{R}(t)$ denote the instantaneous rotation from frame t to frame $t+1$. Hence, we can write $\mathbf{R}^{TB}(\mathbf{x})$ in terms of instantaneous rotations

$$\mathbf{R}(\Delta\mathbf{x})(\prod_{j=t-1}^{0} \mathbf{R}(j))\mathbf{R}(0)^{-1}\mathbf{R}^{TS}(0) = \mathbf{R}^{TS}(t)\,. \tag{4.23}$$

Simplifying $\mathbf{R}(0)^{-1}$ we obtain

$$\mathbf{R}(\Delta\mathbf{x})(\prod_{j=t-1}^{1} \mathbf{R}(j))\mathbf{R}^{TS}(0) = \mathbf{R}^{TS}(t)\,. \tag{4.24}$$

This last equation has a very nice interpretation because the columns of the matrix $(\prod_{j=t-1}^{1} \mathbf{R}(j))\mathbf{R}^{TS}(0)$ are simply the coordinates of the sensor axis in the first frame (columns of $\mathbf{R}^{TS}(0)$), rotated by the accumulated tracking motion from the first frame forward (i.e. not including the initialization motion in frame 0). This last result was very much expected and the interpretation is the following: if we have our rotation matrices defined in a reference frame F^{T}, we can just take the sensor axes in global coordinates in the first frame (columns of $\mathbf{R}^{TS}(0)$) and rotate them at every frame by the instantaneous rotational motions of the tracking. This will result in the estimated sensor axes in world coordinates, which is exactly the *tracking orientation* defined earlier in this Section. Therefore, the problem can be simplified to additional *3D-vector to 3D-vector* constraint equations which can be very conveniently integrated in our twist formulation. Being $\hat{\mathbf{r}}_1^{TS}(\mathbf{x})$,$\hat{\mathbf{r}}_2^{TS}(\mathbf{x})$,$\hat{\mathbf{r}}_3^{TS}(\mathbf{x})$ the *tracking orientation* basis axes at configuration $\mathbf{x}$, and $\boldsymbol{x}(t)$,$\boldsymbol{y}(t)$,$\boldsymbol{z}(t)$ *ground-truth orientation* basis axes in the current frame t, the constraint equations are

$$\mathbf{R}(\Delta\mathbf{x})\begin{bmatrix} \hat{\mathbf{r}}_1^{TS}(\mathbf{x}) & \hat{\mathbf{r}}_2^{TS}(\mathbf{x}) & \hat{\mathbf{r}}_3^{TS}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^{TS}(t) & \mathbf{r}_2^{TS}(t) & \mathbf{r}_3^{TS}(t) \end{bmatrix} \tag{4.25}$$

which can be linearized similarly as we did in the video-based tracker with image points to mesh points correspondences (*2D-point to 3D-point*). The difference now is that since we rotate vectors, only the rotational component of the twists is needed. Each additional sensor results in an additional 9 equations in the linear system

$$\left(\mathbf{I} + \Delta\omega' + \sum_{j \in \mathcal{J}(\mathbf{x})} \Delta\theta_j \widehat{\omega}'_j\right) \hat{\mathbf{r}}_1^{TS}(\mathbf{x}) = \mathbf{r}_1^{TS}(t) \tag{4.26}$$

$$\left(\mathbf{I} + \Delta\omega' + \sum_{j \in \mathcal{J}(\mathbf{x})} \Delta\theta_j \widehat{\omega}'_j\right) \hat{\mathbf{r}}_2^{TS}(\mathbf{x}) = \mathbf{r}_2^{TS}(t) \tag{4.27}$$

$$\left(\mathbf{I} + \Delta\omega' + \sum_{j \in \mathcal{J}(\mathbf{x})} \Delta\theta_j \widehat{\omega}'_j\right) \hat{\mathbf{r}}_3^{TS}(\mathbf{x}) = \mathbf{r}_3^{TS}(t) \tag{4.28}$$

which depends only on $\theta_j \widehat{\omega_j}$. In other words, the constraint equations do not depend at all on the joint axis location nor in the translational motion of the body. This implies that we can integrate the sensor information into the tracking system independently of the initial sensor orientation and location at the body limb. Note that Eq. (4.28) and Eq. (4.17) are equivalent.

4.5 Experiments

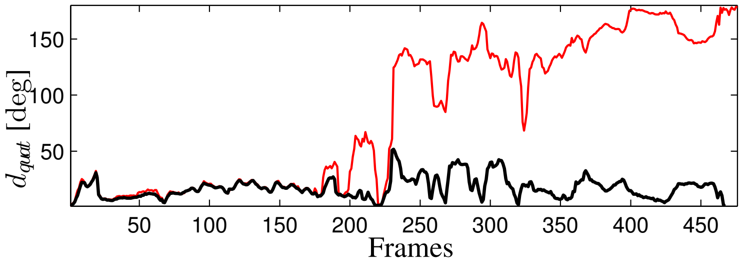


Figure 4.7: Error curves for video-based tracking (red) and hybrid tracking (black), referring to the orientations of the left lower leg for a hopping and jumping motion sequence.

In this section, we evaluate our multisensor-fusion approach for motion tracking by comparing the video-based tracker with our proposed hybrid tracker. Learning-based stabilization methods or joint angle limits can also be integrated into the video-based tracker. However, we did not include further constraints into the video-based tracker to demonstrate a general weakness of silhouette-based approaches. We note that the video-based tracker works well for many sequences, however in these experiments we focus on the occasions where it fails. Even though benchmarks for video-based tracking are publicly available [109], so far no data set comprising video as well as inertial data exist for free use. Therefore, for our experiments, we generated the *MPI08* data set consisting of 54 takes each having a length of roughly 15 seconds. We made the dataset publicly available for the scientific community [127]. In total, more than 10 minutes of tracking results were used for our validation study, which amounts to more than 24 thousand frames at a frame rate of 40 Hz. All takes have been recorded in a lab environment using eight calibrated video cameras and five inertial sensors fixed at the two lower legs, the two hands, and the neck. Our evaluation data set comprises various actions including standard motions such as walking, sitting down and standing up as well as fast and complex motions such as jumping, throwing, arm rotations, and cartwheels. For each of the involved four actors, we also generated a 3D mesh model using a laser scanner.

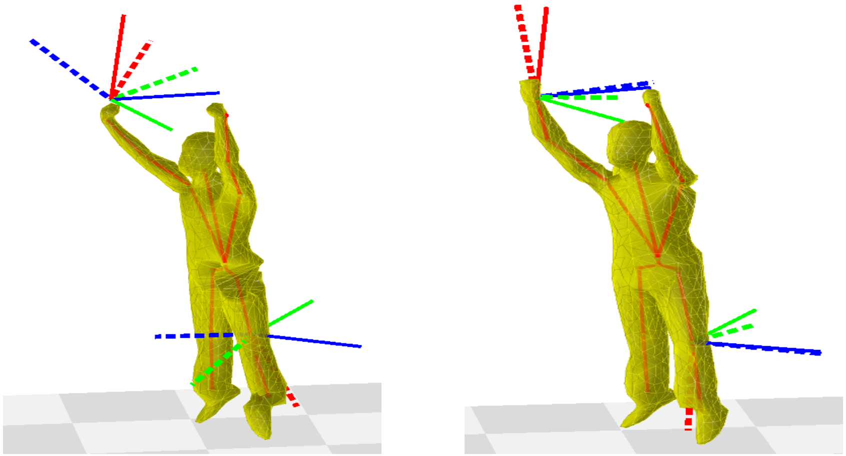

(a) (b)

Figure 4.8: Tracking result for video-based tracking (a) and hybrid tracking (b) for frame 450 of the motion sequence used in Figure 4.7. Ground-truth orientations in solid lines and tracking orientations in by dashed lines.

For a given tracking procedure, we introduce a framewise *error measure* by considering the angular distance between the two orientations $\mathbf{R}^{TS}$ and $\hat{\mathbf{R}}^{TS}$, see Sec. 4.4.2. To compute the error measure we first transform the rotation matrices $\mathbf{R}$ to its quaternion representation denoted as $\mathbf{q}$ This angular distance measured in degrees is defined by the formula, see Sec. 2.1.7

$$d_{\text{ang}}(\mathbf{q}^{TS},\hat{\mathbf{q}^{TS}}) = \frac{360}{\pi} \arccos \left| \left\langle \mathbf{q}^{TS}, \hat{\mathbf{q}}^{TS} \right\rangle \right| . \tag{4.29}$$

For a given motion sequence, we compute the error measure for each frame yielding an *error curve*.

In Figure 4.7, such error curves are shown for two different tracking procedures using the original video-based tracker (red) and the enhanced hybrid tracker (black). For the video-based tracking, there are large deviations between the ground-truth orientations and tracking orientations roughly starting with frame 200. Actually, as a manual inspection revealed, the actor performs in this moment a sudden turn resulting in a failure of the video-based tracking, where the left leg was erroneously twisted by almost 180 degrees. In contrast, the hybrid tracker could successfully track the entire sequence. This is also illustrated by Figure 4.8. Similarly, the figure also shows a tracking error in the right hand, which is corrected by the hybrid tracker as well. As a second example, we consider a very fast motion, where an actor first rotates his right and afterwards his left arm. Figure 4.9 shows the error curves for left and right hand for each of the tracking procedures. The curves reveal that the video-based tracker produced significant orientation errors in both hands. This shows that the hand orientations cannot be captured well considering only visual cues such as silhouettes. Again, the hybrid tracker considerably improved the tracking results, see also Figure 4.10. These examples demonstrate how the additional orientation priors resolve ambiguities from image cues. To estimate the quality of our hybrid tracker on more sequences, we computed the error measures (for lower legs, the two hands, and the neck) for each of the five sensors for all sequences and each actor of the data set. A total of 120210 error measures were computed separately for the hybrid and video tracker. We denote mean values and standard deviations of our error measure by μ_V, σ_V and μ_H, σ_H for the video-based and hybrid tracker, respectively. As summarized in Table 4.1, the sequences of each actor have been improved significantly, dropping the mean error from $30°$ to $13°$. This is also supported by the standard deviations. Let $\tau(s)$ denote the percentage of frames where at least one of the five sensors shows an error of more than s degrees. To show the percentage of corrected severe tracking errors, we computed $\tau_V(45)$ and $\tau_H(45)$ for every actor, see Tab. 4.1. As it turns out, most of the tracking errors are corrected, dropping the percentage of erroneously tracked frames from 19.29% to 2.51% of

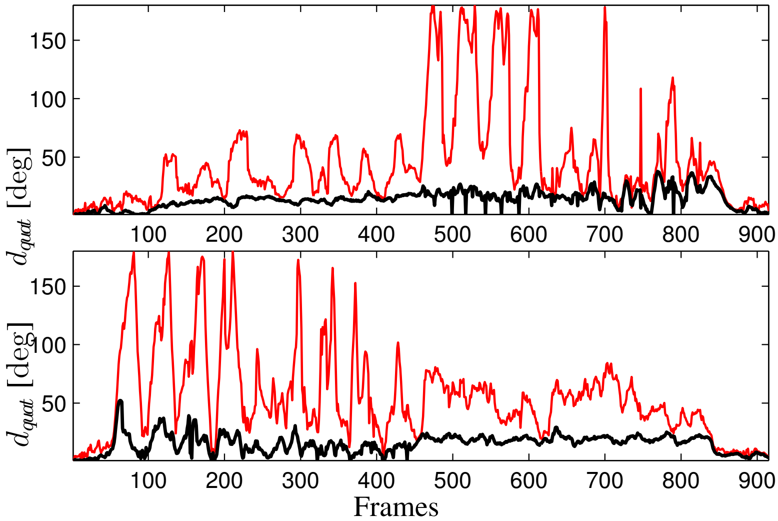


Figure 4.9: Error curves for video-based tracking (red) and hybrid tracking (black) obtained for an arm rotation sequence (first performed by the right and then by the left arm). **Top:** Left hand. **Bottom:** Right hand.

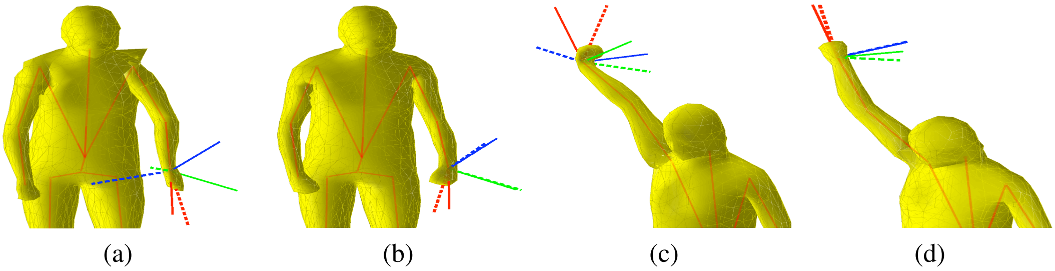


Figure 4.10: Tracking result and orientations for two selected frames of the sequence used in Figure 4.9. (a),(c) Video-based tracking. (b),(d) Hybrid tracking.

all frames. These findings are supported by the normalized histograms of the occurring values of the error measure, see Fig. 4.11. Furthermore, the hybrid tracker does not increase the computation time of the video-based tracker which is less than 4 s per frame.

One reason for the large amount of corrected errors is that the orientation of limbs is hard to estimate from silhouettes, since the cylindrical shape projects to the same silhouettes in many orientations. By combining the visual with orientation cues, these ambiguities are resolved, resulting in a largely improved performance with the hybrid tracker.

4.6 Summary

We presented an approach for stabilizing full-body markerless human motion capturing using a small number of additional inertial sensors. Generally, the goal of reconstructing a 3D pose from 2D video data suffers from inherent ambiguities. We showed that a hybrid approach combining information of multiple sensor types can resolve such ambiguities, significantly improving the tracking quality. In particular, our orientation-based approach could correct tracking errors arising from rotationally symmetric limbs. Using only a small number of inertial sensors fixed at outer extremities stabilized the tracking for the entire underlying kinematic chain.

In the next chapter, we introduce an alternative solution that enables tracking in outdoor settings, for fast motions, and in the presence of occlusions. To this end, we need suitable strategies that do not

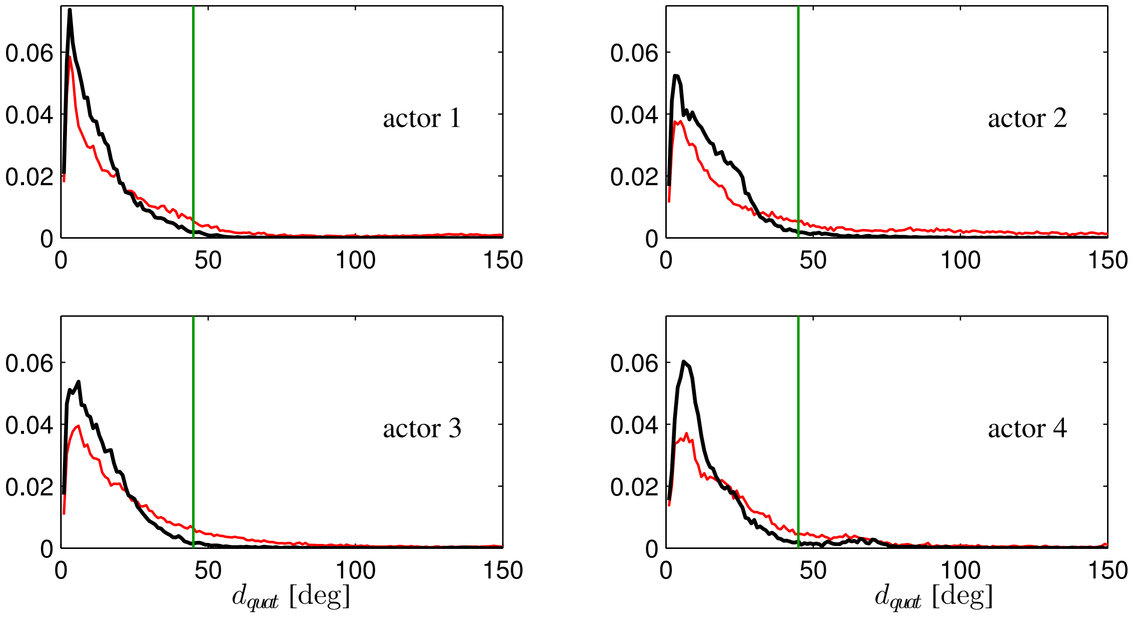


Figure 4.11: Normalized histogram, for each actor, of quaternion distances comparison for the whole database.

Table 4.1: Mean values μ and standard deviations σ for video-based (V) and hybrid (H) tracker for all sequences of the database, separated by actor. Percentage of large tracking errors denoted by $\tau(45)$.

	Actor 1	Actor 2	Actor 3	Actor 4	Average
μ_V [deg]	26.10	40.80	26.20	31.10	30.29
μ_H [deg]	11.50	14.86	13.98	13.85	13.47
σ_V [deg]	33.79	46.99	29.23	38.07	37.08
σ_H [deg]	9.89	13.01	12.25	14.43	12.28
$\tau_V(45)$ [%]	14.27	29.50	16.53	19.42	19.29
$\tau_H(45)$ [%]	0.47	3.33	2.12	6.45	2.51

destabilize the tracking process in the presence of sensor noise and local artifacts. The multimodal data set used is publicly available at [127] to further support this line of research.

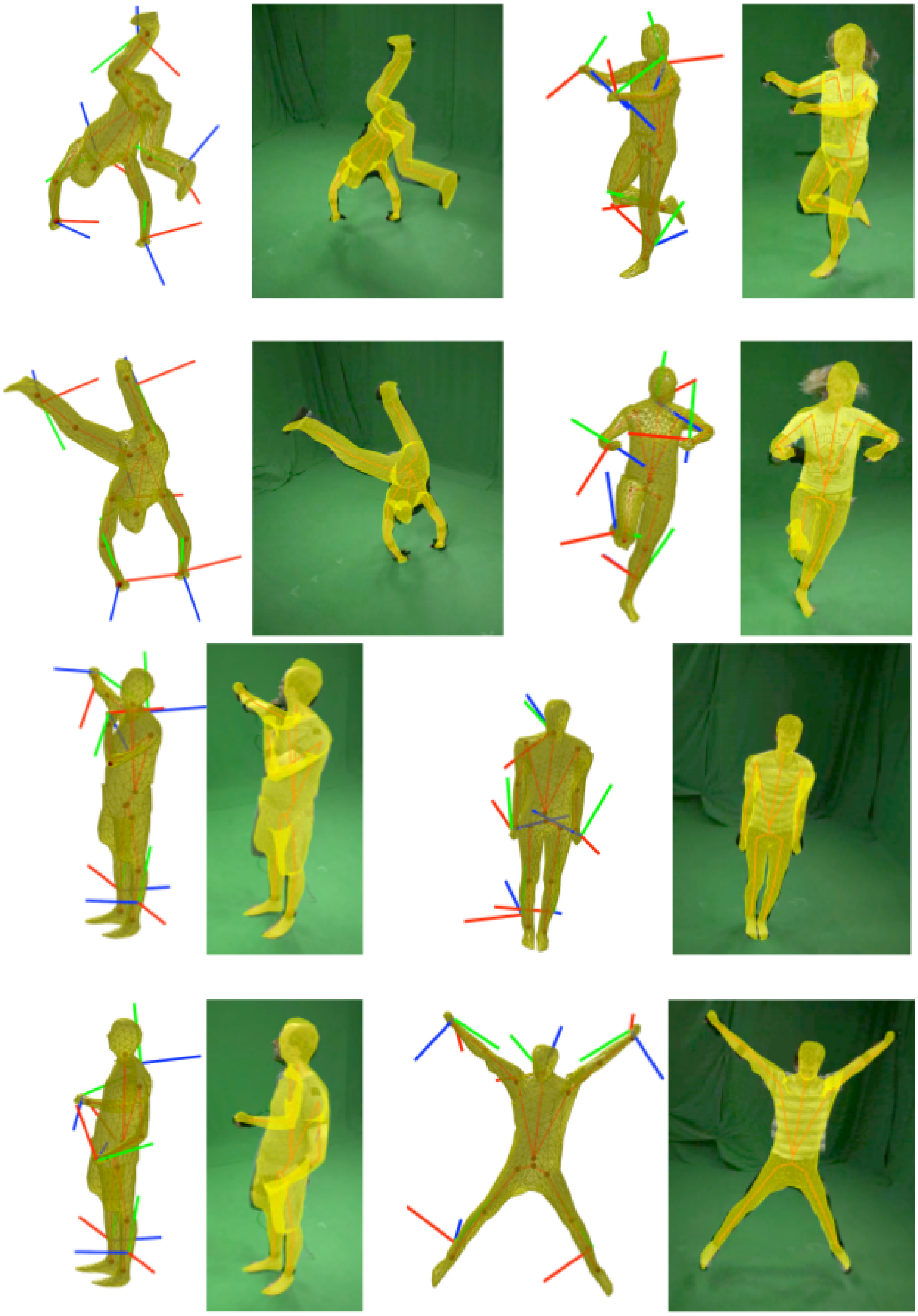
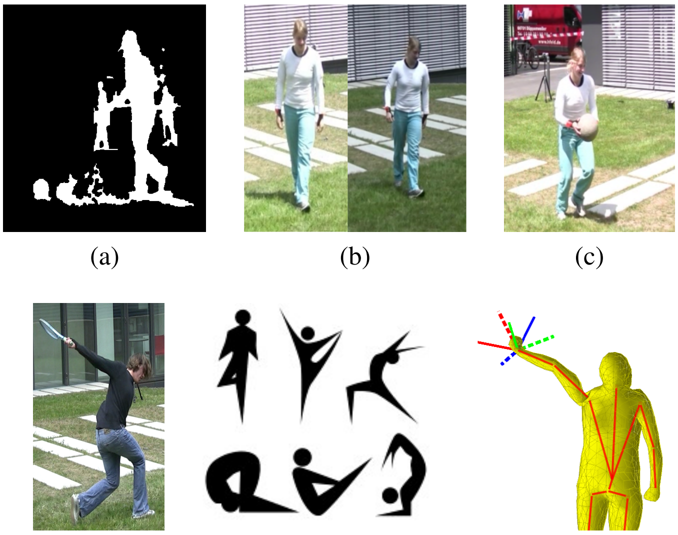


Figure 4.12: Examples of tracking results with our proposed hybrid tracker



(a) (b) (c)

(d) (e) (f)

Figure 5.1: Challenges outdoors: In outdoor scenarios the problem is significantly more difficult due to several factors (a) background clutter, (b) illumination changes, (c) self-occlusions, (d) fast motions and the inherent difficulties in human pose estimation, that is (e) high dimensional space and (f) orientation ambiguities.

5 Inverse Kinematics Sampler

The local hybrid tracker is efficient and performs well when small background and low noise is present in the image term. Unfortunately, in outdoor scenarios, there are a number of additional challenges: noisy silhouettes, illumination changes, calibration errors and miss-synchronization make the problem significantly more difficult. Any tracker based on local optimization will not successfully recover from local minima. In order to deal with uncertainties we rely here on a particle based optimization tracker that is capable of dealing with uncertainties. Specifically, we introduce a novel hybrid HMC system that combines video input with sparse inertial sensor input. Employing an annealing particle-based optimization scheme, our idea is to use orientation cues derived from the inertial input to sample particles from the manifold of valid poses. Then, visual cues derived from the video input are used to weight these particles and to iteratively derive the final pose. As our main contribution, we propose an efficient sampling procedure where the particles are derived analytically using inverse kinematics on the orientation cues. Additionally, we introduce a novel sensor noise model to account for uncertainties based on the von Mises-Fisher distribution. Doing so, orientation constraints are naturally fulfilled and the number of needed particles can be kept very small. More generally, our method can be used to sample poses that fulfill arbitrary orientation or positional kinematic constraints. In the experiments, we show that our system can track even highly dynamic motions in an outdoor environment with changing illumination, background clutter, and shadows.

5.1 Introduction

Recovering 3D human motion from 2D video footage is an active field of research [128, 9, 129, 5, 130, 16]. Although extensive work on human motion capturing (HMC) from multiview image sequences

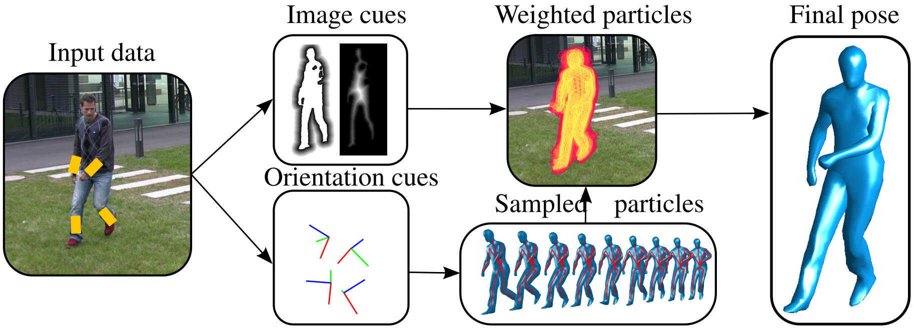


Figure 5.2: Orientation cues extracted from inertial sensors are used to efficiently sample valid poses using inverse kinematics. The generated samples are evaluated against image cues in a particle filter framework to yield the final pose.

has been pursued for decades, there are only few works, *e.g.* [118], that handle challenging motions in outdoor scenes.

To make tracking feasible in complex scenarios, motion priors are often learned to constrain the search space [14, 15, 28, 12, 16]. On the downside, such priors impose certain assumptions on the motions to be tracked, thus limiting the applicability of the tracker to general human motions. While approaches exist to account for transitions between different types of motion [19, 20, 22], general human motion is highly unpredictable and difficult to be modeled by pre-specified action classes.

Even under the use of strong priors, video HMC is limited by current technology: depth ambiguities, occlusions, changes in illumination, as well as shadows and background clutter are frequent in outdoor scenes and make state-of-the-art algorithms break down. Using many cameras does not resolve the main difficulty in outdoor scenes, namely extracting reliable image features. Strong lighting conditions also rule out the use of depth cameras. Inertial sensors (IMU) do not suffer from such limitations but they are intrusive by nature: at least 17 units must be attached to the body which poses a problem from bio-mechanical studies and sports sciences. Additionally, IMU's alone fail to measure accurately translational motion and suffer from drift. Therefore, similar to [25, 49, 37], we argue for a hybrid approach where visual cues are supplemented by orientation cues obtained by a small number of additional inertial sensors. While in [37] only arm motions are considered, the focus in [49] is on indoor motions in a studio environment where the cameras and sensors can be very accurately calibrated and the images are nearly noise- and clutter-free. By contrast, we consider full-body tracking in an outdoor setting where difficult lighting conditions, background clutter, and calibration issues pose additional challenges.

We introduce a novel hybrid tracker that combines video input from four consumer cameras with orientation data from five inertial sensors, see Fig. 5.2. Within a probabilistic optimization framework, we present several contributions that enable robust tracking in challenging outdoor scenarios:

- We show a strategy based on Inverse Kinematics (IK) to sample from the lower dimensional manifold of poses that satisfy the orientation constraints imposed by the sensors.
- We introduce a closed form solution for (IK) based on the Paden-Kahan subproblems
- We show how to integrate a sensor noise model based on the von Mises-Fisher distribution [131].

By sampling from a lower dimensional manifold we gain in efficiency and the kinematic constraints are naturally fulfilled. By including a sensor noise model we can deal with orientation ambiguities

In the experiments, Sec. 5.5, we demonstrate that our approach can track even highly dynamic motions in complex outdoor settings with changing illumination, background clutter, and shadows.

We can resolve typical tracking errors such as miss-estimated orientations of limbs and swapped legs that often occur in pure video-based trackers. Moreover, we compare it with three different alternative methods to integrate orientation data. The challenging dataset and sample code used is made available for scientific use [132]

5.2 Related Work

For solving the high-dimensional pose optimization problem, many approaches rely on local optimization techniques [76, 118, 8], where recovery from false local minima is a major issue. Under challenging conditions, global optimization techniques based on particle filters [129, 5, 133, 2] have proved to be more robust against ambiguities in the data. Thus, we build upon the particle-based annealing optimization scheme described in [5]. Here, one drawback is the computational complexity which constitutes a bottleneck when optimizing in high-dimensional pose spaces.

Several approaches show that constraining particles using external pose information sources can reduce ambiguities [134, 75, 114, 135, 136, 137, 23]. For example, [136] uses the known position of an object a human actor is interacting with and [134, 137] use hand detectors to constrain the pose hypotheses. To integrate such constraints into a particle-based framework, several solutions are possible. Firstly, the cost function that weights the particles can be augmented by additional terms that account for the constraints. Although robustness is added, no benefits in efficiency are achieved, since the dimensionality of the search space is not reduced. Secondly, rejection sampling, as used in [136], discards invalid particles that do not fulfill the constraints. Unfortunately, rejection sampling can be very inefficient and does not scale well with the number of constraints as we will show. Thirdly, approaches such as [138, 75, 24, 23] suggest to explicitly generate valid particles by solving an IK problem on detected body parts. While the proposals in [24, 23] are tailored to deal with depth ambiguities in monocular imagery, [75] relies on local optimization which is not suited for outdoor scenes as we will show. In the context of particle filters, the von Mises-Fisher distribution has been used as prior distribution for extracting white matter fiber pathways from MRI data [139].

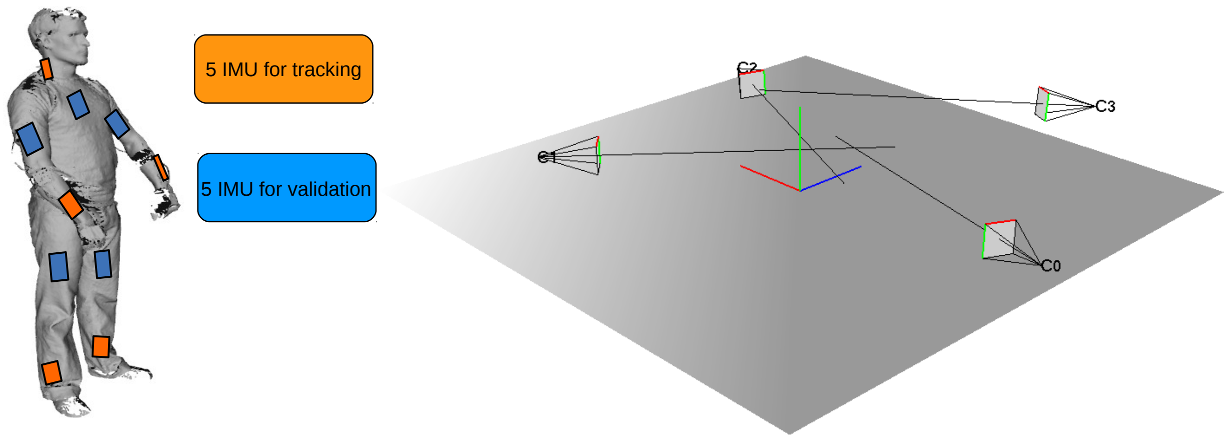
In contrast to previous work, our method can be used to sample particles that fulfill arbitrary kinematic constraints by reducing the dimension of the state space. Furthermore, none of the existing approaches perform a probabilistic optimization in a constrained low-dimensional manifold. We introduce an IK based on the *Paden-Kahan* sub-problems and model rotation noise with the von Mises-Fisher distribution.

5.3 Global Optimization with Sensors

To temporally align and calibrate the input data obtained from a set of uncalibrated and unsynchronized cameras and from a set of orientation sensors, we apply preprocessing steps as explained in Section 5.3.1. Then, we define orientation data within a human motion model (Section 5.3.2) and explain the probabilistic integration of image and orientation cues into a particle-based optimization framework (Section 5.3.3).

5.3.1 Calibration and Synchronization

We recorded several motion sequences of subjects wearing 10 inertial sensors (we used XSens [119]) which we split in two groups of 5: the *tracking sensors* which we use for tracking and the *validation sensors* which we use for evaluation. According to the specifications, the IMU orientation accuracy is around 2° for smooth motions and in absence of magnetic field. In practice, unfortunately, the error is much higher due to different sources of uncertainty, see Sect.5.4.3. The tracking sensors are placed in



(a) (b)

Figure 5.3: (a) IMU placement in the body, 5 for tracking and 5 for validation. (b) Camera-IMU calibration is achieved by placing a calibration cube with one of the axis aligned with the magnetic north.

the back and the lower limbs and the validation sensors are placed on the chest and the upper limbs. In the process of calibrating the camera, the global tracking coordinate system F^T is defined by a calibration cube placed into the recording volume see Sec. 3.2.1 for more details. In order to bring F^I and F^T into correspondence, we carefully place the calibration cube such that the axes of F^T directly correspond to the axes of the known F^I using a compass. Like this, the orientation data $\mathbf{R}_s^{IS}(t)$ also directly maps from the local sensor coordinate system F_s^S to the global tracking coordinate system F^T and we note $\mathbf{R}^{TS} := \mathbf{R}^{IS}$. Note that there might be slight missalignments between the tracking and inertial frame for which we compensate bt introducing a sensor noise model, see Sec. 5.4.3. In this Chapter, we refer to the sensor orientations by $\mathbf{R}^{TS}$ and, where appropriate, by using the corresponding quaternion representation [62], see Sec. 2.1.3. $\mathbf{q}^{TS}$.

The video sequences recorded with four off-the-shelf consumer cameras are synchronized by cross correlating the audio signals as proposed in [118]. Finally, we synchronize the IMU's with the cameras using a clapping motion, which can be detected in the audio data as well as in the acceleration data measured by IMU's.

5.3.2 Human Motion Model

Here again the motion of a human by a skeletal kinematic chain containing $N = 25$ joints that are connected by rigid bones. As before, the configuration of the body is then fully described by the vector of pose parameters $\mathbf{x} = (\xi_0, \Theta)$.

We now describe the relative rigid motion matrix $\mathbf{G}_i$ that expresses the relative transformation introduced by the rotation in the $i - th$ joint

$$\mathbf{G}_i = \exp(\theta_i \widehat{\xi}_i). \tag{5.1}$$

Let $\mathcal{J}_i \subseteq \{1, \ldots, n\}$ be the ordered set of parent joint indices of the $i-th$ bone. The total rigid motion $\mathbf{G}_i^{TB}$ of the bone is given by concatenating the global transformation matrix $\mathbf{G}_0 = \exp(\widehat{\xi}_0)$ and the relative rigid motions matrices $\mathbf{G}_i$ along the chain by

$$\mathbf{G}_i^{TB} = \mathbf{G}_0 \prod_{j \in \mathcal{J}_i} \exp(\theta_j \widehat{\xi}_j). \tag{5.2}$$

The rotation part of $\mathbf{G}_i^{TB}$ is referred to as *tracking bone orientation* of the $i-th$ bone. In the standard configuration of the kinematic chain, *i.e.*, the zero pose, we choose the local frames of each bone to be coincident with the global frame of reference F^T. Thus, $\mathbf{G}_i^{TB}$ also determines the orientation of the bone relative to F^T. A surface mesh of the actor is attached to the kinematic chain by assigning every vertex of the mesh to one of the bones. Let $\bar{\mathbf{p}}$ be the homogeneous coordinate of a mesh vertex $\mathbf{p}$ in the zero pose associated to the $i-th$ bone. For a configuration $\mathbf{x}$ of the kinematic chain, the vertex is transformed to $\bar{\mathbf{p}}'$ using $\bar{\mathbf{p}}' = \mathbf{G}_i^{TB}\bar{\mathbf{p}}$.

5.3.3 Optimization Procedure

If several cues are available, *e.g.*, image silhouettes and sensor orientation $\mathbf{z} = (\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sens}})$, the likelihood is commonly factored in two independent terms, see Sec. 2.5,

$$\arg\max_{\mathbf{x}} \; p(\mathbf{x}|\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sens}}) = p(\mathbf{z}^{\text{im}}|\mathbf{x})p(\mathbf{z}^{\text{sens}}|\mathbf{x})p(\mathbf{x}) \tag{5.3}$$

where it is assumed that the measurements $\mathbf{z}^{\text{im}}$ and $\mathbf{z}^{\text{sens}}$ are conditionally independent given that the pose $\mathbf{x}$ is known. The human pose $\mathbf{x}$ can then be found by minimizing the negative $\log$-likelihood which yields a weighted combination of cost functions for both terms as in [49]. Since in outdoor scenarios the sensors are not perfectly calibrated and the observations are noisy, fine tuning of the weighting parameters would be necessary to achieve good performance. Furthermore, the orientation information is not used to reduce the state space, and thus the optimization cost and ambiguities. Hence, we propose a different probabilistic formulation of the problem:

$$p(\mathbf{x}|\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sens}}) = \frac{p(\mathbf{z}^{\text{im}},\mathbf{z}^{\text{sens}}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}^{\text{im}},\mathbf{z}^{\text{sens}})} = \frac{p(\mathbf{z}^{\text{im}}|\mathbf{x})p(\mathbf{z}^{\text{sens}}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}^{\text{im}})p(\mathbf{z}^{\text{sens}})} \tag{5.4}$$

where we assumed independence between sensors and using

$$p(\mathbf{x}|\mathbf{z}^{\text{sens}}) = \frac{p(\mathbf{z}^{\text{sens}}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}^{\text{sens}})}$$

we obtain the following factorized posterior

$$p(\mathbf{x}|\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sens}}) \propto p(\mathbf{z}^{\text{im}}|\mathbf{x})p(\mathbf{x}|\mathbf{z}^{\text{sens}}). \tag{5.5}$$

that can be optimized globally and efficiently. We disregard the normalization factor $p(\mathbf{z}^{\text{im}})$ since it does not depend on the pose $\mathbf{x}$. The weighting function $p(\mathbf{z}^{\text{im}}|\mathbf{x})$ can be modeled by any image-based likelihood function. Our proposed model of $p(\mathbf{x}|\mathbf{z}^{\text{sens}})$, as introduced in Section 5.4, integrates uncertainties in the sensor data and constrains the poses to be evaluated to a lower dimensional manifold. For single frame pose estimation, optimization is typically performed by importance sampling [108], *i.e.*sampling from the prior $p(\mathbf{x})$ and weighting by the likelihood function $p(\mathbf{z}^{\text{im}}|\mathbf{x})$. The problem with this is that the prior is broad compared to $p(\mathbf{z}^{\text{im}}|\mathbf{x})$ that is peaky and typically multivalued. By drawing proposals directly from $p(\mathbf{x}|\mathbf{z}^{\text{sens}})$ we are effectively reducing the number of wasted samples, *i.e.*we are concentrating samples on the likelihood region. For optimization, we use the method proposed in [5]; the implementation details are given in Section 5.4.4.

5.4 Manifold Sampling

Assuming that the orientation data $\mathbf{z}^{\text{sens}}$ of the N_s orientation sensors is accurate and that each sensor has 3 DoF that are not redundant [1], the D dimensional pose $\mathbf{x}$ can be reconstructed from a lower

---

[1] Since the sensors are placed in different body parts they are not redundant because they explain different DoF in the kinematic chain

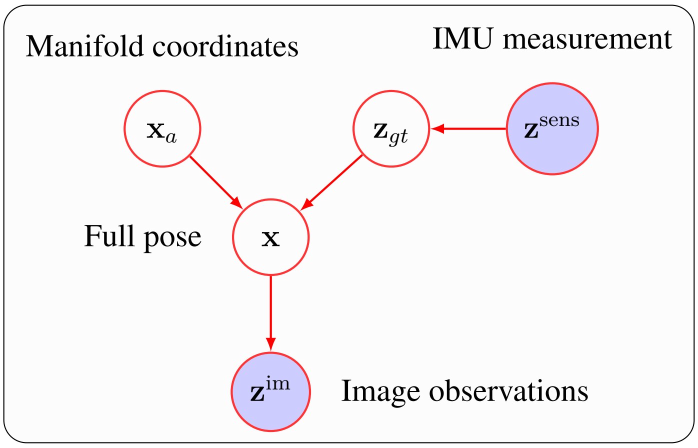


Figure 5.4: Graphical model of the approach. The measurements $\mathbf{z}^{\text{im}}$ and $\mathbf{z}^{\text{sens}}$ are shown as shaded nodes because they are observable during inference. The manifold coordinates, $\mathbf{x}_a$, the full state pose $\mathbf{x}$ and the true orientations $\mathbf{z}_{gt}$ are hidden. To infer the full state pose $\mathbf{x}$ we optimize the manifold coordinates and marginalize out $\mathbf{z}_{gt}$. To integrate out $\mathbf{z}_{gt}$, we assume it follows a von-Mises-Fisher distribution with mean direction $\boldsymbol{\mu} = \mathbf{z}^{\text{sens}}$.

dimensional vector $\mathbf{x}_a \in \mathbb{R}^d$ where $d = D - 3N_s$. In our experiments, a 31 DoF model can be represented by a 16 dimensional manifold using 5 inertial sensors as shown in Fig. 5.7 (a). The mapping is denoted by $\mathbf{x} = g^{-1}(\mathbf{x}_a,\mathbf{z}^{\text{sens}})$ and is described in Section 5.4.1. In this setting, Eq. (5.3) can be rewritten as

$$\arg\max_{\mathbf{x}_a} \; p\left(\mathbf{z}^{\text{im}}|g^{-1}(\mathbf{x}_a,\mathbf{z}^{\text{sens}})\right). \tag{5.6}$$

Since the orientation data $\mathbf{z}^{\text{sens}}$ is not always accurate due to sensor noise and calibration errors, we introduce a term $p(\mathbf{z}_{gt}^{\text{sens}}|\mathbf{z}^{\text{sens}})$ that models the sensor uncertainty, *i.e.*, the probability of the true orientation $\mathbf{z}_{gt}^{\text{sens}}$ given the sensor data $\mathbf{z}^{\text{sens}}$. We assume the conditional probability $p(\mathbf{z}_{gt}^{\text{sens}}|\mathbf{z}^{\text{sens}})$ follows a *von-Mises Fisher* distribution and it is described in detail Section 5.4.3. Hence, we get the final objective function:

$$\arg\max_{\mathbf{x}_a} \int p\left(\mathbf{z}^{\text{im}}|g^{-1}(\mathbf{x}_a,\mathbf{z}_{gt}^{\text{sens}})\right) p\left(\mathbf{z}_{gt}^{\text{sens}}|\mathbf{z}^{\text{sens}}\right) d\mathbf{z}_{gt}^{\text{sens}}. \tag{5.7}$$

where we marginalize out the sensor noise and optimize the manifold coordinates. The integral can be approximated by importance sampling, *i.e.*, drawing particles from $p(\mathbf{z}_{gt}^{\text{sens}}|\mathbf{z}^{\text{sens}})$ and weighting them by $p(\mathbf{z}^{\text{im}}|\mathbf{x})$. Consequently, we can efficiently concentrate the search space in the neighborhood region of a low dimensional manifold. In addition, we can guarantee that the kinematic constraints are satisfied.

5.4.1 Inverse Kinematics using Inertial Sensors

For solving Eq. (5.7), we derive an analytical solution for the map $g : \mathbb{R}^D \mapsto \mathbb{R}^{D-3N_s}$ and its inverse g^{-1}. Here, g projects $\mathbf{x} \in \mathbb{R}^D$ to a lower dimensional space and its inverse function g^{-1} uses the sensor orientations and the coordinates in the lower dimensional space $\mathbf{x}_a \in \mathbb{R}^{D-3N_s}$ to reconstruct the parameters of the full pose, *i.e.*,

$$g(\mathbf{x}) = \mathbf{x}_a \qquad g^{-1}(\mathbf{x}_a,\mathbf{z}^{\text{sens}}) = \mathbf{x}. \tag{5.8}$$

To derive a set of minimal coordinates, we observe that given the full set of parameters $\mathbf{x}$ and the kinematic constraints placed by the sensor orientations, a subset of these parameters can be written as a function $f(\cdot)$ of the others, see Fig. 5.5 for an intuitive illustration. Specifically, the full set of

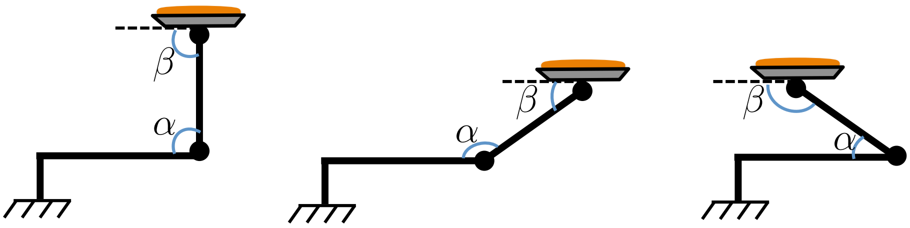


Figure 5.5: Toy example to illustrate our idea to sample from lower dimensional manifolds. For this simple kinematic chain the state vector has 2 DoF, $\mathbf{x} = (\alpha, \beta)$. If we impose the constraint that the cake plate must be perpendicular to the ground the true state vector has dimensionality 1. The constraint is $\alpha + \beta = \pi$ and therefore the state vector can be reparameterized as $\mathbf{x} = (\alpha, \pi - \alpha)$. For the problem of human pose estimation however the constraints are non-linear and therefore re-parametrization is achieved by solving small Inverse Kinematics subproblems.

parameters is decomposed into a set of *active parameters* $\mathbf{x}_a$ which we want to optimize according to Eq. (5.7) and a set of *passive parameters* $\mathbf{x}_p$ that can be derived from the constraint equations and the active set. Writing the state as $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_p)$ with $\mathbf{x}_a \in \mathbb{R}^d$ and $\mathbf{x}_p \in \mathbb{R}^{D-d}$ we have

$$f(\mathbf{x}_a, \mathbf{z}^{\text{sens}}) = \mathbf{x}_p \qquad \Longrightarrow \qquad g^{-1}(\mathbf{x}_a, \mathbf{z}^{\text{sens}}) = (\mathbf{x}_a, f(\mathbf{x}_a, \mathbf{z}^{\text{sens}})). \tag{5.9}$$

Thereby, the direct mapping g is trivial since from the full set only the active parameters are retained. The inverse mapping g^{-1} can be found by solving *inverse kinematics* (IK) sub-problems. Several choices for the decomposition into active and passive set are possible. To guarantee the existence of solution for all cases, we choose the passive parameters to be the set of 3 DoF joints that lie on the kinematic branches where a sensor is placed. In our experiments using 5 sensors, we choose the passive parameters to be the two shoulder joints, the two hips and the root joint adding up to a total of 15 parameters which corresponds to $3N_s$ constraint equations, see Fig. 5.7 (a). Hence, the passive parameters consist of N_s triplets of joint angles $\mathbf{x}_p = (\theta_{j_1}, \theta_{j_2}, \theta_{j_3})^T$, $j \in \{1 \dots N_s\}$ with corresponding rotation matrices $\mathbf{R}_j$. Since each sensor $s \in \{1 \dots N_s\}$ is rigidly attached to a bone, there exists a constant rotational offset $\mathbf{R}_s^{SB}$ between the i-th bone and the local coordinate system F_s^S of the sensor attached to it. This offset can be computed from the tracking bone orientation $\mathbf{R}_{i,0}^{TB}$ in the first frame and the sensor orientation $\mathbf{R}_{s,0}^{TS}$

$$\mathbf{R}_s^{SB} = (\mathbf{R}_{s,0}^{TS})^T \mathbf{R}_{i,0}^{TB}. \tag{5.10}$$

At each frame t, we obtain *sensor bone orientations* $\mathbf{R}_{s,t}^{TS} \mathbf{R}_s^{SB}$ by applying the rotational offset. In the absence of sensor noise, it is desired to enforce that the tracking bone orientation and the sensor bone orientation are equal:

$$\mathbf{R}_{i,t}^{TB} = \mathbf{R}_{s,t}^{TS} \mathbf{R}_s^{SB} \tag{5.11}$$

In Section 5.4.3 we show how to deal with noise in the measurements. Let $\mathbf{R}_j$ be the relative rotation of the j-th joint given by the rotational part of Eq. (5.1). The relative rotation $\mathbf{R}_j$ associated with the passive parameters can be isolated from Eq. (5.11). To this end, we expand the tracking bone orientation $\mathbf{R}_{i,t}^{TB}$ to the product of 3 relative rotations[2] $\mathbf{R}_j^p$, the total rotation motion of parent joints in the chain, $\mathbf{R}_j$, the unknown rotation of the joint associated with the passive parameters, and $\mathbf{R}_j^c$, the relative motion between the j-th joint and the i-th joint where the sensor is placed:

$$\mathbf{R}_j^p \mathbf{R}_j \mathbf{R}_j^c = \mathbf{R}_s^{TS} \mathbf{R}_s^{SB} \tag{5.12}$$

---

[2] The temporal index t is omitted for the sake of clarity

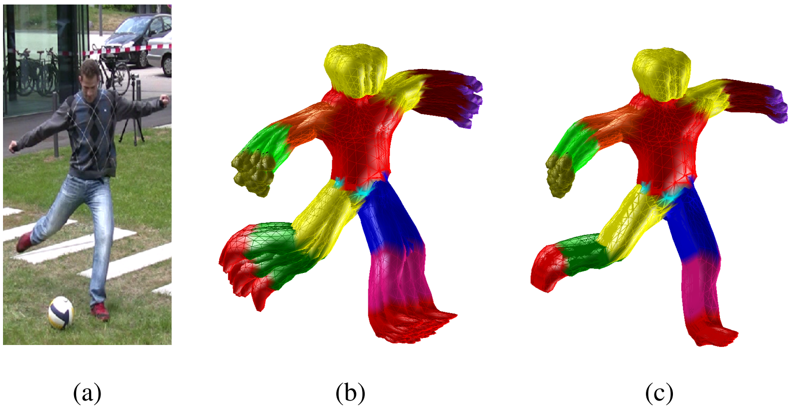

(a) (b) (c)

Figure 5.6: Manifold Sampling: **(a)** Original image. **(b)** Full space sampling. **(c)** Manifold sampling. Note that the generated samples in **(c)** have parallel end-effector orientations because they satisfy the constraints and uncertainty is therefore reduced.

Note that $\mathbf{R}_j^p$ and $\mathbf{R}_j^c$ are constructed from the active set of parameters $\mathbf{x}_a$ using the product of exponentials formula (5.2). From Eq. (5.12), we obtain the relative rotation matrix

$$\mathbf{R}_j = (\mathbf{R}_j^p)^T \mathbf{R}_s^{TS} \mathbf{R}_s^{SB} (\mathbf{R}_j^c)^T. \tag{5.13}$$

Having $\mathbf{R}_j$ and the known fixed rotation axes ω_{j_1},ω_{j_2},ω_{j_3} of the j-th joint, the rotation angles θ_{j_1},θ_{j_2},θ_{j_3}, *i.e.*, the passive parameters, must be determined such that

$$\exp(\theta_{j_1}\widehat{\omega}_{j_1})\exp(\theta_{j_2}\widehat{\omega}_{j_2})\exp(\theta_{j_3}\widehat{\omega}_{j_3}) = \mathbf{R}_j. \tag{5.14}$$

This problem can be solved by decomposing it into sub-problems [140], see Sec. 5.4.2. By solving these sub-problems for every sensor, we are able to reconstruct the full state $\mathbf{x}$ using only a subset of the parameters $\mathbf{x}_a$ and the sensor measurements $\mathbf{z}^{\text{sens}}$. In this way, the inverse mapping $g^{-1}(\mathbf{x}_a,\mathbf{z}_{\text{sens}}) = \mathbf{x}$ is fully defined and we can efficiently sample from the manifold, see Fig. 5.6.

5.4.2 Paden-Kahan subproblems

We are interested in solving the following problem:

$$\exp(\theta_1\widehat{\omega}_1)\exp(\theta_2\widehat{\omega}_2)\exp(\theta_3\widehat{\omega}_3) = \mathbf{R}_j. \tag{5.15}$$

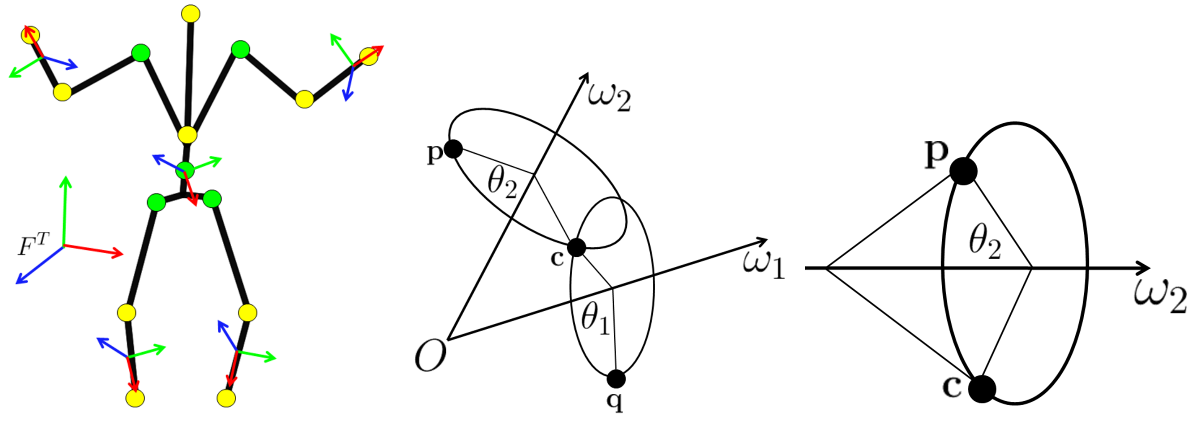
This problem can be solved by decomposing it into sub-problems as proposed in [140]. A comprehensive description of the Paden-Kahan subproblems applied to several inverse kinematic problems can also be found in [61]. The basic technique for simplification is to apply the kinematic equations to specific points. By using the property that the rotation of a point on the rotation axis is the point itself, we can pick a point $\mathbf{p}$ on the third axis ω_3 and apply it to both sides of Eq. (5.15) to obtain

$$\exp(\theta_1\widehat{\omega}_1)\exp(\theta_2\widehat{\omega}_2)\mathbf{p} = \mathbf{R}_j\mathbf{p} = \mathbf{q} \tag{5.16}$$

which is known as the *Paden-Kahan sub-problem 2*. For our problem the 3 rotation axes intersect at the same joint location. Consequently, since we are only interested in the orientations, we can translate the joint location to the origin $\mathbf{q}_j = O = (0,0,0)^T$. In this way, any point $\mathbf{p} = \lambda\omega_3$ with $\lambda \in \mathbb{R}$, $\lambda \neq 0$ is a valid choice for $\mathbf{p}$. Eq. (5.16) can decomposed in two subproblems

$$\exp(\theta_2\widehat{\omega}_2)\mathbf{p} = \mathbf{c} \quad \text{and} \quad \exp(-\theta_1\widehat{\omega}_1)\mathbf{q} = \mathbf{c}, \tag{5.17}$$

where $\mathbf{c}$ is the intersection point between the circles created by the rotating point $\mathbf{p}$ around axis ω_2 and the point $\mathbf{q}$ rotating around axis ω_1 as shown in Fig. 5.7 (b). In order for Eq. (5.17) to have a



(a) (b) (c)

Figure 5.7: Inverse Kinematics: **(a)** decomposition into active (yellow) and passive (green) parameters. Paden-Kahan sub-problem 2 **(b)** and sub-problem 1 **(c)**.

solution, the points $\mathbf{p}$, $\mathbf{c}$ must lie in the same plane perpendicular to ω_2, and $\mathbf{q}$, $\mathbf{c}$ must lie in the same plane perpendicular to ω_1. This implies that the projection of the position vectors[3] $\mathbf{p}$,$\mathbf{c}$,$\mathbf{q}$ onto the span of ω_1,ω_2 respectively must be equal, see Fig. 5.8

$$\omega_2^T \mathbf{p} = \omega_2^T \mathbf{c} \quad \text{and} \quad \omega_1^T \mathbf{q} = \omega_1^T \mathbf{c} \tag{5.18}$$

Additionally, the norm of a vector is preserved after rotation and therefore

$$\|\mathbf{p}\| = \|\mathbf{c}\| = \|\mathbf{q}\| \tag{5.19}$$

Since ω_1 and ω_2 are not parallel, the vectors $\omega_1, \omega_2, \omega_1 \times \omega_2$ form a basis that span $\mathbb{R}^3$. Hence, we can write $\mathbf{c}$ in the new basis as

$$\mathbf{c} = \alpha\omega_1 + \beta\omega_2 + \gamma(\omega_1 \times \omega_2) \tag{5.20}$$

where α, β,γ are the new coordinates of $\mathbf{c}$. Now, using the fact that $\omega_2 \perp \omega_1 \times \omega_2$ and $\omega_1 \perp \omega_1 \times \omega_2$, we can substitute Eq. (5.20) into Eq. (5.18) to obtain a system of two equations with two unknowns $(\alpha$,$\beta)$

$$\begin{aligned} \omega_2^T \mathbf{p} &= \alpha\omega_2^T\omega_1 + \beta \\ \omega_1^T \mathbf{q} &= \alpha + \beta\omega_1^T\omega_2 \end{aligned} \tag{5.21}$$

from which we can isolate the first two coordinates of $\mathbf{c}$

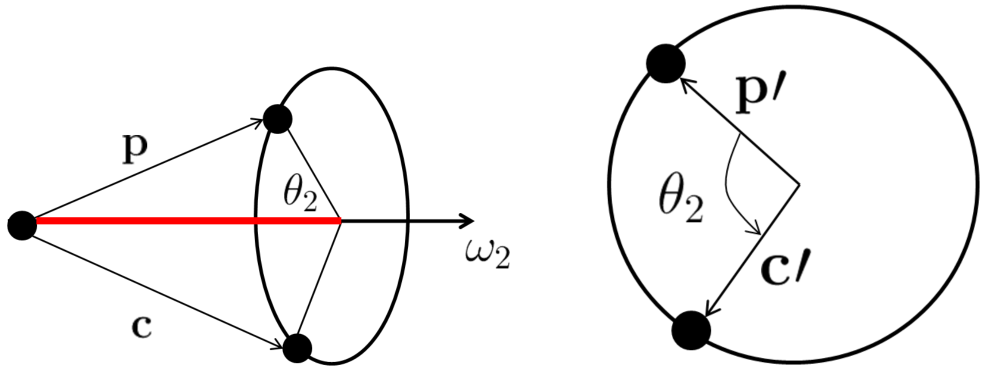
$$\begin{aligned} \alpha &= \frac{(\omega_1^T\omega_2)\omega_2^T\mathbf{p} - \omega_1^T\mathbf{q}}{(\omega_1^T\omega_2)^2 - 1} \\ \beta &= \frac{(\omega_1^T\omega_2)\omega_1^T\mathbf{q} - \omega_2^T\mathbf{p}}{(\omega_1^T\omega_2)^2 - 1}. \end{aligned} \tag{5.22}$$

From Eq. (5.19) and Eq. (5.20) we can write

$$\|\mathbf{p}\|^2 = \|\mathbf{c}\|^2 = \alpha^2 + \beta^2 + 2\alpha\beta\omega_1^T\omega_2 + \gamma^2\|\omega_1 \times \omega_2\|^2 \tag{5.23}$$

and obtain the third coordinate γ as

$$\gamma^2 = \frac{\|\mathbf{p}\|^2 - \alpha^2 - \beta^2 - 2\alpha\beta\omega_1^T\omega_2}{\|\omega_1 \times \omega_2\|^2} \tag{5.24}$$



(a) (b)

Figure 5.8: Paden-Kahan subproblem 1: **(a)** the projection length of $\mathbf{p}$ and $\mathbf{c}$ onto ω_2 must be equal, **(b)** the projection of the vectors $\mathbf{p}$ and $\mathbf{c}$ onto the orthogonal plane to the rotation axes ω_2

This last equation has no solution when the circles do not intersect, one solution when the circles are tangential and two solutions when the circles intersect at two points. For our choice of decomposition, the passive parameters correspond to $3DoF$ joints which are modeled as 3 concatenated revolute joints whose axis are mutually orthogonal. Therefore, there always exists a solution [61]. We note that the inverse kinematic solutions presented here are also valid for other decompositions, *e.g.*one could choose as passive parameters two rotation axes of the shoulder joint and one rotation axis of the elbow joints. However, the existence of solution should then be checked during the sampling process. Once we have the new coordinates (α, β, γ) we can obtain the intersection point $\mathbf{c}$ in the original coordinates using equation Eq. (5.20). Thereafter, Eq. (5.17) can be decomposed into two problems of the form

$$\begin{aligned} \exp(\theta_2 \widehat{\omega}_2)\mathbf{p} &= \mathbf{c} \\ \exp(-\theta_1 \widehat{\omega}_1)\mathbf{q} &= \mathbf{c} \end{aligned} \tag{5.25}$$

which simplifies to finding the rotation angle about a fixed axis that brings a point $\mathbf{p}$ to a second one $\mathbf{c}$, which is known as *Paden-Kahan sub-problem 1*

$$\exp(\theta_2 \widehat{\omega}_2)\mathbf{p} = \mathbf{c}. \tag{5.26}$$

This problem has a solution when the projections of the vectors $\mathbf{p}$ and $\mathbf{c}$ onto the orthogonal plane to ω_2 have equal lengths. Let $\mathbf{p}'$ and $\mathbf{c}'$ be the projections of $\mathbf{p}, \mathbf{c}$ onto the plane perpendicular to ω_2, see Fig. 5.8,

$$\mathbf{p}' = \mathbf{p} - \omega_2 \omega_2^T \mathbf{p} \quad \text{and} \quad \mathbf{c}' = \mathbf{c} - \omega_2 \omega_2^T \mathbf{c}. \tag{5.27}$$

If the projections have equal lengths $\|\mathbf{p}'\| = \|\mathbf{c}'\|$ then the problem is as simple as finding the angle between the two vectors

$$\begin{aligned} \omega_2^T(\mathbf{p}' \times \mathbf{c}') &= \sin\theta_2 \|\mathbf{p}'\| \|\mathbf{c}'\| \\ \mathbf{p}' \cdot \mathbf{c}' &= \cos\theta_2 \|\mathbf{p}'\| \|\mathbf{c}'\| \end{aligned} \tag{5.28}$$

By dividing the equations we finally obtain the rotation angle using the arc tangent

$$\theta_2 = \text{atan2}(\omega_2^T(\mathbf{p}' \times \mathbf{c}'), \mathbf{p}' \cdot \mathbf{c}'). \tag{5.29}$$

---

[3] Since we translated the joint location to the origin we can consider the points as vectors with origin at the joint location $\mathbf{q}_j$.

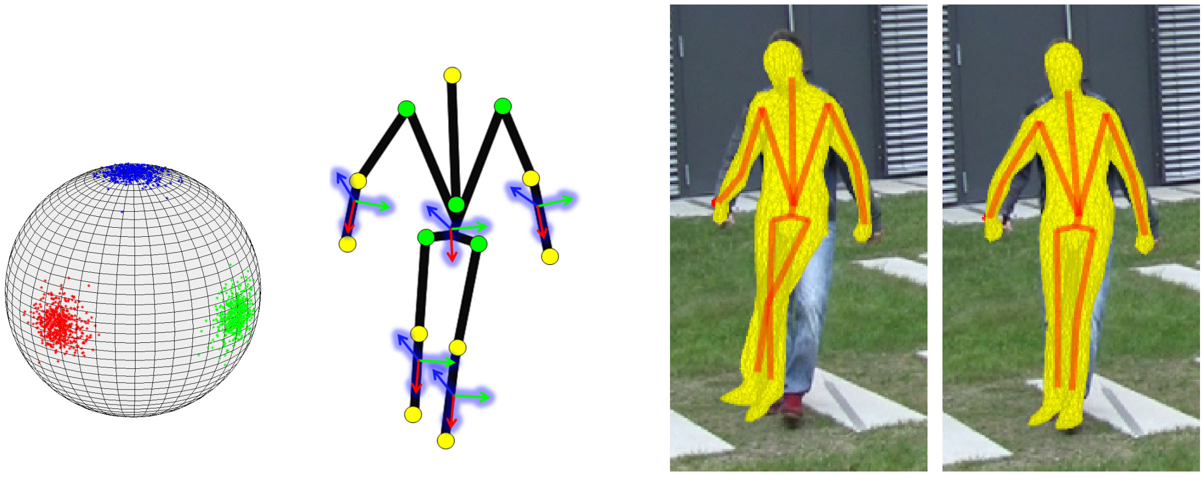

(a) (b) (c) (d)

Figure 5.9: Sensor noise model. **(a)** Points disturbed with rotations sampled from a von Mises-Fisher distribution. **(b)** The orientation of the particles can deviate from the sensor measurements. Tracking without **(c)** and with **(d)** sensor noise model.

We can find θ_1 using the same procedure. Finally, θ_3 is obtained from Eq. (5.15) after substituting θ_1 and θ_2

$$\exp(\theta_3\widehat{\omega}_3) = \exp(\theta_1\widehat{\omega}_1)^T \exp(\theta_2\widehat{\omega}_2)^T \mathbf{R}_j = \mathbf{R} \tag{5.30}$$

where the rotation matrix $\mathbf{R}$ is known. The rotation angle θ_3 satisfies

$$2\cos\theta_3 = (\text{trace}(\mathbf{R}) - 1) \tag{5.31}$$

$$2\sin\theta_3 = \omega_3^T \mathbf{r} \tag{5.32}$$

where $\mathbf{r} = (\mathbf{R}_{32} - \mathbf{R}_{23}, \mathbf{R}_{13} - \mathbf{R}_{31}, \mathbf{R}_{21} - \mathbf{R}_{12})$ (page 584 of [141]). Finally, the rotation angle θ_3 can be computed from $\cos\theta_3$ and $\sin\theta_3$ using atan2. By solving these sub-problems for every sensor, we are able to reconstruct the full state $\mathbf{x}$ using only a subset of the parameters $\mathbf{x}_a$ and the sensor measurements $\mathbf{z}^{\text{sens}}$. The good property of this geometric algorithms for solving inverse kinematics is that they are numerically very stable. More importantly, the same principle can be applied to solve more complex IK problems involving a number of positional and orientational constraints.

5.4.3 Sensor Noise Model

In practice, perfect alignment and synchronization of inertial and video data is not possible. In fact, there are at least four sources of uncertainty in the inertial sensor measurements, namely inherent sensor noise from the device, temporal unsynchronization with the images, small alignment errors between the tracking coordinate frame F^T and the inertial frame F^I, and errors in the estimation of $\mathbf{R}_s^{SB}$. Hence, we introduce a noise model $p(\mathbf{z}_{gt}^{\text{sens}}|\mathbf{z}^{\text{sens}})$ in our objective function (5.7). Rotation errors are typically modeled by assuming that the measured rotations are distributed according to a Gaussian in the tangent spaces which is implemented by adding Gaussian noise v^i on the parameter components, *i.e.*, $\tilde{\mathbf{x}}_j = \mathbf{x}_j + v^i$. The topological structure of the elements, a 3-sphere S^3 in case of quaternions, is therefore ignored. The *von Mises-Fisher* (MF) distribution models errors of elements that lie on a unit sphere S^{p-1} [131] and is defined as

$$f_p(\mathbf{x};\boldsymbol{\mu},\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{d/2-1}(\kappa)} \exp(\kappa\boldsymbol{\mu}^T\mathbf{x}) \tag{5.33}$$

where I_v denotes the modified Bessel function of the first kind, $\boldsymbol{\mu}$ is the mean direction, and κ is a concentration parameter that determines the dispersion form the true position. The distribution is illustrated in Fig. 5.9. For our problem, $p = 4$ and thus the elements $\mathbf{x}$ are quaternions. Therefore, on the one hand samples of the MF distribution are quaternions whose corresponding axis of rotation are uniformly distributed in all directions. On the other hand, the sample concentration decays with the angle of rotation. To see this, observe that the distribution can be expressed as a function of the angular rotation θ from the mean $\boldsymbol{\mu}$ where we replaced the inner product $\boldsymbol{\mu}^T\mathbf{x}$ by $\cos\left(\frac{\theta}{2}\right)$ (the inner prodcut between two quaternions results in $\cos(\frac{\theta}{2})$, where θ is the geodesic angle distance between rotations).

In order to approximate the integral in Eq. (5.7) by importance sampling, we use the method proposed in [142] to draw samples $\mathbf{q}_w$ from the von Mises-Fisher distribution with $p = 4$ and $\boldsymbol{\mu} = (1{,}0{,}0{,}0)^T$, which is the quaternion representation of the identity. We use a fixed dispersion parameter of $\kappa = 1000$. The sensor quaternions are then rotated by the random samples $\mathbf{q}_w$:

$$\tilde{\mathbf{q}}_s^{TS} = \mathbf{q}_s^{TS} \circ \mathbf{q}_w \tag{5.34}$$

where $\circ$ denotes quaternion multiplication. In this way, for every particle, samples $\tilde{\mathbf{q}}_s^{TS}$ are drawn from $p(\mathbf{z}_{gt}^{\text{sens}}|\mathbf{z}^{\text{sens}})$ using Eq. (5.34) obtaining a set of distributed measurements $\tilde{\mathbf{z}}^{\text{sens}} = \left(\tilde{\mathbf{q}}_1^{TS} \ldots \tilde{\mathbf{q}}_{N_s}^{TS}\right)$. This can be interpreted as the analogous of additive Gaussian Noise where $\mathbf{q}_w$ is a rotation noise sample. Thereafter, the full pose is reconstructed from the newly computed orientations with $g^{-1}(\mathbf{x}_a,\tilde{\mathbf{z}}^{\text{sens}})$ as explained in Section 5.4.1 and weighted by $p(\mathbf{z}^{\text{im}}|\mathbf{x})$.

In Fig. 5.10, we compare the inverse kinematic solutions of 500 samples $i \in \{1 \ldots 500\}$ by simply adding Gaussian noise *only* on the passive parameters $\{g^{-1}(\mathbf{x}_a,\mathbf{z}^{\text{sens}}) + \mathbf{v}^i\}_i$ and by modeling sensor noise with the von Mises-Fisher distribution $\{g^{-1}(\mathbf{x}_a,\tilde{\mathbf{z}}^{sens,i})\}_i$. For the generated samples, we fixed the vector of manifold coordinates $\mathbf{x}_a$ and we used equivalent dispersion parameters for both methods. To visualize the reconstructed poses we only show, for each sample, the elbow location represented as a point in the sphere. This example shows that simply adding Gaussian noise on the parameters is biased towards one direction that depends on the current pose $\mathbf{x}$. By contrast, the samples using von Mises-Fisher are uniformly distributed in all directions and the concentration decays with the angular error from the mean. Note, however, that Fig. 5.10 is a 3D visualization, in reality the bone orientations of the reconstructed poses should be visualized as points in a 3-sphere S^3.

$$f_p(\theta;\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{d/2-1}(\kappa)} \exp\left(\kappa \cos\left(\frac{\theta}{2}\right)\right) \tag{5.35}$$

5.4.4 Implementation Details

To optimize Eq. (5.7), we have implemented ISA (Interacted Simulated Annealing), the global optimization approach that has been proposed in [5] and use only the first stage of the algorithm, *i.e.*we do not locally optimize. ISA is based on simulated annealing which is a stochastic optimization technique to locate a good approximation of the global optimum of a cost function in a large search space. In the remainder of this chapter we will use the term global optimization whenever ISA was used for optimization to make the distinction with local optimization methods. As cost function, we use the silhouette and color terms

$$E(\mathbf{x}) = \lambda_1 E_{silh}(\mathbf{x}) + \lambda_2 E_{app}(\mathbf{x}) \tag{5.36}$$

with the setting $\lambda_1 = 2$ and $\lambda_2 = 40$. Although a good likelihood model is essential for good performance, it is not the focus of our work and we refer the interested reader to [2] for more details. During tracking, the initial particles $\{\mathbf{x}_a^i\}_i$ are predicted from the particles in the previous frame using a 3rd order autoregression and projected to the low-dimensional manifold using the mapping g; see

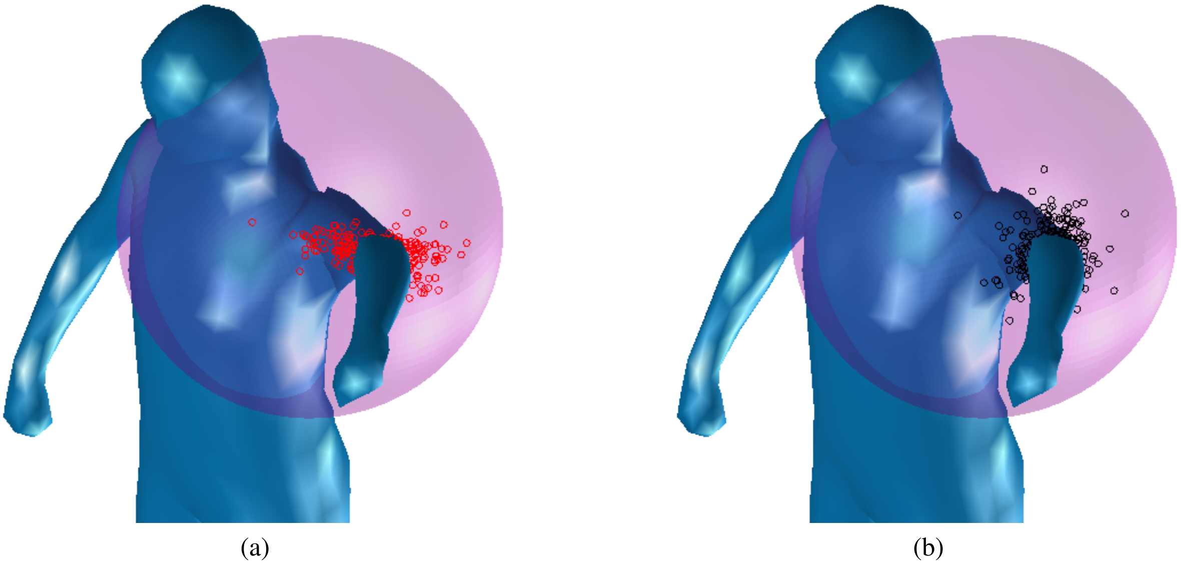

(a) (b)

Figure 5.10: Sensor noise model. 500 samples of the IK elbow location are shown as points using: **(a)** added Gaussian noise and **(b)** noise from the von Mises-Fisher distribution.

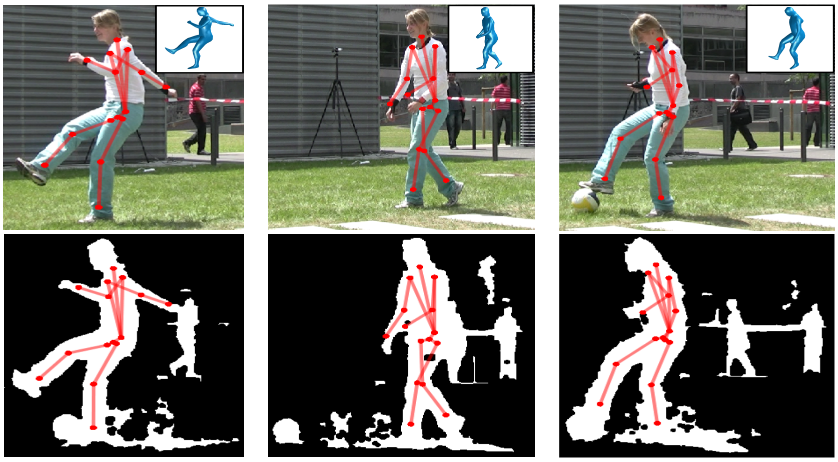


Figure 5.11: Tracking with background clutter.

Section 5.4.1. The optimization is performed only over the active parameters $\mathbf{x}_a \in \mathbb{R}^{D-3N_s}$, *i.e.*, the diffusion step is performed in $\mathbb{R}^{D-3N_s}$. Specifically, diffusion is performed with a Gaussian kernel with zero mean and covariance matrix

$$\Sigma_{a,k} = \frac{\alpha_\Sigma}{N-1}\left(\rho\mathbf{I} + \sum_i^N (\mathbf{x}_{a,k}^{(i)} - \mu_{a,k})(\mathbf{x}_{a,k}^{(i)} - \mu_{a,k})^T\right) \tag{5.37}$$

proportional to the sampling covariance matrix scaled by α_Σ where μ_k is the particle set mean at the current iteration k.

For the weighting step, we use the approach described in Section 5.4.3 to generate a sample $\tilde{\mathbf{z}}^{\mathrm{sens},i}$ from $p(\mathbf{z}_{gt}^{\mathrm{sens}}|\mathbf{z}^{\mathrm{sens}})$ for each particle $\mathbf{x}_a^i$. Consequently, we can map each particle back to the full space using $\mathbf{x}^i = g^{-1}(\mathbf{x}_a^i, \tilde{\mathbf{z}}^{\mathrm{sens,i}})$ and weight it by

$$\pi_k^{(i)} = \exp\left(-\beta_k \cdot E\left(g^{-1}(\mathbf{x}_{a,k}^i, \tilde{\mathbf{z}}^{\mathrm{sens}})\right)\right), \tag{5.38}$$

where β_k is the inverse temperature of the annealing scheme at iteration k and $E(\cdot)$ is the image cost function defined in Eq. (5.36). From the obtained set of weighted particles $\{\pi_k^{(i)}, \mathbf{x}_{a,k}^{(i)}\}_{i=1}^N$ we draw a new set of particles with resampling and probability equal to the normalized weights. The weighting, resampling and diffusion step are iterated M times before going to the next frame. In our experiments, we used 15 iterations for optimization. Finally, the pose estimate is obtained from the remaining particle set at the last iteration as

$$\hat{\mathbf{x}}_t = \sum_i \pi_k^{(i)} \, g^{-1}(\mathbf{x}_{a,k}^{(i)}, \tilde{\mathbf{z}}^{sens,i}). \tag{5.39}$$

The steps of our proposed sampling scheme are outlined in Algorithm 1.

Dynamics: To model the dynamics we use a 3rd order auto-regression using Gaussian Process regression that provides a prediction $\mathbf{x}^{pred}$ and a covariance matrix Σ^{pred} related with the confidence of the prediction. Thereby, the particles from the previous frame are drifted towards the predicted mean $\mathbf{x}^{pred}$ and diffused with a Gaussian kernel with zero mean and covariance Σ^{pred}. In order to obtain the low dimensional particle set, every particle is projected $g(\mathbf{x}_t^i) = \mathbf{x}_{a,t}^{(i)}$ [4]. We note that we do not learn a mapping directly in the low dimensional space since the previous estimates of passive parameters $\mathbf{x}_{p,t-4:t-1}$ are in general also correlated with the active parameters $\mathbf{x}_{a,t}$. The particle set is used as the initial proposal distribution for the first iteration of ISA.

Algorithm 2 Proposed algorithm

```
Require: number of layers $M$, number of samples $N$, initial distribution $\mathcal{L}_0$, sensor orientations
  $\mathbf{z}^{sens}$, image cost function $E(\cdot)$
  Initialize: Draw $N$ initial samples from $\mathcal{L}_0 \to \vec{x}_{a,k}^{(i)}$
  for layer $k = 0$ to $M$ do
    1. MANIFOLD SAMPLING
    start from the set of un-weighted particles of the previous layer
    for $i = 1$ to $N$ do
      1.1 SENSOR NOISE
      /* draw a sample $\tilde{\mathbf{z}}^{\text{sens},i}$ from $p(\mathbf{z}_{gt}^{\text{sens}} \mathbf{z}^{\text{sens}})$ */
      for $s = 1$ to $N_s$ do
        draw sample from von-Mises Fisher $f_p(\boldsymbol{\mu},\kappa) \to \mathbf{q}_w$
        $\tilde{\mathbf{q}}_s^{TS} = \mathbf{q}_s^{TS} \circ \mathbf{q}_w$
      end for
      set $\tilde{\mathbf{z}}^{\text{sens},i} = (\tilde{\mathbf{q}}_1^{TS} \dots \tilde{\mathbf{q}}_{N_s}^{TS})^T$
      1.1 INVERSE KINEMATICS
      /* computation of $\mathbf{x}_k^{(i)} = g^{-1}(\mathbf{x}_{a,k}^i, \tilde{\mathbf{z}}^{\text{sens}})$ */
      for $j = 1$ to $N_s$ do
        compute:  $\mathbf{R}_s^{TS} = \text{quat2mat}(\tilde{\mathbf{q}}_j^{TS})$
        compute:  $\mathcal{F}(\mathbf{x}_a) \to \mathbf{R}_j^p, \mathbf{R}_j^c$
        set:      $\mathbf{R}_j = (\mathbf{R}_j^p)^T \mathbf{R}_s^{TS} \mathbf{R}_s^{SB} (\mathbf{R}_j^c)^T$
        solve:    $\exp(\theta_{j_1}\widehat{\omega}_{j_1}) \exp(\theta_{j_2}\widehat{\omega}_{j_2}) \exp(\theta_{j_3}\widehat{\omega}_{j_3}) = \mathbf{R}_j$
      end for
      set: $\pi_k^{(i)} = \exp\left(-\beta_k \cdot E\left(\mathbf{x}_k^{(i)}\right)\right)$
    end for
    set: $\mathcal{L}_k = \{\pi_k^{(i)}, \mathbf{x}_{a,k}^{(i)}\}_{i=1}^N$
    2. RESAMPLING
    draw $N$ samples from $\mathcal{L}_k \to \vec{x}_{a,k}^{(i)}$
    3. DIFFUSION
    $\mathbf{x}_{a,k+1}^{(i)} = \mathbf{x}_{a,k}^{(i)} + \vec{B}_k$        {$\vec{B}_k$ is a sample from $\mathcal{N}(0, \Sigma_a)$}
  end for
```

5.5 Experiments

The standard benchmark for human motion capture is *HumanEva* that consists of indoor sequences. However, no outdoor benchmark data comprising video as well as inertial data exists for free use yet. Therefore, we recorded eight sequences of two subjects performing four different activities,

---

[4] Since the basic Gaussian process does not take the correlation of the output variables into account the process is equivalent to a 3rd order regression from previous full state estimates to the manifold coordinates

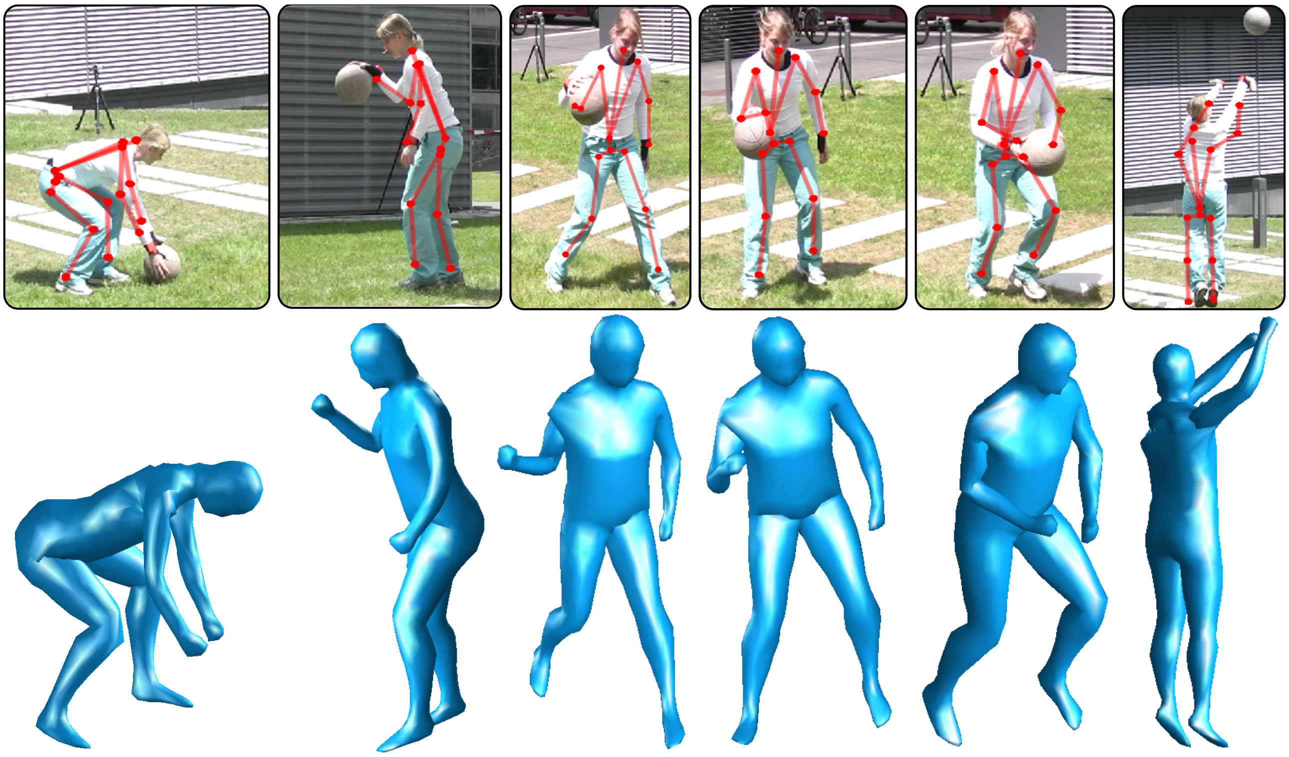


Figure 5.12: Tracking with strong illumination

namely walking, karate, basketball and soccer. Multiview image sequences are recorded using four unsynchronized off-the-shelf video cameras. To record orientation data, we used an Xsens Xbus Kit [119] with 10 sensors, see Sec. 5.3.1. Five of the sensors, placed at the lower limbs and the back, were used for tracking, and five of the sensors, placed at the upper limbs and at the chest, were used for validation. As for any comparison measurements taken from sensors or marker-based systems, the accuracy of the validation data is not perfect but is useful to evaluate the performance of a given approach. The eight sequences in the data set comprise over 3 minutes of footage sampled at $25\,\mathrm{Hz}$. Note that the sequences are significantly more difficult than the sequences of *HumanEva* since they include fast motions, illumination changes, shadows, reflections and background clutter. For the validation of the proposed method, we additionally implemented five baseline trackers: two video-based trackers based on local (L) and global optimization (G) respectively and three hybrid trackers that also integrate orientation data: local optimization (LS), global optimization (GS) and rejection sampling (RS) which we briefly describe here

- (L): Local optimization tracker. The model is projected to the image to find correspondences between the image silhouette contours and the model points. Then, the non-linear least squares problem is solved using a variant of *Levenberg-Marquardt algorithm*, see [118, 52] for more details.

- (G): Global Particle based optimization. Optimization here is performed by means of simulated annealing, *i.e.*, pose hypotheses are generated and weighted with progressively smooth versions of the image likelihood. The final pose is obtained as the average of the particle set in the last annealing layer, see [4, 5] for more details.

- (LS): Local optimization + inertial Sensors. Optimization is again performed by means of non-linear least squares but the cost function to be minimized consists of an image term and a term that models the likelihood of the inertial sensor measurements as explained in Chap. 4

$$E(\mathbf{x}) = \mu_1 E^{\text{im}}(\mathbf{x}) + \mu_2 E_1^{\text{sens}}(\mathbf{x})$$

where $E_1^{\text{sens}}(\mathbf{x})$ is defined as the squared Frobenious norm between the sensor and the tracking bone orientation matrices. Both the model-image Jacobian and the orientational Jacobian are derived analytically for better accuracy and efficiency. The algorithm is the based on [49].

- (GS): Global particle based optimization with Sensors. Like the (G) method but including the inertial sensor measurements in the weighting function. We optimize a cost function

  $$E(\mathbf{x}) = \mu_1 E^{\text{im}}(\mathbf{x}) + \mu_2 E_2^{\text{sens}}(\mathbf{x})$$

  where the image term $E^{\text{im}}(\mathbf{x})$ is the one defined in Eq. (5.36) and is chosen to be to be a piecewise increasing linear function of the angular error between the tracking and the sensor bone orientations. That is, for angular errors bigger than 10 degrees we scale the cost by a factor of 5. Big deviations from the orientation measurement could in principle be penalized with a quadratic function but this yields to many particles being rejected in early stages and results in lower performance. Note that although $\mu_2 E_2^{\text{sens}}(\mathbf{x})$ and $\mu_2 E_1^{\text{sens}}(\mathbf{x})$ are not identical they are both functions of distance metrics for rotations and are thus equivalent. For (LS) we optimize $\mu_2 E_1^{\text{sens}}(\mathbf{x})$ because derivatives are easier to compute. We hand tuned the influence weights μ_1,μ_2 to obtain the best possible performance.

- (RS): Rejection Sampling. This method is commonly used to sample hypotheses that satisfy a set of constraints. The method works by sampling hypotheses and rejecting hypotheses that do not satisfy the constraints up to a certain tolerance. It was for example used in [136] to integrate object interaction constraints. For our problem, to combine inertial data with video images we draw particles directly from $p(\mathbf{x}_t|\mathbf{z}^{\text{sens}})$ using a rejection sampling scheme. In our implementation of (RS), we reject a particle when the angular error for any of the constraints is bigger than 10 degrees.

For a comprehensive overview of model based methods for human pose estimation we refer the interested reader to [2].

Let the *validation set* be the set of quaternions representing the sensor bone orientations *not* used for tracking as $\mathbf{v}^{\text{sens}} = \{\mathbf{q}_1^{val}, \ldots, \mathbf{q}_5^{val}\}$. Let $i_s, s \in \{1 \ldots t\}$ be the corresponding bone index, and $\mathbf{q}_{i_s}^{TB}$ the quaternions of the tracking bone orientation (Section 5.3.2). We define the *error measure* as the average geodesic angle between the sensor bone orientation and the tracking orientation for a sequence of T frames as

$$d_{quat} = \frac{1}{5\,T} \sum_{s=1}^{5} \sum_{t=1}^{T} \frac{180^\circ}{\pi} 2 \arccos \left| \left\langle \mathbf{q}_s^{val}(t), \mathbf{q}_{i_s}^{TB}(t) \right\rangle \right| . \tag{5.40}$$

Comparison with video and local trackers: We compare the performance of four different tracking algorithms using the distance measure, namely (L), (G), (LS) and our proposed approach (P). We show d_{quat} for the eight sequences and each of the four trackers in Fig. 5.14. For (G) and (P) we used the same number of particles $N = 200$. As it is apparent from the results, local optimization is not suitable for outdoor scenes as it gets trapped in local minima almost immediately. Our experiments show that LS as proposed in [49] works well until there is a tracking failure in which case the tracker recovers only by chance. Even using (G), the results are unstable since the video-based cues are too ambiguous and the motions too fast to obtain reliable pose estimates. By contrast, our proposed tracker achieves an average error of $10.78^\circ \pm 8.5^\circ$ and clearly outperforms the pure video-based trackers and (LS).

Comparison with GS: In Fig. 5.15 (a), we show d_{quat} for a varying number of particles using the (GS) and our proposed algorithm (P) for a walking sequence.

The error values show that optimizing a combined cost function leads to bigger errors for the same number of particles when compared to our method. This was an expected result since we reduce

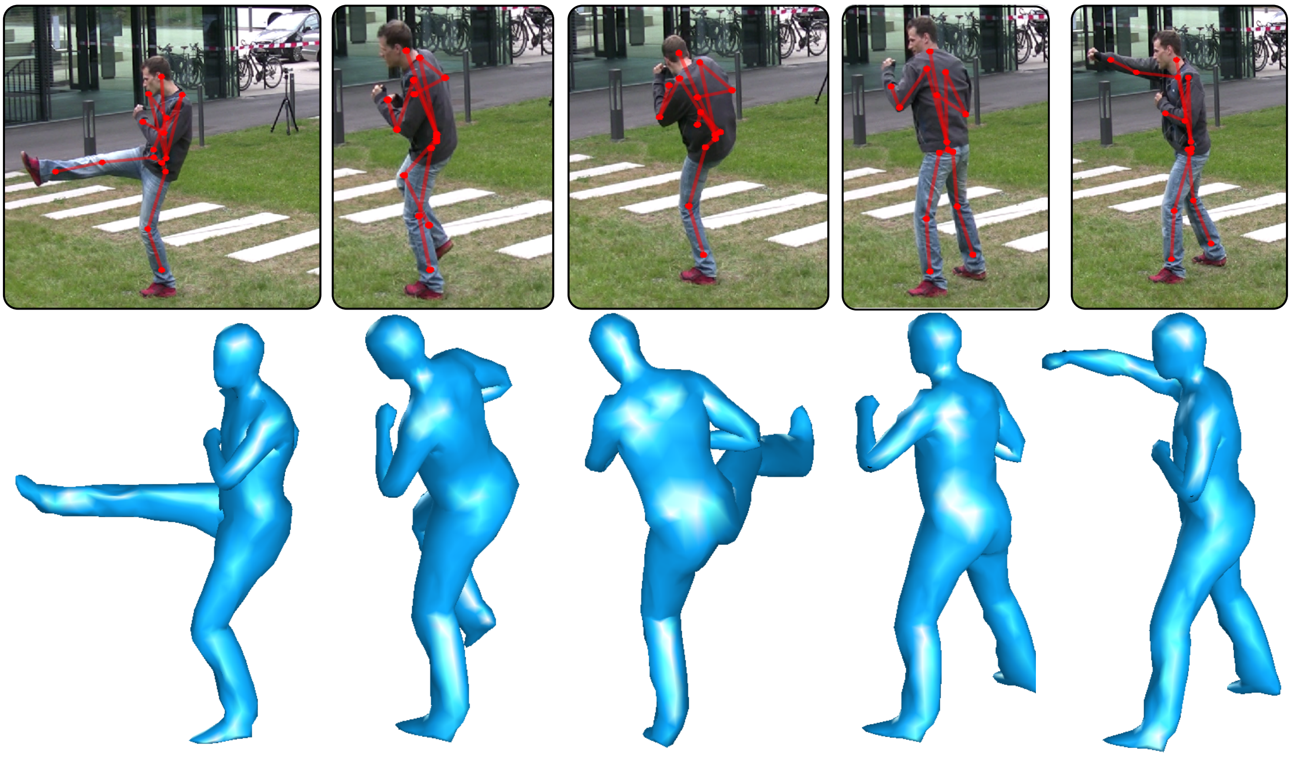


Figure 5.13: Tracking results of a karate sequence

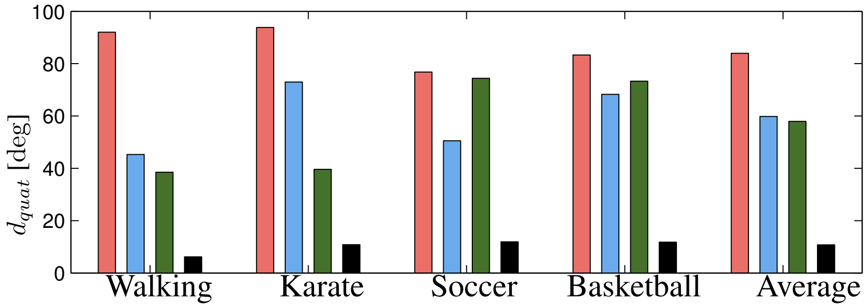
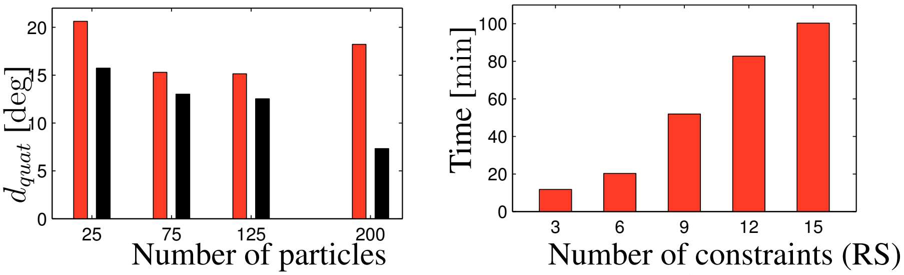


Figure 5.14: Mean orientation error of our 8 sequences (2 subjects) for methods (bars left to right) L (local optimization in red), LS (local+sensors in blue), GL (global optimization in green), and ours P (proposed in black).



(a) (b)

Figure 5.15: **(a)**: Orientation error with respect to number of particles with (red) the GS method and (black) our algorithm. **(b)**: Running time per frame of *rejection sampling* (RS) with respect to number of constraints. By contrast our proposed method takes 0.016 seconds for 15 DoF constraints. The time to evaluate the image likelihood is excluded as it is independent of the algorithm.

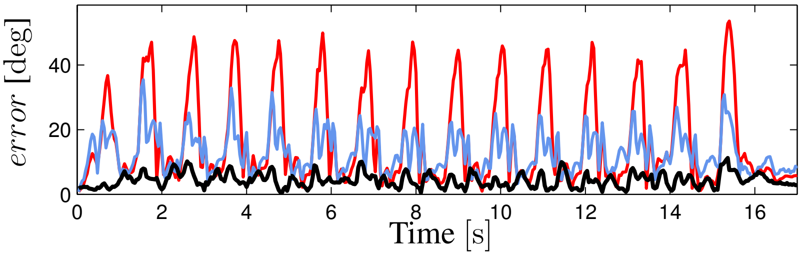


Figure 5.16: Angular error for the left hip of a walking motion with (red) no sensor noise model (NN), (blue) Gaussian noise model (GN) and (black) our proposed (MFN).

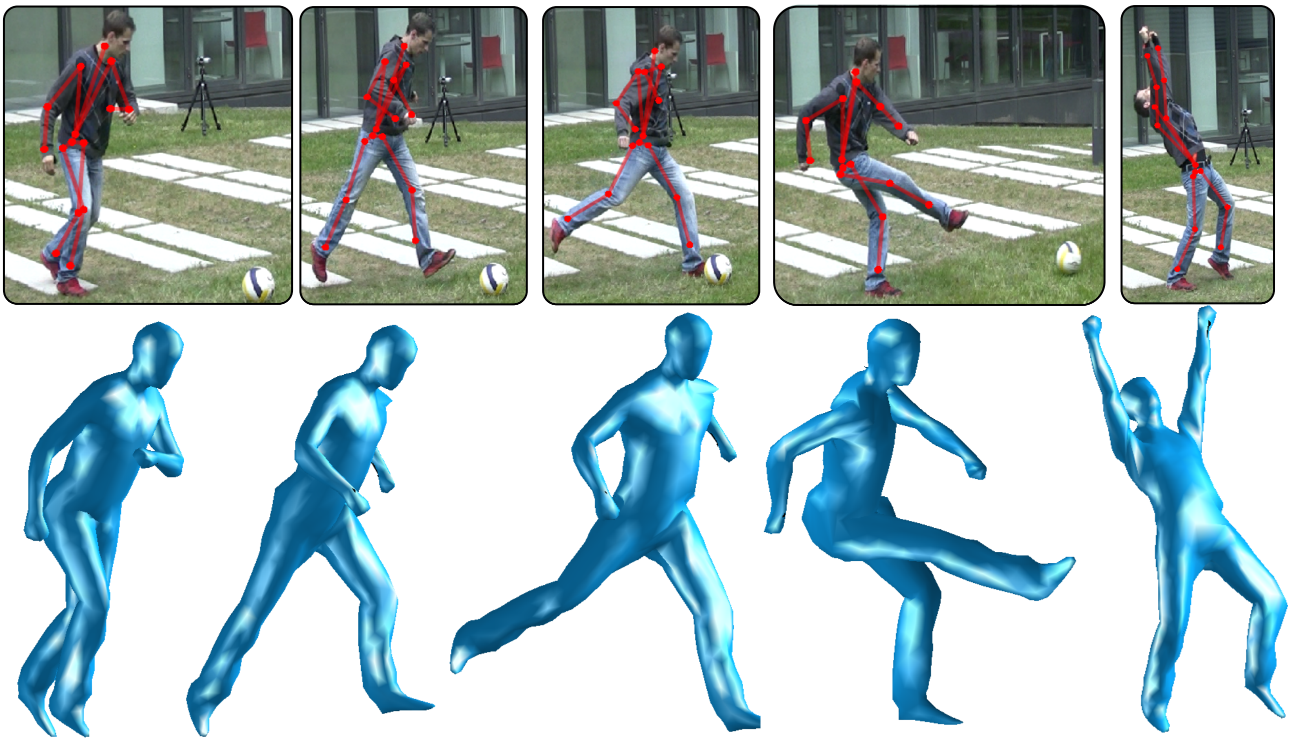


Figure 5.17: Tracking results of a soccer sequence

the dimension of the search space by sampling from the manifold and consequently less particles are needed for equal accuracy. Most importantly, the visual quality of the 3D animation deteriorates more rapidly with (GS) as the number of particles are reduced[5]. This is partly due to the fact that the constraints are not always satisfied when additional error terms guide the optimization.

---

[5] see the video for a comparison of the estimated motions at http://www.tnt.uni-hannover.de/~pons/

Comparison with Rejection Sampling (RS): Another option for combining inertial data with video images is to draw particles directly from $p(\mathbf{x}_t|\mathbf{z}^{\text{sens}})$ using a simple rejection sampling scheme. In our implementation of (RS), we reject a particle when the angular error is bigger than 10 degrees. Unfortunately, this approach can be very inefficient especially if the manifold of poses that fulfill the constraints lies in a narrow region of the parameter space. This is illustrated in Fig. 5.15 (b) where we show the processing time per frame (excluding image likelihood evaluation) using 200 particles as a function of the number of constraints. Unsurprisingly, rejection sampling does not scale well with the number of constraints taking as much as 100 minutes for 15 DoF constraints imposed by the 5 sensors. By contrast, our proposed sampling method takes in the worst case (using 5 sensors) 0.016 seconds per frame. These findings show that sampling directly from the manifold of valid poses is a much more efficient alternative.

Sensor Noise Model: To evaluate the influence of the sensor noise model, we tracked one of the walking sequences in our dataset using no noise (NN), additive Gaussian noise (GN) in the passive parameters and noise from the von Mises-Fisher (MFN) distribution as proposed in Section 5.4.3. In Fig. 5.16 we show the angular error of the left hip using each of the three methods. With (NN) error peaks occur when the left leg is matched with the right leg during walking, see Fig. 5.9. This typical example shows that slight misalignment (as little as $5^\circ - 10^\circ$) between video and sensor data can miss-guide the tracker if no noise model is used. The error measure was 26.8° with no noise model, 13° using Gaussian noise and 7.3° with the proposed model. The error is reduced by 43% with (MFN) compared to (GN) which indicates that the von Mises-Fisher is a more suited distribution to explore orientation spaces than the commonly used Gaussian. This last result might be of relevance not only to model sensor noise but to any particle-based HMC approach. Finally, pose estimation results for typical sequences of our dataset are shown in Fig. 5.11, 5.12, 5.13 and 5.17. A video of the proposed approach along with tracking results can be found in [132].

5.6 Discussion and limitations

State-of-the-art video trackers, either based on local or global optimization, suffer from 3D ambiguities inherent in video and usually fail to recover from errors. Our experiments reveal that video based pose estimation algorithms benefit from using a set of small IMUs, specially in outdoor scenarios where the image observation models are weak and ambiguous. Nonetheless, combining inertial and video measurements poses a difficult optimization problem that has to be dealt efficiently. Local optimization is fast and accurate in indoor scenarios. However, our findings indicate that to integrate orientation, (LS) is not suited in outdoor scenarios because it suffers from tracking failures that occur frequently. Optimizing a global cost function (GS) is also not the best choice since it yields an optimization in a high dimensional space which is computationally more expensive. In particular, a high number of hypotheses have to be generated since the search space volume is huge. Rejection sampling (RS) is not suited because it scales very poorly with the number of constraints and the computational time grows exponentially. Finally, we showed that the commonly used Gaussian Noise is outperformed by the proposed von Mises-Fisher noise model when it comes to modeling orientation ambiguities. The reason is that spherical sampling in the joint angle domain does not yield spatially spherical joint configurations as opposed to sampling using (MF). Our proposed method overcomes much of the described limitations: on the one hand the search space is explored only in the region that satisfies the constraints, and on the other hand sampling using Inverse Kinematics has a reinitialization power that overcomes tracking failures in many occasions. Unfortunately, the proposed method is limited by the availability of IMUs. Even though the IMUs are very small and we use only five, they are unavailable in several applications such as surveillance or MoCap and scene understanding from video archives. Another issue that requires improvement is robustness to unsynchronization produced by the IMUs lag during fast motions. The performance of our proposed tracker is still affected from such unsynchronization between IMUs and the video cameras. Since IMUs do not provide any positional measurement, our tracker fails when the body limbs (specially the arms) are not detectable due to long term occlusions. Finally, even though we achieve considerable computational gains w.r.t optimizing the full state space, evaluating the image cost function for every sample is still a bottle neck. To further reduce computational time, an option would be to use very few particles *e.g.*25 and then locally optimize to obtain better accuracy. Although in this work we have presented an algorithm to combine IMUs with video, the ideas shown here are of significant relevance for the computer vision community. Firstly, the Inverse Kinematics sampling scheme can be used to generate pose hypotheses that satisfy a set of kinematic constraints (we leave extensions to positional constraints as interesting future work). Secondly, the proposed sensor noise model can be used in any problem that involves modeling or optimization of rotation elements.

5.7 Summary

By combining video with IMU input, we introduced a novel particle-based hybrid tracker that enables robust 3D pose estimation of arbitrary human motions in outdoor scenarios. As the two main contributions, we first presented an analytic procedure based on Inverse Kinematics for efficiently sampling from the manifold of poses that fulfill orientation constraints. Notably, we show how the IK can be solved in closed form by solving smaller Paden-Kahan subproblems. Secondly, robustness to uncertainties in the orientation data was achieved by introducing a sensor noise model based on the von Mises-Fisher distribution instead of the commonly used Gaussian distribution. Our experiments on diverse complex outdoor video sequences reveal major improvements in the stability and time performance compared to other state-of-the art trackers. Although in this work we focused on the integration of constraints derived from IMU, the proposed sampling scheme can be used to integrate general kinematic constraints.

6 Posebits for Monocular Pose Estimation

We have seen in the previous sections that the combination of image data with inertial sensors results in significant accuracy gains. These approaches are very suitable for applications such as biomechanics or character animation where accuracy is the main requirement.
Unfortunately, a vast amount of visual data is already available on the Internet and this data typically consists of monocular videos. In order to analyze and understand this data, reliable human pose estimation algorithms that can operate on monocular images are necessary. However, monocular pose estimation is an extremely hard problem due to inherent depth ambiguities corresponding to bits of information that are not observable by typical generative trackers.
We tackle this by introducing *posebits*. Posebits are units of information that resolve typical ambiguities in monocular imagery. They are boolean geometric relationships between body parts designed to mimic human perception of poses (*e.g.* left-leg in front of right-leg or hands close to each other). We use classifiers trained discriminatively to infer posebits from image features. This has the advantage of being able to focus on the little details that can disambiguate the multi-modality of typical likelihood models. Furthermore, using posebits as a mid-layer representation for inference has several other advantages: Firstly, pose estimation becomes a much simpler task conditioned on these bits of information. Secondly, annotation simplifies to answering a small set of simple yes/no questions, and 3D MoCap data can be easily clustered in semantically similar classes. This allows for fast data collection at large in contrast to manual annotation of 3D poses from images. In Chap. 5 we have introduced a scheme to sample pose hypotheses consistent with orientation data. By contrast, here we introduce a scheme to sample pose hypotheses consistent with a set of inferred posebits. There exist several new potential applications of posebits, in this chapter we show how they can be used successfully to estimate pose from a single image and for semantic retrieval of related images.

The posebits pose estimator proposed here could be used to initialize the previously described hybrid trackers. Furthermore, since the the method proposed in this chapter works on single images, it may also be used to stabilize tracking and prevent drifting.

6.1 Introduction

While tremendous effort has focused on the extraction of metric 3D pose from images and video, in this work we consider the estimation of qualitative pose information, called posebits. Posebits are attributes about the pose that specify the relative positions between some parts in the body. They have the advantage that they can be inferred from images reliably, ground truth posebit labels are easy to collect to build significant training datasets, and posebits provide extremely useful information for resolving pose ambiguities in subsequent pose inference.
Generative methods [12, 9, 76, 4, 143, 5] model the complex image formation process which typically involves careful modeling of shape and motion models. While such methods are powerful when there is sufficient image evidence and can generalize better, they typically fail when the data is incomplete as in the monocular scenarios studied here. The main limitation is that the likelihood models used are weak and therefore are highly multi-modal. Discriminative methods are more direct and learn direct mappings from image features to 3D poses. This group of algorithms are powerful when the data term is weak and partially observable. However, they require well aligned training pairs of image features and 3D poses which are difficult to obtain in practice. One option to obtain such training

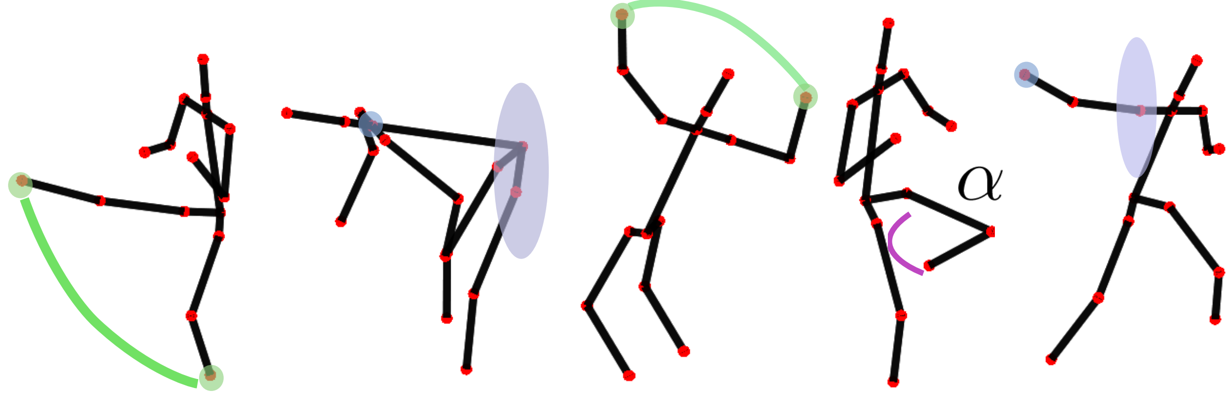


Figure 6.1: We propose to describe poses through semantic descriptions such as: *the left foot is far from the right foot or the right hand is on the right of the shoulder*. This has the advantage of encoding semantic relationships more closely related to how humans perceive poses.

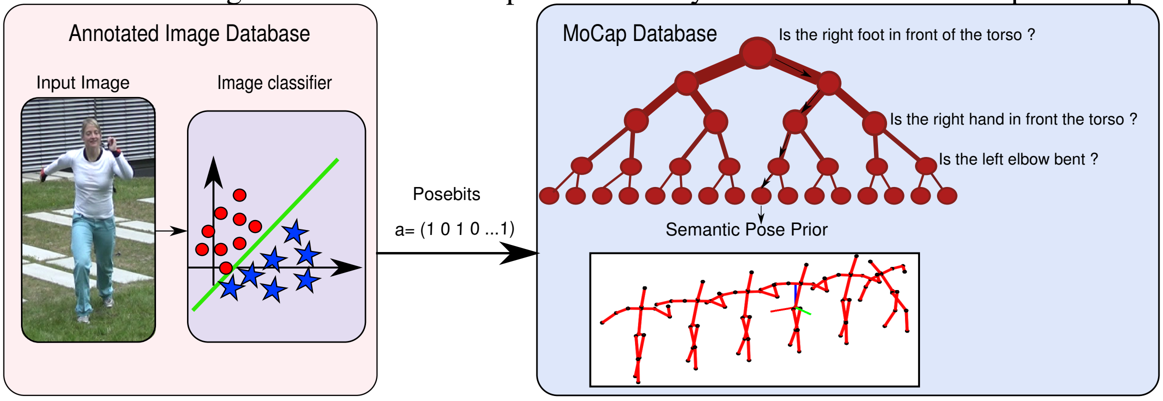


Figure 6.2: By representing poses using posebits we can be readily obtain annotations from humans, in contrast to skillful and laborious work that is required to annotate 3D positions. We have collected a database of annotated images using Amazon Mechanical Turk from which we can learn a classifier to infer posebits directly from image features. Additionally, posebits can be easily computed from MoCap data, which allows us to cluster poses in semantically meaningful classes. Given the inferred posebits we can draw proposals that are semantically related to the image content. Our work facilitates the complicated task of obtaining 3D pose information of humans in natural scenes.

sets is to use commercial marker-based MoCap systems(*e.g.*Vicon) synchronized with video cameras [109]. The problem here is that the images obtained are limited to indoor lab environments which are far from capturing the true variability seen in natural scenes [144]. A second option is to manually annotate the 3D pose from images but this is obviously time consuming and prone to errors [45, 145]. We argue that it would be extremely practical and powerful to learn useful 3D information directly from training pairs that can be easily annotated by humans. It appears that people perceive poses as relative positions between body parts, see Fig. 6.1 rather than absolute positions or joint angle representations commonly used for pose estimation. For example, common human verbalizations of pose are : *the left leg in front of right leg, left hand in front of the torso*. Therefore, we adopt such high-level representation of poses and we refer to as *posebits*. A posebit is an attribute about the pose that determines the relative positions between some parts in the body. It turns out that once some of this posebits of information are known, the difficult monocular pose estimation problem becomes much less ambiguous. The main contribution of our work is to introduce *posebits which constitute a semantically powerful pose descriptor* that can be inferred from the images. We select a subset of posebits out of a large pool of candidates. Inspired by works on feature selection and decision trees we propose a selection method that greedily selects a set of attributes by maximizing information gain. In contrast to traditional decision trees, we also account for uncertainty in the estimation of features. For inference we use structural SVMs [146] and learn a single discriminant function that exploits both the correlation among classes as well as co-occurrences of posebits. We show that conditioning on the inferred posebits we can draw pose proposals that are semantically with the image and we are able to infer poses from single images.

6.2 Related Work

To deal with monocular ambiguities and multimodality **generative** approaches have relied on strong action specific motion priors. Our work is partially related to these works [19, 21, 22] in that we also define an intermediate pose representation. However, posebits are more flexible than action classes because they are compositional and do not suffer from big intra-class variability. **Discriminative** methods model a mapping from image features to the pose space by using training examples. Approaches are either based on nearest neighbors (KNN) schemes [28, 29] or global parametric predictors [31]. Recent works [32, 33, 34] build online local models from a subset of training exemplars found using *e.g.*, KNN or decision trees [35]. Another way to correlate the input features with the output poses is through shared latent variables [21, 34]. In contrast to fully discriminative methods our proposed algorithm does not require training pairs of images and poses which enables easy annotation and data collection at large in natural scenes.

Attributes: Our work is also related, at least conceptually, to recent works using attribute representations. The advantages of using attributes have been demonstrated for object categorization [40, 41], and human action recognition [42, 43] with a special emphasis on transfer learning between classes. Similar attributes as the ones we propose here have been used for content based retrieval of motion capture data [44], where they noted that similar motions may be numerically not similar. In [19], they also used attributes for retrieving motion priors to stabilize tracking. They do it in an iterative scheme in which they start tracking with no prior knowledge and use noisy tracking estimates to retrieve action specific priors. None of these works infer attributes directly from images as we do. Our work is also inspired by [45] where they introduce poselets. A poselet is a new notion of part, they noted that parts need not to correspond to physical body parts as in [46, 47, 48]. Instead, they argue that detecting subgroups of body parts in a certain configuration is easier. However, whereas poselets have shown good performance for people detection, here we focus on estimating the pose from monocular images. Furthermore, we do not require 3D annotations for training. Posebits provide the bridge between the pose and the image space and at the same time allow to build conditional class models of semantically similar poses.

6.3 Posebits

The use of posebits for pose estimation comprises the following two problems that are tightly coupled: 1) How to define posebit candidates and select them, and how to infer them from images, and 2) given that we can extract posebits, how we can use them to help pose estimation.

Provided with a set of selected posebits, we infer them from image features using a structural SVM classifier, (Sec. 6.3.2). This allows one to build a proposal distribution of poses conditioned on the extracted posebits, (Sec. 6.3.3). The main advantage is that by sampling from this proposal distribution, many ambiguities in the 3D pose are removed. In this way, the final pose is obtained by scoring the proposals using an image based likelihood, (Sec. 6.4). Generation of posebit candidates and subsequent selection criteria depends on both the inference method (Sec. 6.3.2) used and on the pose proposal distribution (Sec. 6.3.3). Hence, we explain later in Sec. 6.3.4 how to automatically generate and select posebit candidates.

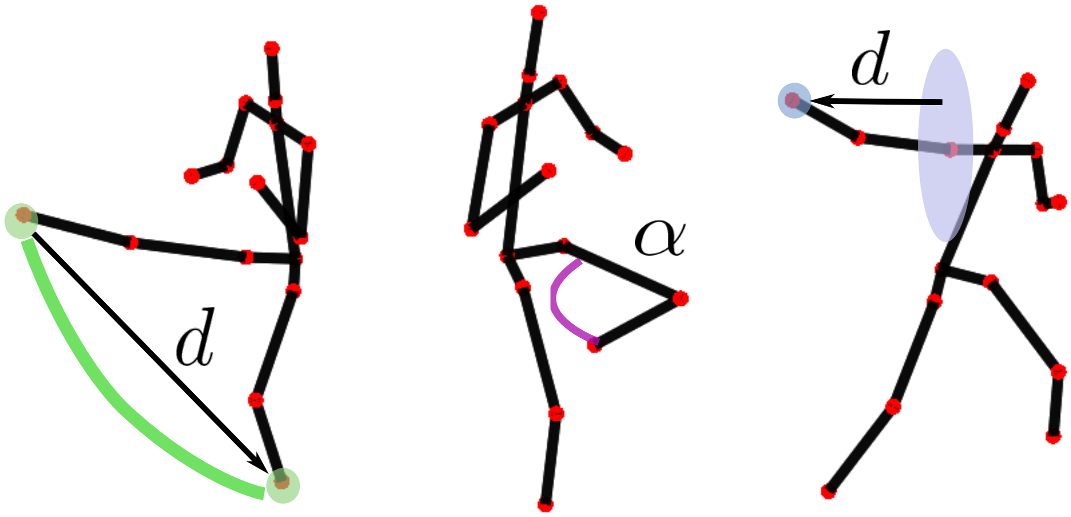


Figure 6.3: The 3 basic posebit types: From left to right, parts distance, articulation angle and relative position. In this example the posebit of type 3 (right image) checks whether the right hand is on the right of the shoulder. The plane is centered at the shoulder and the normal direction is obtained from the vector going from the chest to the shoulder joint.

6.3.1 Automatic Posebit Generation

Every posebit is a bit of information about the pose: it describes the geometric relationship between body parts. To avoid the tedious work of defining each posebit manually we define 3 posebits types that constitute the basis for generating many specific instances. The three kind of posebits are:

1. *Parts distance:* Posebits of the first kind check whether two joints in the body are closer or further than a given threshold.
2. *Articulation angle:* Posebits of the second kind are activated when a given articulation is bent more than α degrees.
3. *Relative position:* Posebits of the third kind check whether a body part A is to the left, right, above, below in front or behind relative to a second body part B. To compute this kind of posebits the signed distance between body part A and a plane centered at body part B is computed. By looking at the sign of the distance we can determine in which half-space of the plane A is located. The plane is always computed from reference points in the body.

Given these three rules, see also Fig. 6.3, we automatically generate a random set of 30 posebits. Our selection Sec. 6.3.4 will rank and retain a set of M posebits out of the pool of candidates. While it may be useful to have an overcomplete set, it requires longer times to label images. As we will see in the experiments $8-10$ posebits are a good trade-off between accuracy and annotation effort.

6.3.2 Learning

We assume here a set of m poseibts generated and selected by our method (Sec. 6.3.4). We infer the set $\mathbf{a} = (a_1, \dots .a_m) \in \mathcal{A}^m$, directly from raw image features $\mathbf{r} \in \mathcal{R}^d$ using a model based on structural SVM [146], where $\mathcal{A} = \{a | a \in \{-1{,}1\}\}$. For convenience we refer to the vector of posebits $\mathbf{a}$ as *posebyte* although we are not restricted to using 8 posebits. Assuming a bounding box of the person, we construct the feature vector $\mathbf{r}$ by computing spatial pyramid features [147], which are spatially localized HOG (Histogram of Oriented Gradients) over increasing cells of sizes 8, 16, 32

and 64 pixels. Histograming over larger windows adds robustness to miss-alignments in the training data. For learning we only require an image dataset with posebit labels $\mathcal{I} = \{\mathbf{r}_i, \mathbf{a}_i\} \in \mathcal{R}^d \times \mathcal{A}^m$. The image dataset consist of image-posebits pairs. Annotations are obtained using Amazon Mechanical Turk [148] as explained in Sec. 6.6. The structural SVM [146] learns discriminant function $F : \mathcal{X}^D \times \mathcal{A} \mapsto \mathbb{R}$ that provides a score for the values the posebyte can take. The posebit can be then estimated by maximizing this function

$$\hat{a} = \arg\max_{a_i} \; F(\mathbf{r},a_i,\mathbf{w}_{a_i}) = \mathbf{w}_{a_i}^T \phi(a_i,\mathbf{r}) \tag{6.1}$$

where $\phi(a_i,\mathbf{r}) = a_i \; \mathbf{w}_{a_i}^T \mathbf{r}$ is the joint feature map of input and output. However, for m number of posebits, this leads to 2^m potential number of classes. Posebits are not independent and we wish to exploit the shared information among classes , *i.e.*, classes with similar posebit strings will be semantically more similar in pose space. Therefore, we learn a discriminant function $G : \mathcal{X}^D \times \mathcal{A}^m \mapsto \mathbb{R}$ over input output pairs from which we can derive prediction by maximizing over the response variable $\mathbf{a}$ for a given input $\mathbf{r}$. The joint SVM scoring function reads:

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a} \in \mathcal{A}^m} \; G(\mathbf{r},\mathbf{a},\beta_{\mathbf{a}}) = \mathbf{a}^T \mathbf{B} \; \mathbf{r} + \mathbf{b}^T \psi(\mathbf{a}) \tag{6.2}$$

where the rows of matrix $\mathbf{B}$ are separating hyperplanes to each of the posebits and $\psi(\mathbf{a})$ is a prior that captures co-occurrences of posebits. For efficiency and to prevent over-fitting we factorize the prior in pair-wise terms

$$G(\mathbf{r},\mathbf{a},\beta_{\mathbf{a}}) = \sum_i^m \mathbf{b}_{a_i}^T \phi(\mathbf{r},a_i) + \sum_i \sum_k b_{a_i,a_k} \psi(a_i,a_k) \tag{6.3}$$

where $\beta_{\mathbf{a}}$ is the vector of weights that we want to learn, $\mathbf{b}_{a_i}^T$ is the i-th row of $\mathbf{B}$ providing the score of posebit a_i and $\psi(a_i,a_k)$ is the pairwise potential computed from mean and variance normalized histograms learned from mocap data. Since the prior only depends on the output variable it can be precomputed resulting in big computational savings. The expression above can be written as a scalar product $\Psi(\mathbf{r},\mathbf{a},\beta_{\mathbf{a}}) = \langle \beta_{\mathbf{a}}, \Phi(\mathbf{r},\mathbf{a}) \rangle$ which gives us a score for a given training input output pair $(\mathbf{r},\mathbf{a_i})$. Having zero training error means that the model scores better the true outputs than any other output. Learning the model weights $\beta_{\mathbf{a}}$ involves solving the following quadratic optimization problem:

$$\begin{aligned}
&\min_{\beta_{\mathbf{a}},\xi} \; \frac{1}{2} \|\beta_{\mathbf{a}}\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i \\
&s.t \quad \forall i, \quad \forall \mathbf{a} \in \mathcal{A}^m \backslash \mathbf{a}_i, \; \xi_i > 0 \\
&\langle \beta_{\mathbf{a}}, \Phi(\mathbf{r}_i,\mathbf{a}_i) - \Phi(\mathbf{r}_i,\mathbf{a}) \rangle \geq 1 - \frac{\xi_i}{\Delta(\mathbf{a}_i,\mathbf{a})}
\end{aligned}$$

The above constraint states that the true output $\mathbf{a}_i$ should score at least a unit better (the margin) than the best runner-up. The objective function penalizes violations of these constraints using slack variables ξ_i. Intuitively, violating a margin constraint involving $\mathbf{a} \neq \mathbf{a_i}$ with a high loss $\Delta(\mathbf{a}_i,\mathbf{a})$ should be penalized more severely. This can be accomplished by scaling the slack variables with the inverse loss $\Delta(\mathbf{a}_i,\mathbf{a})$. The loss function is the Hamming distance ($\Delta(\mathbf{a}_i,\mathbf{a}_j) = \mathbf{a}_i \bigoplus \mathbf{a}_j$) that measures how many posebits are different.

6.3.3 Single frame proposals

Before explaining our method for selecting a good set of posebits it is important to understand how the pose proposals are generated. Therefore, we describe here our method to generate pose proposals given a set m of selected posebits. Hence, we assume that the set of m posebits has been selected already and that a classifier has been trained. Our goal is to draw pose proposals that are semantically similar to the pose observed in the image. The first step of our algorithm is to discriminatively infer a set of posebits denoted as a from image features. Although our framework is not limited to using 8 posebits, let us denote a string of posebits $\mathbf{a} = (a_1, \ldots .a_m) \in \mathcal{A}^m$ as a *posebyte*. The posebits inferred from the raw image features are not always reliable. Let $\mathbf{x}$ denote a 3D pose. Then, we introduce the following proposal distribution that takes into account the uncertainty in the classifier estimation of $\mathbf{a}$:

$$Q_{\mathbf{a}}(\mathbf{x}) = \sum_{\mathcal{A}^m} p(\mathbf{x}|\mathbf{a})p(\mathbf{a}|\mathbf{r}) \tag{6.4}$$

where $p(\mathbf{a}|\mathbf{r})$ is the probability of the posebyte given the raw image feature $\mathbf{r}$, and $p(\mathbf{x}|\mathbf{a})$ is the conditional pose probability given a posebyte. The integral should be computed for every possible posebyte which in the worst case equals 2^m. We approximate the probability $p(\mathbf{a}|\mathbf{r})$ using the trained classifier scores. Let $G : \mathcal{X}^D \times \mathcal{A}^m \mapsto \mathbb{R}$ denote the classifier scoring function that given the raw image feature $\mathcal{X}^D$ provides a score for every posebyte $\mathbf{a} \in \mathcal{A}^m$. For computational efficiency we only consider the top N ranked posebytes and approximate $p(\mathbf{a}|\mathbf{r})$ as a multinomial distribution $p(\mathbf{a}|\mathbf{r}) = \sum_i \pi_i \delta(\mathbf{a} - \mathbf{a}_i)$ with probabilities proportional to the classifier scores $\pi_i \propto G(\mathbf{a}_i,\mathbf{r})$. Specifically, probabilities are computed using soft-max on the top ranked posebytes

$$\pi_i = \frac{\exp\left(G(\mathbf{a}_i,\mathbf{r})/\tau\right)}{\sum_j \exp\left(G(\mathbf{a}_j,\mathbf{r}/\tau)\right))}. \tag{6.5}$$

where τ is the temperature parameter. For $\tau \mapsto \infty$ all posebytes are equally probable and for $\tau \mapsto 0^+$ all the probability mass is allocated on the top ranked posebyte. In this setting, the integral in Eq. (6.4) becomes a conditional class mixture model with weights proportional to the posebyte probabilities

$$Q_{\mathbf{a}}(\mathbf{x}) = \sum_{\mathcal{A}^m} p(\mathbf{x}|\mathbf{a})p(\mathbf{a}|\mathbf{r})\, d\mathbf{a} \sim \sum_n^N \pi_{\mathbf{a}_n} p(\mathbf{x}|\mathbf{a}_n) \tag{6.6}$$

In the simplest case, $p(\mathbf{a}|\mathbf{r})$ is a single delta with probability one at the maximum score $p(\mathbf{a}|\mathbf{r}) = \delta(\mathbf{a} - \mathbf{a}_{\text{max}})$. Note that in the ideal scenario where posebits can be perfectly determined by the image feature $\mathbf{r}$ with no uncertainty the mixture model approaches the true conditional distribution $Q_{\mathbf{a}}(\mathbf{x}) \to Q_{\mathbf{a}}^{\text{opt}}(\mathbf{x}) = p(\mathbf{x}|\mathbf{a}_{gt})$ given the ground truth posebyte $\mathbf{a}_{gt}$. This observation will be important for the selection of posebits, see Sec. 6.3.4.

Sampling: Provided a model for $p(\mathbf{x}|\mathbf{a})$, sampling from the mixture model is straightforward. One can draw samples using Monte Carlo simulations, *i.e.*, sampling a posebyte $\tilde{\mathbf{a}}_i$ with probability $\pi_{\mathbf{a}_i}$ and sampling a pose from $p(\mathbf{x}|\tilde{\mathbf{a}}_i) \sim \tilde{\mathbf{x}}_i$.

To build a model for $p(\mathbf{x}|\mathbf{a})$ we segment the poses in a MoCap database in 2^m classes, one for each posebyte. We then represent each class distribution by computing k-medoids obtaining K representatives $\{\mathbf{x}_{k,\mathbf{a}_n}\}_{k=1}^K$ for class $\mathbf{a}_n$. To avoid unwanted bias we assume the K poses are equally probable, *i.e.*, $p(\mathbf{x}|\mathbf{a}_n) = \sum_{k=1}^K \frac{1}{K}\delta(\mathbf{x} - \mathbf{x}_{k,\mathbf{a}_n})$. In this way, we always sample a fixed set of K codeposes per class, see Fig. 6.4. Therefore, the distribution is modeled by $K \times N$ weighted samples $Q(\mathbf{x}|\mathbf{r}) = \{w_{k,\mathbf{a}_n},\mathbf{x}_{k,\mathbf{a}_n}\}$, with $k \in \{1 \ldots K\}$ and $n \in \{1 \ldots N\}$, with weights $w_{k,\mathbf{a_n}} = \frac{1}{K}\pi_{\mathbf{a}_n}$.

6.3.4 Posebits selection

For selecting a good set of posebits we assume small set of training pairs of images and 3D poses and typically bigger set of 3D poses with no image pairs. The task is to retain m attributes out of candidate set of M. Let $\mathcal{L} = (\mathbf{r}_i,\mathbf{x}_i) \in \mathcal{R}^d \times \mathcal{X}^D$ denote the labeled set and $\mathcal{U} = (\mathbf{x}_i) \in \mathcal{R}^d$ denote the unlabeled set of 3D poses. The core inference algorithm does *not* make use of training pairs $\mathcal{L}$. Conceptually, one would like to retain the set posebits that satisfy the following criteria:

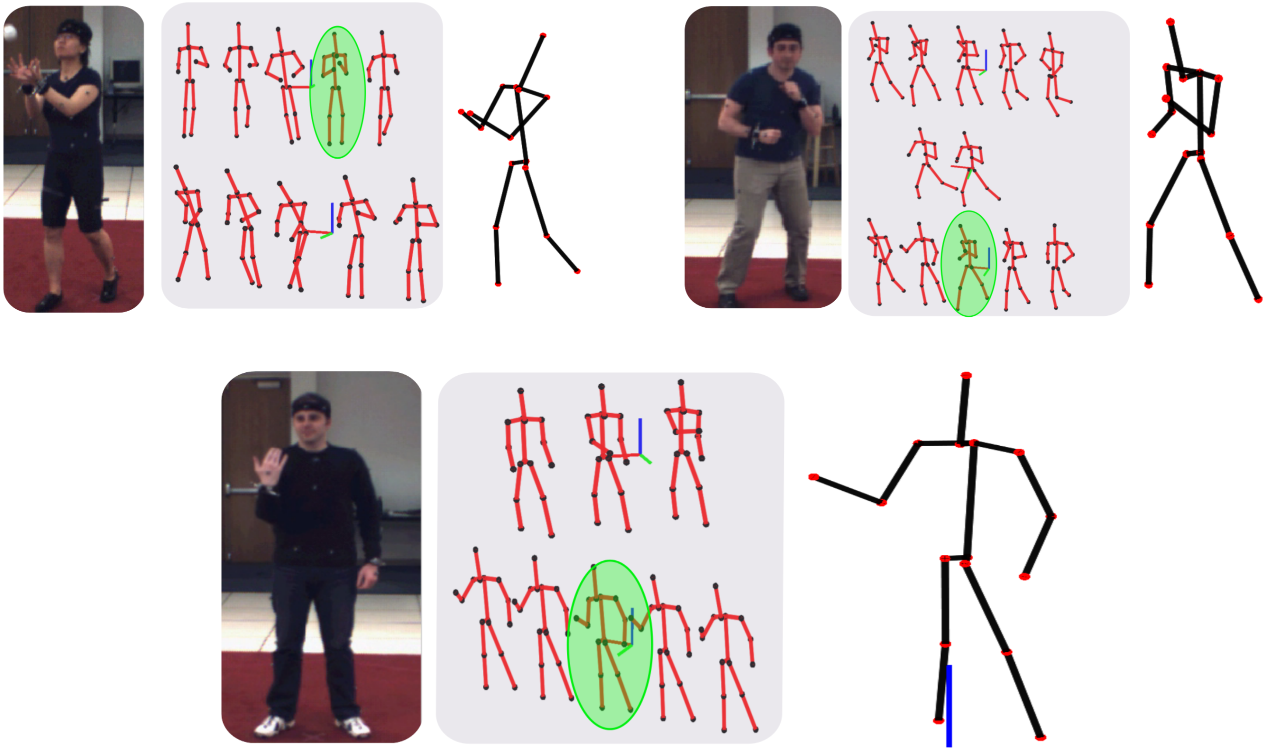


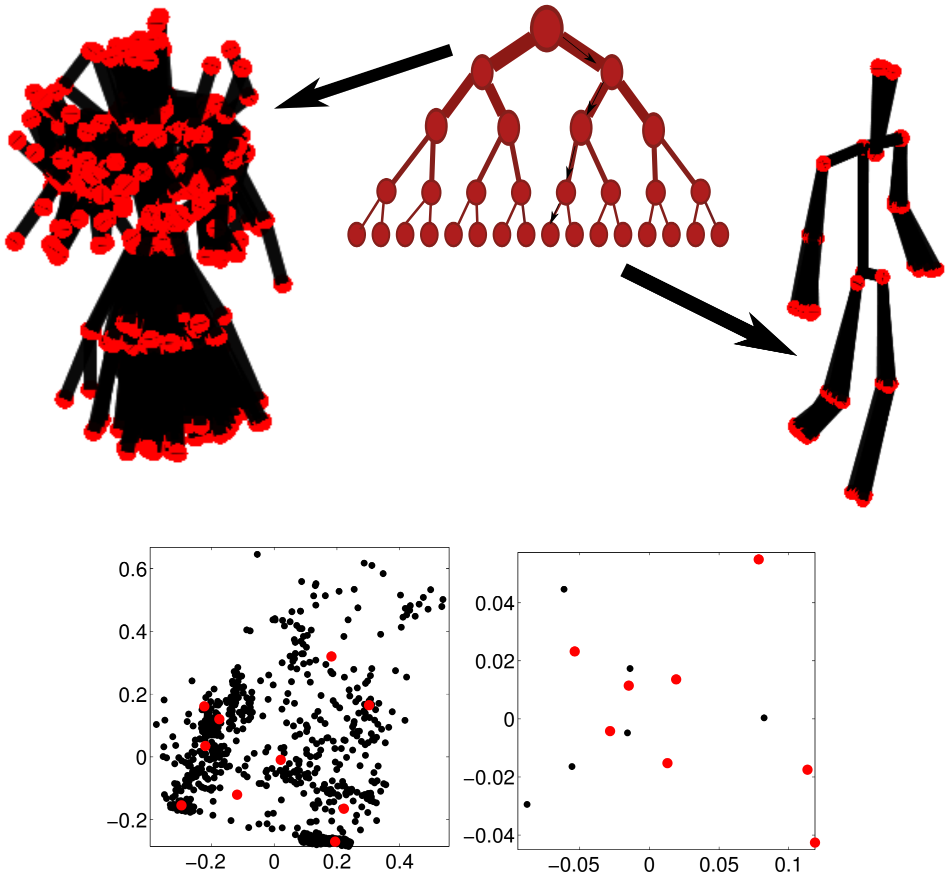
Figure 6.4: Pose estimation results in the HumanEva dataset. We show the input image on the right, the pose proposals in the center and the inferred 3D pose (in black bones) on the right. In the center, each row corresponds to one mixture in our mixture model, where the red poses are the K code poses of the class model.

1. **Reliability:** Attributes that can be correctly inferred from the images. More precisely we want the attributes that better approximate the true conditional class distributions $p(\mathbf{x}|\mathbf{a})$ given only image features.
2. **Clustering:** Attributes that minimize uncertainty about the underlying pose. Therefore, attributes that create tight clusters.

More formally, to satisfy 1) we want our mixture model $Q_\mathbf{a}$ to approximate $Q_\mathbf{a}^{\text{opt}}(\mathbf{x})$ the optimal conditional distribution. Mathematically, this amounts to maximize the expected KL-divergence (D_{KL}) between distributions $\mathbb{E}_\mathbf{r}\{D_{\text{KL}}(Q_\mathbf{a}^{\text{opt}}(\mathbf{x}),Q_\mathbf{a}(\mathbf{x}))\}$, where the expectation is over the image features $\mathbf{r}$. This enforces either to select posebits with low classification errors or posebits with classification errors that have a low impact on the conditional pose distributions. To satisfy the second criteria we have to minimize the entropy of the conditional distributions. When selecting posebits there are several things to consider, for example a posebit might be very useful in its own but become redundant given another, or vice-versa. There might be different subsets of posebits that are suited for pose estimation; obtaining the optimal set involves a big combinatorial problem that we want to avoid. Therefore, we select them greedily using a forward selection method based on information gain similar to decision trees. Let $\mathcal{S}_\mathcal{A}$ denote the pool of posebits from which we want to retain a small subset $\mathcal{S}_m$. The goal is to maximize the information gain every time we include an additional attribute. The objective function is

$$a_i^* = \arg\max_{a_i \in \mathcal{S}_\mathcal{A}} \quad I_i = I_i^\mathcal{C} + \mu \cdot I_i^\mathcal{R} \tag{6.7}$$

where I_i is the mixed information gain at the *i-th* level of the tree consisting of a *reliability* $I_i^\mathcal{R}$ term and a *clustering* $I_i^\mathcal{C}$ term.



(b) (c)

Figure 6.5: At the top the tree the entropy is high and the poses have high variance, at the bottom, not only the variance is vastly reduced but also the poses are clustered in semantic classes. On the top left we show a few poses conditioned on a single posebit, on the top right we show the poses conditioned on 10 posebits. On the bottom row we show the PCA projection of the cluster poses, black dots are poses and the red dots are the k-medoids of the cluster. The variance is vastly reduced, notice the scale of the axis differs by one order of magnitude.

Clustering Every time we add a posebit to the set, the expected clustering information gain is

$$I_i^{\mathcal{C}} = H_{i-1} - H_i \tag{6.8}$$

where H_i is the expected entropy at the *i-th* level computed as

$$\mathbb{E}_{p(c)}\{H_i\} = \sum_{c=1}^{2^i} \frac{|\mathcal{S}_c^y|}{|\mathcal{S}_i^y|} H(\mathcal{S}_c^y) \tag{6.9}$$

where $\mathcal{S}_c^y \subseteq (\mathcal{U} \cap \mathcal{L})$ denotes the subset of poses $\mathbf{x}_i$ in cluster c. Here, $H(\mathcal{S})$ is the continuous differential entropy of the pose density at the cluster. In practice, however that is difficult to estimate; therefore we follow the standard approach of minimizing the cluster variance. While this is a very crude assumption specially at the first levels of the tree, it is a measure of cluster compactness and works well in practice. In Fig. 6.5 we show how the entropy in the poses is drastically reduced as we traverse the tree to the bottom.

Reliability It is also desirable that posebits can be inferred reliably from image features. To measure the information loss of using our mixture model $Q_{\mathbf{a}}(\mathbf{x})$ as opposed to $Q_{\mathbf{a}}^{\text{opt}}(\mathbf{x})$ we use the KL-divergence between the two distributions. Let D_{KL}^i denote the expected KL distance at level i of the

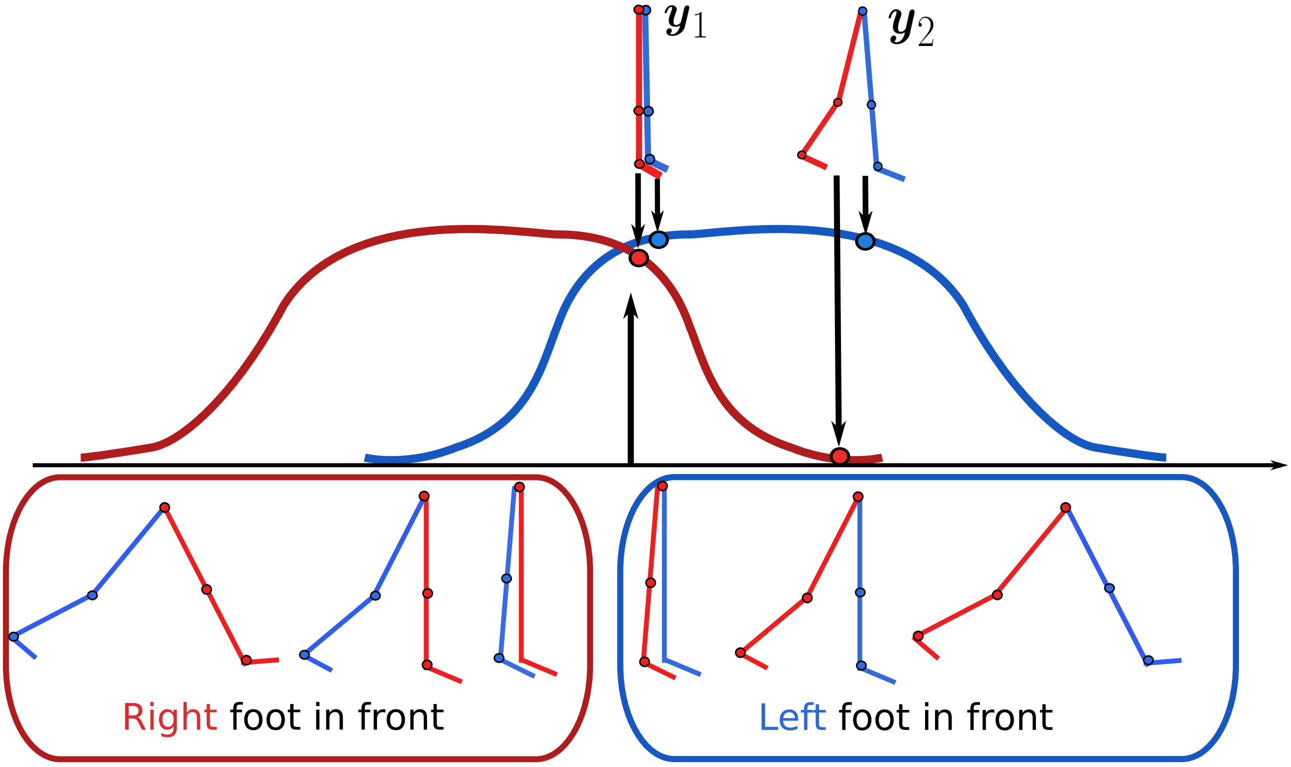


Figure 6.6: Reliability objective: Posebits that can be correctly inferred from images should be preferred. However, certain miss-classifications have a more severe loss in the pose estimation. Consider the simple example for a posebit that checks whether the right foot is in front of the left foot. The two test poses shown above $\mathbf{x}_1$,$\mathbf{x}_2$ both belong to the right class (blue): left foot in front. A miss-classification error for pose $\mathbf{y_2}$ clearly results in a bad pose prior. A miss-classification error for pose $\mathbf{y_1}$ on the other hand is not so important since pose $\mathbf{y_1}$ is also relatively well explained by the other class.

tree

$$D^i_{\text{KL}} = \mathbb{E}_{\mathbf{r}}\{D_{\text{KL}}(Q^{\text{opt}}_{\mathbf{a}}(\mathbf{x}),Q_{\mathbf{a}}(\mathbf{x}))\} \quad \text{with} \quad \mathbf{a} \in \mathcal{A}^i. \tag{6.10}$$

The reliability information gain of adding an additional posebit to the string is then $I^{\mathcal{C}} = D^{i-1}_{\text{KL}} - D^i_{\text{KL}}$ which measures how much closer we get to the ground truth distribution. When this quantity is negative it means that the approximation gets worse as we go down the tree. Note that although $Q^{\text{opt}}_{\mathbf{a}}(\mathbf{x})$ is harder to approximate as we go down the tree, the information gain increases as $Q^{\text{opt}}_{\mathbf{a}}(\mathbf{x})$ carries much more information about the pose. We compute the discrete KL-divergence using training examples in the labeled set $\mathcal{L} = \{\mathbf{a}_i,\mathbf{x}_i,\mathbf{r}_i\}$

$$\begin{aligned} D^i_{\text{KL}} &= \sum_{i\in\mathcal{L}} Q^{\text{opt}}_{\mathbf{a}}(\mathbf{x}_i) \log\left(\frac{Q^{\text{opt}}_{\mathbf{a}}(\mathbf{x}_i)}{Q_{\mathbf{a}}(\mathbf{x}_i)}\right) \\ &= \sum_{i\in\mathcal{L}} p(\mathbf{x}_i|\mathbf{a}_i) \log\left(\frac{p(\mathbf{x}_i|\mathbf{a}_i)}{\sum_n^N \pi_{\mathbf{a}_n} p(\mathbf{x}_i|\mathbf{a}_n)}\right) \end{aligned} \tag{6.11}$$

This penalizes posebits with classification errors far away from the border between classes more severely [1]. This, in turn, will favor potentially very useful posebits despite having only modest classification accuracies as we will see in the experimental section. Note that this measure is also related to maximum likelihood.

---

[1] To avoid having to train a structural SVM every time we test a new posebit, during selection we use a mixture with only one component which allows us to use a precomputed single svm classifier.

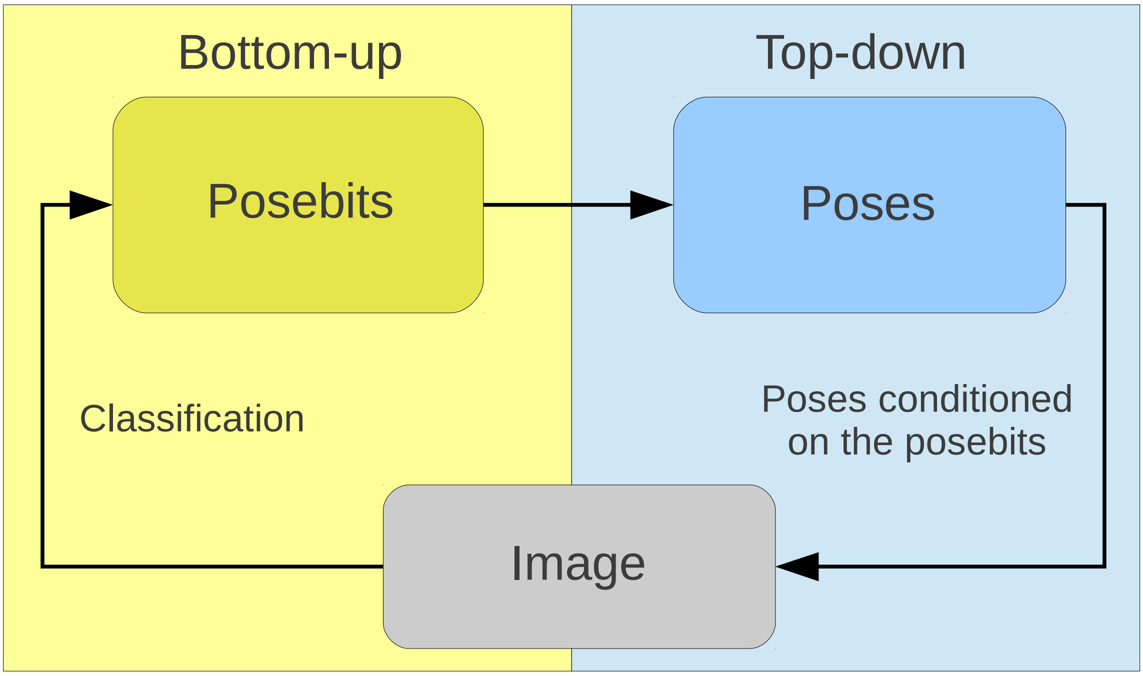


Figure 6.7: Diagram of the proposed pose estimation method: Posebits are inferred directly from image features bottom-up. These posebits are then used to constraint the pose space and sample plausible hypothesis. The likelihood of the hypotheses are finally evaluated against the image top-down.

6.4 3D Pose Estimation

We show here how the proposal distribution $Q(\mathbf{x}|\mathbf{r})$ (Sec. 6.3.3) can be used to improve pose estimation. We formulate pose estimation in a top-down generative model where the proposal distribution $Q(\mathbf{x}|\mathbf{r})$ is used to generate the hypotheses. In tracking, hypotheses are typically sampled from a proposal distribution given past measurements in a sequence. In our setting however, the proposal distribution is formed from the posebits inferred bottom-up from the image. Given image observations $\mathbf{z}$, of different nature than $\mathbf{r}$ (the ones we used for posebit inference), the posterior can be written as

$$p(\mathbf{x}|\mathbf{z},\mathbf{r}) \propto p(\mathbf{z}|\mathbf{x},\mathbf{r})p(\mathbf{x}|\mathbf{r}) \simeq p(\mathbf{z}|\mathbf{x})Q(\mathbf{x}|\mathbf{r}) \tag{6.12}$$

where we assume that $\mathbf{z}$ and $\mathbf{r}$ are conditionally independent given that the pose $\mathbf{x}$ is known, analogous to filtering where it is assumed that $\mathbf{z}$ is independent of previous measurements given the state.

Image Likelihood: A number of excellent papers can be found focusing on the design of high fidelity likelihood models such as [149]. Although the likelihood is a key ingredient in pose estimation it is not the focus of our work. Therefore, we assume here that unlabeled 2D joint locations are available. 2D locations could be obtained from a 2D pose estimation algorithm for example. Hence, the additional image features $\mathbf{z} = (\mathbf{m}_1 \ldots \mathbf{m}_J)$ consist of a collection of 2D points $\mathbf{m}_i \in \mathbb{R}^2$. Let $\mathcal{F}(\mathbf{x}; j) : \mathcal{X}^D \mapsto \mathbb{R}^3$ be a function that maps a pose $\mathbf{x}$ to the j-th 3D joint position. We model $p(\mathbf{z}|\mathbf{x})$ in Eq. (6.12) as a product of isotropic 2D Gaussians centered at the joint locations

$$p(\mathbf{z}|\mathbf{x}) = \frac{1}{C} \exp\left(-\sum_{i}^{P} e^2(\mathbf{m}_i|\mathbf{x})\right) \tag{6.13}$$

where C is a normalization constant, and $e(\mathbf{m}_i|\mathbf{x})$ is the Euclidean distance between the part location and the closest projected 3D joint location in the image,

$$e(\mathbf{m}_i|\mathbf{x}) = \min_j \|\mathbf{m}_i - \mathrm{Proj}(\mathcal{F}(\mathbf{x}; j))\|_2. \tag{6.14}$$

Here, Proj is the function projecting 3D points to the image plane. We assume an orthographic camera model where the scale is set to match the person's height in the image plane. We find the pose estimate as the mode of the posterior $p(\mathbf{x}|\mathbf{z},\mathbf{r})$ in Eq. (6.12) by evaluating the $K \times N$ poses of the proposal distribution $Q(\mathbf{x}|\mathbf{r}) = \{\mathbf{x}_{k,\mathbf{a}_n}, w_{k,\mathbf{a}_n}\}$. Recall, that $Q(\mathbf{x}|\mathbf{r})$ was represented by K poses for each of the N classes of the top ranked posebytes. The root orientation is estimated by uniformly sampling the rotation angle θ_{root} about the vertical axis at 32 equally spaced angles. Let $\mathcal{M}(\theta; \mathbf{x})$ denote the function that rotates a pose $\mathbf{x}$ by θ degrees. Then, the pose estimate is obtained by maximizing

$$\mathbf{x}^* = \arg\max_{\mathbf{x}_{k,\mathbf{a}_n}} \left(\max_{\theta_{\mathrm{root}}} \left(p\left(\mathbf{z}|\mathcal{M}(\theta_{\mathrm{root}}; \mathbf{x}_{k,\mathbf{a}_n})\right) w_{k,\mathbf{a}_n} \right) \right)$$

where $\mathbf{x}_{k,\mathbf{a}_n}$ is the k-th pose of the class corresponding to posebyte $\mathbf{a}_n$ and $w_{k,\mathbf{a}_n} \propto p(\mathbf{x}|\mathbf{r})$ are the importance sampling weights explained in Sec. 6.3.3. Further implementation details are given in Sec. 6.7

6.5 Posebits database

To evaluate our method we collected a new database that we call the The *Posebits Database PbDb*. The *Posebits Database PbDb* supplements previously published databases with posebit annotations. Specifically, it contains images from HumanEva-I [109], HMODB [25], Fashion [150] and Parse datasets [48]. To test our proposed method we collected a new database that we call *posebits database (PbDb)*, which consists of two independent sets: about 5000 annotated images and a MoCap corpora that is segmented in semantic clusters driven by the selected posebits. We collected the images from 4 publicly available databases: 1500 images Human-Eva [109], 1500 images from HMODB [25], 685 images Fashion[150] and 305 images from Parse [48]. Human-Eva and HMODB contain image-3D pose pairs which we used to simulate the annotations. For the Fashion and Parse datasets we collected annotations from Amazon Mechanical Turk. We split the image datasets in two subsets of 2500 images for training and testing. Every image contains an annotation consisting of 30 semantic questions(posebits). We automatically generated a pool of 30 posebits. We define 3 core rules that generate 3 different kind of posebit instances: 1) a posebit of the first kind checks whether two body parts are close or far apart, 2) a posebit of the second kind is activated when a given articulation is bent. The third type checks the spatial arrangement between two parts in the body, that is, to the left,right, in front or behind. Although we are more interested in semantic pose inference, Human-Eva is a standard benchmark for pose estimation algorithms.

Specifically, (PbDb) consists of two independent databases, see also Tabl. 6.1, 6.2:

1. *Posebit Annotated Image Database*: This database consists of a set of images with corresponding posebit annotations. Each image is annotated with 30 posebits. The images have been annotated by workers using Amazon Mechanical Turk [148]. For the databases that contain $3D$ pose information we have simulated the posebits directly from the 3D poses.

2. *Posebit MoCap Database*: This database consists of a set of MoCap poses obtained from the HumanEva-I and HMODB databases. Given a subset of m selected posebits this database contains poses clustered in 2^m separate classes.

These are two independent sets that are both used at test time during inference. It is important to note that in contrast to previous databases, PbDb does not make use of training pairs of poses and images. The poses in the Posebit MoCap database are all scaled to a unit pose, *i.e.*, all bones are re-scaled to the size of a template pose. In addition, all poses are centered to the origin and the yaw angle [2] is removed. This pre-processing step has proven crucial to compute the clustering objective during posebit selection. One nice feature of PdDb is that it allows to do semantic queries, see Fig. 6.13.

---

[2] The viewpoint w.r.t. the camera is arbitrary

Table 6.1: Posebits Annotated Image Database: we show, for each database, the number of frames, the type of posebit annotations available and rank each database according to its level of pose variability and background clutter.

Database	Number of frames	Posebit Annotation	Pose variability	Background clutter
HumanEva-I	10350	Simulated	Little	Little
HMODB	12370	Simulated	Very high	High
Parse	305	Human Annotation	Very high	Very High
Fashion	685	Human Annotation	Moderate	High

Table 6.2: Posebits MoCap Database: we show for each database the number of poses.

Database	Number of Poses	Motion Patterns
HumanEva-I	29722	Walking, Jogging, Gestures, Throw and Catch and Boxing
HMODB	12370	Walking, Running, Kicking, Rotate Arm Gymnastics, Tennis, Soccer and Basket

6.6 Annotation with Amazon Mechanical Turk

For HumanEva-I and HMODB, since we have 3D pose information, for each image we can easily simulate the image annotations. Unfortunately, it is a very laborious and difficult task to capture such databases since synchronization of MoCap and image data is difficult. Most importantly, motion capture systems are intrusive and is therefore almost impossible to obtain 3D pose of people into the wild, *i.e.*, in the streets, in the office or during sport events. Annotating posebits in images on the other hand simplifies the task to answering simple yes/no questions for which we use Amazon Mechanical Turk [148].

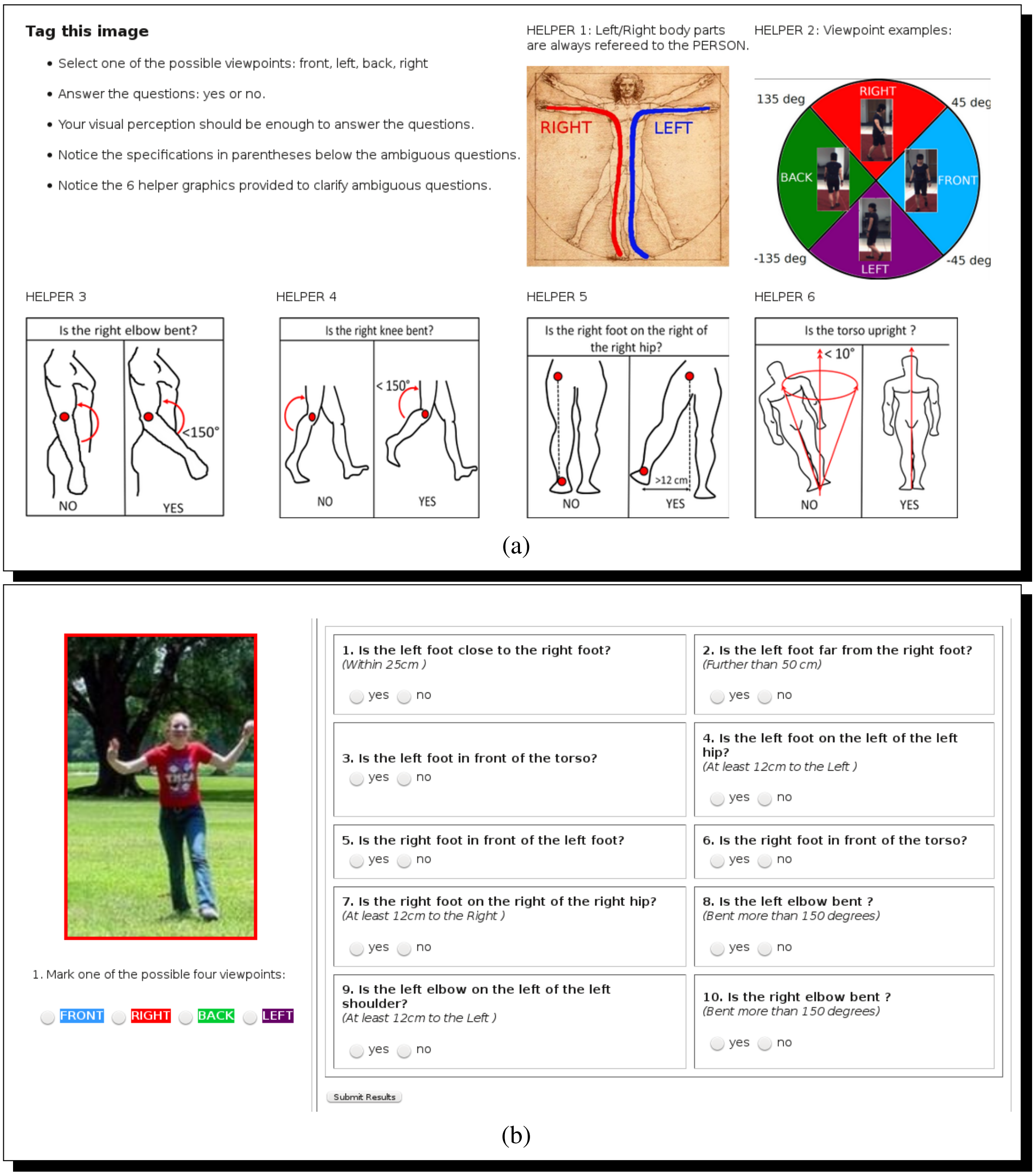
We split the set of 30 automatically generated posebits into 3 sets of 10 questions. Hence, a worker in AMT is shown an image along with 10 simple questions to answer, see Fig. 6.8. The interface questions and layout is also automatically generated from the list of posebits. It takes on average from 20 to 40 seconds for untrained workers to annotate one image. This shows that collection of data is substantially easier than annotating $3D$ pose which for trained users takes well over 5 minutes.

6.7 Setup and Parameter Settings

In this section we detail the parameter settings and experimental setup we used for posebit selection, learning and pose estimation.

6.7.1 Learning of Posebits Classifier

To obtain a more balanced set across the 4 different databases: HumanEva, HMODB, Parse and Fashion, we sub-sampled a subset of images to train our image classifier. Specially, we used a subset of 1500 images from HumanEva training set, 1500 images from HMODB, 305 from Parse and 685 from the Fashion dataset making a total of 3990 images. We further split it into two equal sets of 1995 images for training and 1995 images for testing. The image classifiers used in all the experiments were trained on these 1995 training images. For the validation our image classifier we used the test set of 1995 images. Our input features $\mathbf{r}$, are spatial pyramid features [147], which are spatially localized HOG (Histogram of Oriented Gradients) over increasing cells of 8,16,32 and 64. Histograming over larger windows adds robustness to miss-alignments in the training data.



Tag this image

- Select one of the possible viewpoints: front, left, back, right
- Answer the questions: yes or no.
- Your visual perception should be enough to answer the questions.
- Notice the specifications in parentheses below the ambiguous questions.
- Notice the 6 helper graphics provided to clarify ambiguous questions.

HELPER 1: Left/Right body parts are always refereed to the PERSON.

HELPER 2: Viewpoint examples:

HELPER 3

HELPER 4

HELPER 5

HELPER 6

(a)

1. Mark one of the possible four viewpoints:

○ FRONT ○ RIGHT ○ BACK ○ LEFT

1. Is the left foot close to the right foot?
(Within 25cm)
○ yes ○ no

2. Is the left foot far from the right foot?
(Further than 50 cm)
○ yes ○ no

3. Is the left foot in front of the torso?
○ yes ○ no

4. Is the left foot on the left of the left hip?
(At least 12cm to the Left)
○ yes ○ no

5. Is the right foot in front of the left foot?
○ yes ○ no

6. Is the right foot in front of the torso?
○ yes ○ no

7. Is the right foot on the right of the right hip?
(At least 12cm to the Right)
○ yes ○ no

8. Is the left elbow bent ?
(Bent more than 150 degrees)
○ yes ○ no

9. Is the left elbow on the left of the left shoulder?
(At least 12cm to the Left)
○ yes ○ no

10. Is the right elbow bent ?
(Bent more than 150 degrees)
○ yes ○ no

(b)

Figure 6.8: Mechanical Turk annotation interface: (a) Helpers shown to workers, (b) The interface, 10 simple questions to be answered by annotators.

6.7.2 Posebit Selection

We use two criteria to select the posebits greedily: clustering and reliability.

Reliability: To compute the reliability objective we use a labeled set of image-pose pairs $\mathcal{L} = (\mathbf{r}_i, \mathbf{x}_i) \in \mathcal{R}^d \times \mathcal{X}^D$. The labeled set $\mathcal{L}$ consists of 2000 additional images taken from HumanEva-I and HMODB. These images are only used for selection. To avoid having to train a structural SVM every time we test a new posebit, we pre-train a single SVM for each of the 30 posebit candidates using the same subset of 1995 images from the Posebit Annotated Image Database. Then, for every training example in the labeled set $\mathcal{L}$ we run the learned classifier obtaining a score π for every possible posebyte string in the set $\{\mathbf{a}_n, \pi_{\mathbf{a}_n}\}_{n=1}^{n=2^m}$. For computational efficiency, during selection, we take the top ranked posebyte with weight $\pi_{\mathbf{a}_{max}} = 1$. To evaluate Eq. (6.11) we compute the likelihood of pose $\mathbf{x}_i$ under our estimated single mixture distribution $Q_{\mathbf{a}}(\mathbf{x})$ and under the ground truth distribution $Q_{\mathbf{a}}^{\text{opt}}(\mathbf{x})$, see Fig. 6.6
NOTE: In a scenario where a labeled set is not available, one option is to directly use the classifier accuracies on a validation set to compute a related measure. We also have found good results following this approach. We wish however to penalize more severely errors for poses that are not on the border between classes but clearly belong to one class.

Clustering: Additionally, to compute the clustering information gain we use the a set of 5000 unlabeled poses from the Posebit Mocap Database $\mathcal{U} = (\mathbf{x}_i) \in \mathcal{R}^d$. To compute this objective the cluster pose entropy or a related measure needs to be calculated.
NOTE: Here we considered several options: one could for example use kernel density estimation (KDE) or fit a mixture of Gaussians (GMM) to the cluster samples. Unfortunately, it is well known that KDE is problematic in high dimensional spaces and GMM is computationally expensive and one would have to resort to approximations of the entropy as there exists no closed form solution. In decision trees, people often make the working assumption of Gaussian distribution at the leafs because it is fast to compute and the Gaussian has a closed form solution for the entropy $H\left(\mathcal{N}(\mu;\Sigma)\right) = \log\left(\det(\Sigma)\right.$, where $\det\left(\Sigma(\mathcal{S}_c^y)\right)$ is the determinant of the covariance matrix of the subset. However, we found the Gaussian approximation to be numerically unstable when the covariance was singular or close to singular.
Therefore, since we are not concerned in estimating the entropy very accurately but rather in selecting good posebit candidates we follow a standard pose variance minimization which proved to be numerically much more stable.

6.7.3 3D Pose Estimation

Our likelihood model defined in Eq. 6.13 is as simple as it can be: we simulate a set 2D joint locations detections in the image by projecting the ground truth 3D pose locations on the image. We take as input this set of 2D locations without any body part label. While working with real 2D joint detections makes the problem harder, the problem we tackle here is still heavily i-ll conditioned, the depth and left right ambiguities still persist. For the validation of our pose estimator we used the validation set of the HumanEva-I, which roughly amounts to 4000 images. Here we note that we needed to test our algorithm in the validation set because we are currently using projected 2D joint locations as input for our image likelihood. For our mixture model in Eq. 6.6 we use a fixed number of 4 mixtures with the temperature parameter of soft-max τ set to 0.5. Every class is described by $K = 10$ representatives. To compute the likelihood, for every mixture, every codepose in the mixture class is projected to the image at 32 uniformly sampled yaw angle locations. The scale is estimated from the ratio between the unit size train pose and the image plane height. The image plane height is estimated as the maximum minus the minimum $2D$ joint locations in the image. We note that the likelihood model is a very important ingredient for pose estimation; it is however not the focus of our work and therefore we use the simplest model here.

6.8 Experiments

We show here several experiments in order to answer the following questions:

1. *Selection*: Are the posebits retained by our selection method better for inference than a random subset?
2. *Classification*: Is it possible to infer posebits directly from image features with sufficient accuracy?
3. *Pose Estimation*: Are the prior poses qualitatively similar to the observed pose ?
4. *Pose Estimation*: Are posebits improving the pose estimation accuracy ?

To validate our proposed method we present two potential applications of posebit inference: 1) for monocular pose estimation, 2) for semantic pose estimation and retrieval.

Selection To understand the influence of posebit selection we compared the performance of our 10 ranked posebits with the performance of picking random subsets of 10 posebits. For this experiment we tested our method using 20 different random subsets of 10 posebits and averaged the results. We can see in Fig. 6.9 we perfom significantly better on average which means that the selection provides a good set of posebits to use. It is important to note that the selected posebits are not necessarily the ones with better classification scores as those might be uninformative or redundant together with others. This is reflected in Fig 6.11 where we show the classification accuracy for the pool of 30 posebits. Other combinations might work equally well but due to computational constraints we are not seeking for the optimal subset but rather a good set.

Classification In Fig 6.10 we show the classification accuracies in the test sets of the 4 datasets, Heva, HMODB, Fashion and Parse. As we can observe, our model can predict posebits from images with remarkably high accuracies (70-90%). The dataset where we perform more modestly is Parse. That is probably due to the high variability in pose and appearance and due to the fact that we only use 150 images for training which is one order than magnitude less than for the other datasets. In Fig. 6.14 we show a few examples of poses sampled from our mixture model for different test images of the PbDb. We perform significantly better in the images from HumanEva-I, HMODB and Fashion both in classification accuracies and semantic quality of the 3D estimation results. This is due to the fact that the data has more redundancy and therefore learning works better. In addition, our MoCap database is still limited. To improve our predictions we are currently extending both the Annotated Image Database and the MoCap database to cover a wider range of poses and appearances. In Fig. 6.11 we show the accuracies of the pool of 30 posebits. Notice that our algorithm selects the most informative posebits (according to the criteria explained in the selection section) which are not necessarily the ones with the better classification accuracies. It is interesting to see that our selection method selects a subset of symmetric posebits (*i.e.*, left and right counterparts) which is what intuitively one might expect.

3D Pose Estimation: First of all, we want to demonstrate the usefulness of using posebits to cluster poses in semantic classes. In Fig. 6.13 we query the PbDb for 3 posebits. As it can be seen in Fig. 6.13 as little as 3 posebits already cluster poses in qualitatively similar classes. This allows us to

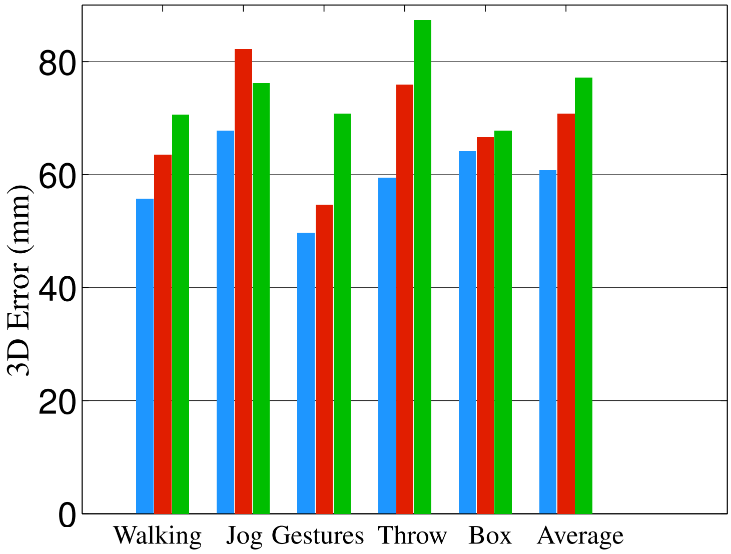


Figure 6.9: Posebit selection: 3D error (mm), we show in red the error using 10 selected posebits, in green we show the average error of 20 different random subsets of 10. We also include in blue the accuracy given the ground truth posebits to see the impact of classification errors in the final pose results.

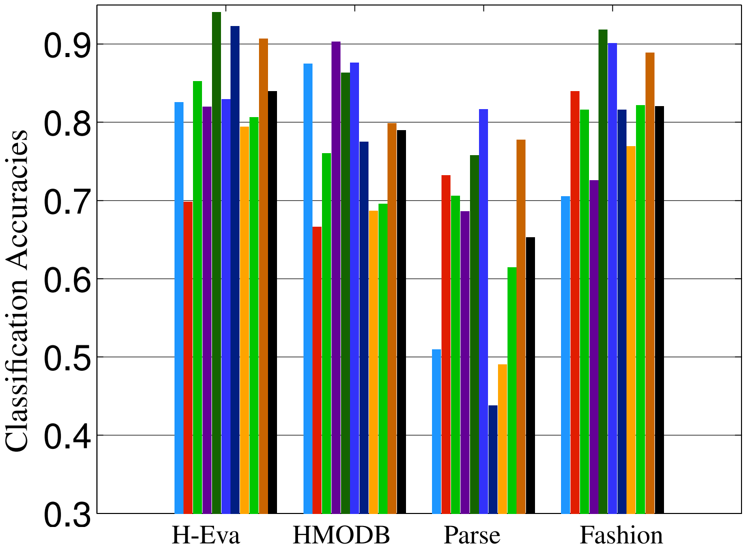


Figure 6.10: Classification accuracies for top 10 ranked posebits selected by our algorithm. The average classification accuracy is shown in black. We group the results according to databases, from left to right: Human-Eva, HMODB, Parse and Fashion. As we can obseve we obtain very good accuracies for Human-eva, HMODB and Fashion. On the Parse datasets some of the posebits can not be reliably detected due to the high variability in the poses seen in the images.

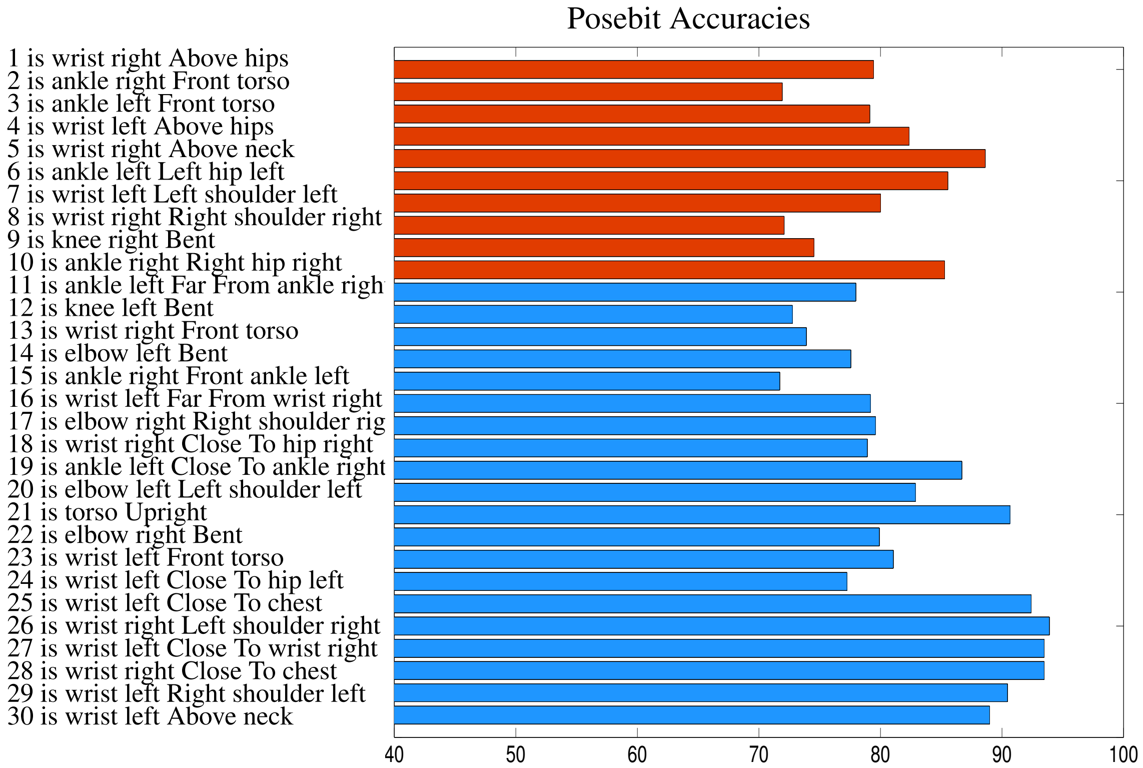


Figure 6.11: Posebit accuracies on the test set including images from HumanEva, HMODB, Parse and Fashion. Posebits are sorted from top to bottom according to the ranking assigned by our selection algorithm. As it can be seen, posebits with very high accuracies are not necessarily selected; they might carry little information about the pose or they might be redundant with other posebits. One example of this is the posebit *is the torso upright?* which ranked very well on its own but did not add any further information when the posebits 2 and 3 *is the left/right ankle in front of the torso* were included. That is because when the feet are both above the torso it usually corresponds to a non-upright torso, see the last row of Fig. 6.14. We show in red the 10 top ranked posebits that we use in most of our pose estimation results.

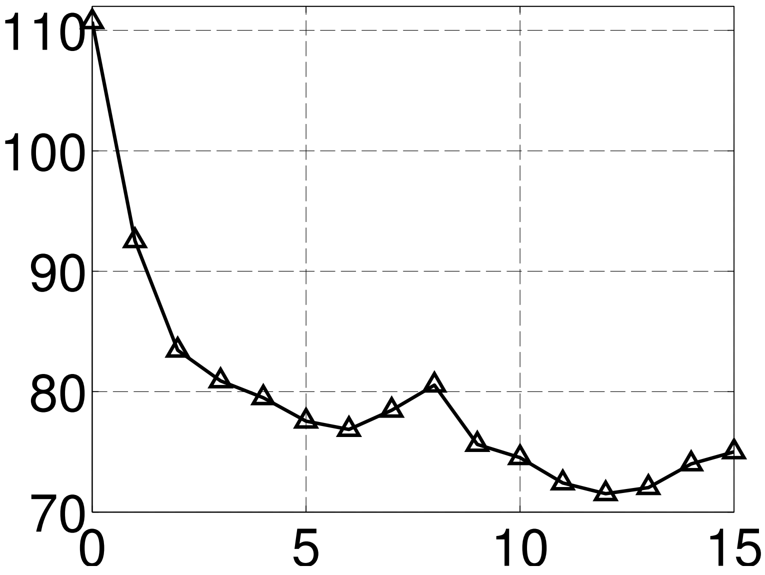


Figure 6.12: We show here the mean 3D pose error (mm) as a function of the number of posebits. Notice the significant improvement decreasing the error from 110 mm down to almost 70 mm when using 12 posebits. This clearly demonstrates the usefulness of using posebits during inference.

either retrieve similar images or similar poses. In Fig. 6.14 we show how the inferred posebits can be used to retrieve poses that are qualitatively similar to the observed image. This forms the basis of our algorithm to deal with typical monocular ambiguities.

To obtain a quantitative validation, we reserved a set of 4000 images from Heva for testing our model. The test set includes images from all motion patterns and actors. In Fig. 6.12 we show the mean pose error as a function of the number of posebits. As expected, the more posebits we use, the more clustered the proposals are and the less ambiguity, hence a much better performance is obtained. The best results are obtained using 12 posebits; nonetheless in the next experiments we used only 10 as we believe it is a good trade-off between accuracy and annotation effort to collect training data. Notice the big drop in pose error as we increase the number of posebits. Posebits can also potentially reduce the *computational burden*, the more bits of information we infer beforehand the less poses have to be evaluated against the image for the same error requirements. Our current unoptimized Matlab implementation runs at an average of 22 frames per second using 10 posebits, 4 mixtures and 10 code-poses per cluster.

Image Retrieval: Finally, we show another potential application of posebits for image retrieval of images related semantically by the pose. The top ranked posebytes by our classifier are used to retrieve annotated images in the database with the similar posebyte strings, see Fig 6.15.

6.9 Discussion

Our model is still limited to images where the full pose is visible. In addition, we do not have yet enough variation in our training data to obtain excellent performance in very difficult images with occlusions and lots of background clutter. Also, retaining a fixed subset of posebits has the advantage that a powerful model can be learned but is less flexible than choosing posebits at test time. Indeed, some preliminary experiments reveal that the posebit confidences are strongly correlated to accuracies. Therefore, one strategy we are also analyzing is to choose a variable subset of posebits at test time. We are currently extending both the Annotated Image Database and the MoCap database to

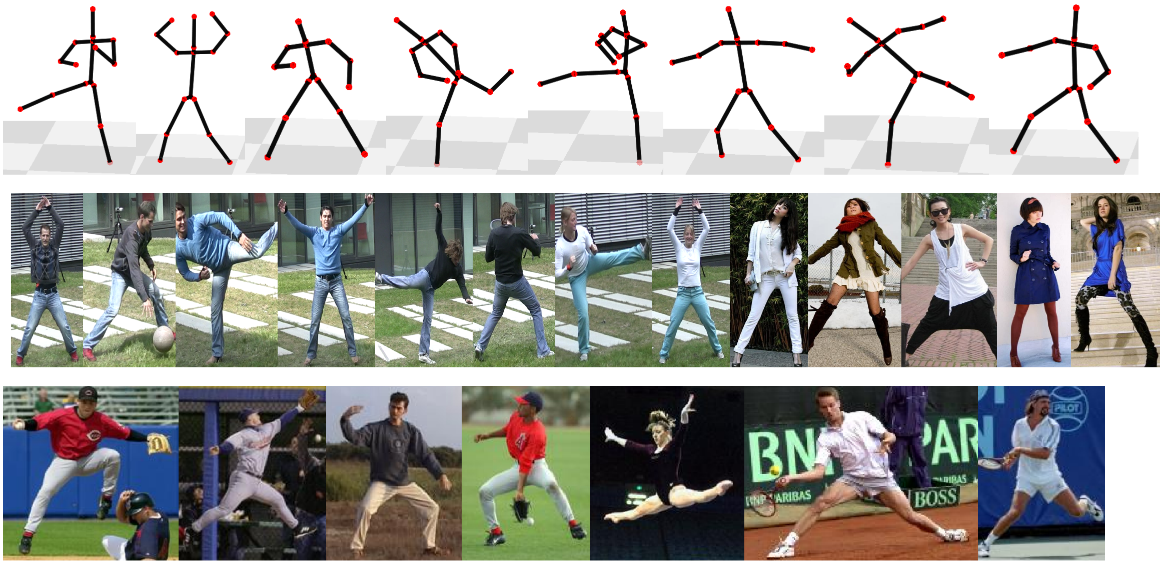


Figure 6.13: Semantic queries: in this example we queried the Posebit database for 3 posebits: *left foot to the left of the hip, right foot to the right of the hip and left foot far from the right foot*. On the top row we show a few example poses retrieved from the MoCap database and on the bottom rows we show examples retrieved from the Annotated Image database. Notice that the poses are already quite clustered semantically by just fixing as little as 3 posebits .

cover a wider range of poses and appearances. We have demonstrated that posebits can also improve 3D estimation accuracy. We want to emphasize however, that for many applications a semantic inference of pose might be sufficient or even more powerful. This first results are very encouraging and we strongly believe that posebits open a promising line of research in the pose estimation field.

6.10 Summary and Future Work

We have introduced posebits which constitute a semantically powerful pose descriptor. Experiments show that our selection method learns a good set of posebits, *i.e.*, retains the ones that can be reliably inferred from images and are informative about the pose. We have also shown that using posebits as a mid-layer representation can improve monocular pose estimation. One of the main advantages of the proposed method is that annotation is easier and more intuitive for the human observer. This enables easy collection of training data. The experiments reveal that posebits can resolve many of the monocular ambiguities and can be useful as basis for many potential applications. In particular, we do not see posebits as a competitor to existing approaches but rather as a powerful complementary feature. For future work, we plan on annotating more data, and to explore more posebit applications.

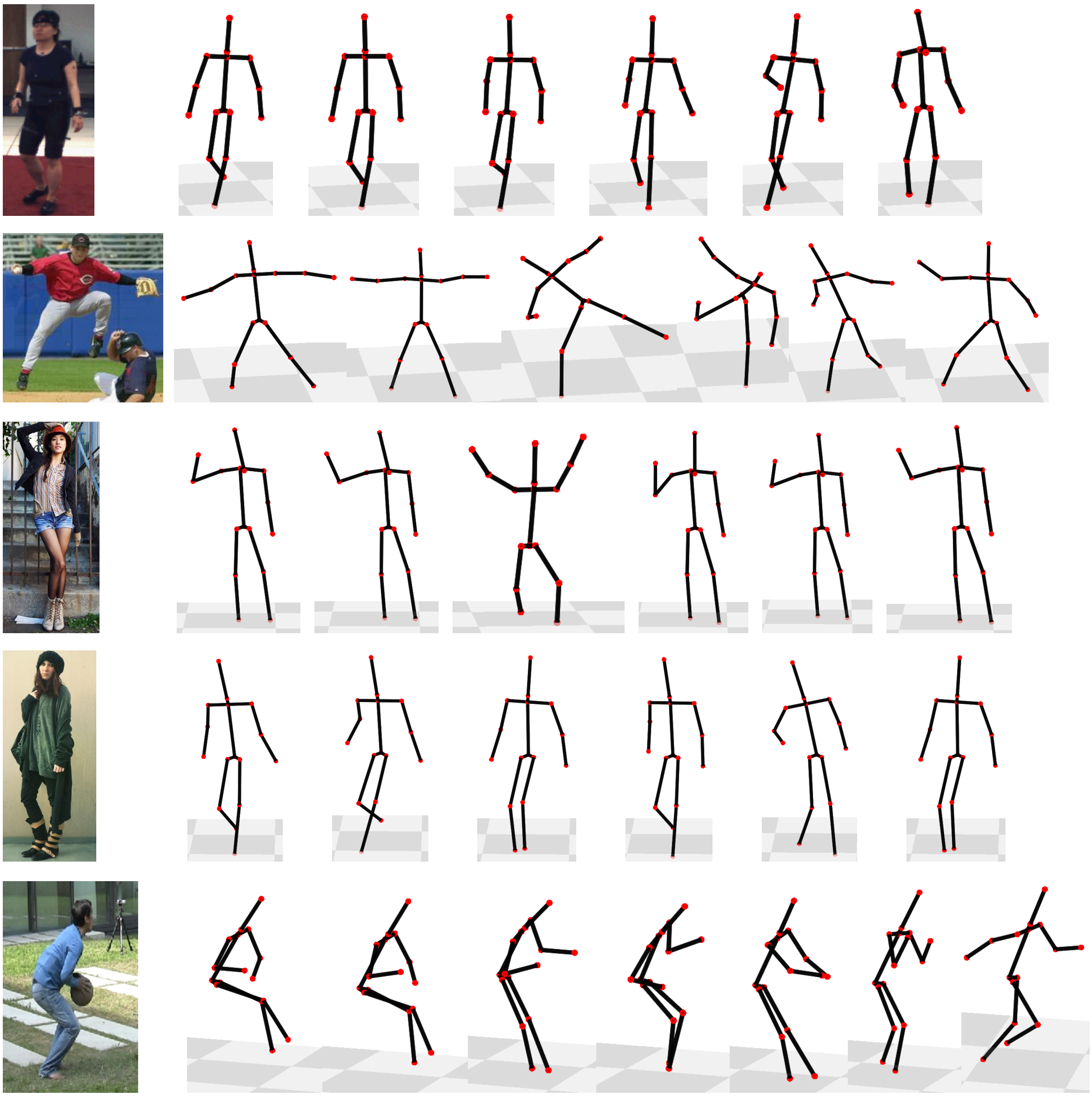
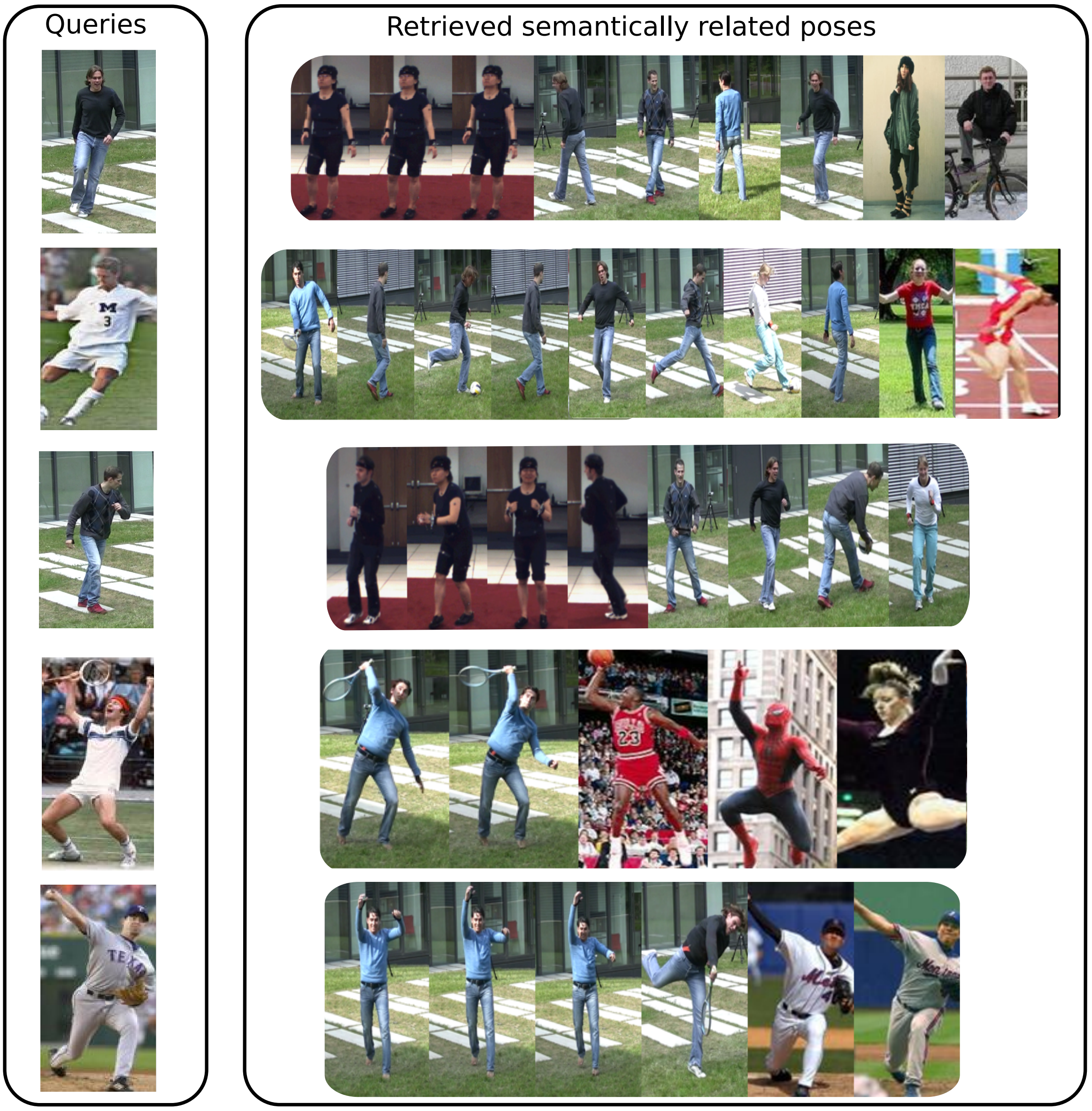


Figure 6.14: Examples poses drawn from our mixture model. Here we used the top 10 ranked posebits and 4 mixtures. Even when there are several miss-classifications we can retrieve meaningful poses thanks to our mixture model and to the fact that in many cases a few posebits already cluster the poses significantly. We note that in the fourth row most posebits are correctly classified and nonetheless the poses are not very similar to the image. That is because our MoCap database is still limited, we plan to add more poses from our own recordings, from CMU [151] and 3.6M [152] Database.



(a)

Figure 6.15: We can use the inferred posebits from the image to retrieve images in our database with similar posebit annotations. In particular, here we retrieve images with posebyte annotations that match any of the top 2 ranked posbytes given by our model. We show the query images on the left and the retrieved images on the right. Notice the sematic similarity in the images.

7 Conclusions

In this thesis we have addressed the problem of human pose estimation from video and IMUs in indoor controlled, outdoors semi-controlled and uncontrolled environments such as monocular images. The design of a human pose estimation system should be application dependent. The application will determine which *sensors* are available for the task and the required *level of detail of the pose*. In turn, this will condition the *inference* techniques to be used. Some applications such as character animation for movie production or motion analysis for medical applications require high degree of accuracy. Other applications such as scene understanding, or action recognition require only a coarse knowledge of the pose. In addition, certain applications allow to engineer the recording room. For movie production for example, asking the subject to wear a small set of inertial sensors does not suppose a major limitation. Similarly, multiple camera setups are realistic scenarios for such applications. However, if the pose estimation algorithm has to work on monocular images that one can find on the Internet, one can not assume for example orientation data coming from IMUs is available.

Different scenarios require different inference techniques. For example, if multiple cameras with high quality silhouettes and IMUs are available, one can achieve very good performance using local optimization. If however, the image cues are ambiguous and contain lots of background clutter, occlusions and illumination changes local optimization fails because it can not recover from errors during tracking. In such scenarios, inference using particle based optimization techniques can be more reliable. In particular, particle based optimization algorithms allow to propagate multiple pose hypotheses over time and therefore can potentially recover from tracking errors. When the data is even weaker, as in monocular imagery, one can not expect to recover the pose with a very high degree of accuracy. Indeed, although humans can resolve most of the depth ambiguities from monocular imagery, our reconstruction of pose is more qualitative than quantitative. In such scenarios, we aim at recovering a coarse semantic representation of the pose as opposed to highly accurate estimates.

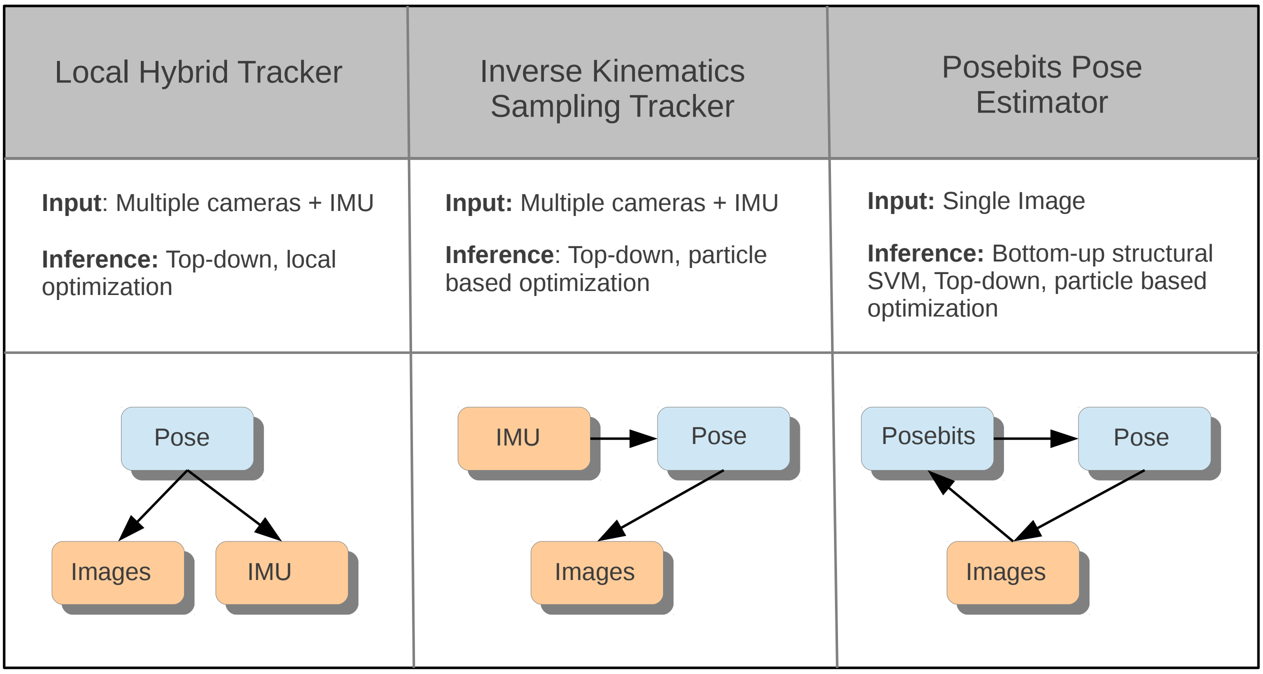
To summarize, to design a human pose estimation system, one should answer the following two application dependent questions:

- Which sensors are available ? To what extent does the application allow to engineer the recordings ?
- What level of detail is required or what is a realistic level of detail that can be inferred from the available observations ?

The first question will determine the inference method to be used, and the second one will determine the chosen representation of the pose. Bearing this in mind, we have presented three novel tracking systems at different levels of pose detail that operate in different scenarios. We summarize the significance of the proposed tracking systems below.

7.1 Local Optimization

We presented an approach for stabilizing full-body markerless human motion capturing using a small number of additional inertial sensors. Generally, the goal of reconstructing a 3D pose from 2D video data suffers from inherent ambiguities. We showed that a hybrid approach combining information of multiple sensor types can resolve such ambiguities, significantly improving the tracking quality. Information from the video input and the IMUs is integrated by jointly optimizing a combined energy.



Local Hybrid Tracker	Inverse Kinematics Sampling Tracker	Posebits Pose Estimator
Input: Multiple cameras + IMU **Inference:** Top-down, local optimization	**Input:** Multiple cameras + IMU **Inference**: Top-down, particle based optimization	**Input:** Single Image **Inference:** Bottom-up structural SVM, Top-down, particle based optimization

Figure 7.1: Summary table of the proposed pose estimation algorithms. On the first row we specify the input data and the main inference techniques used by our methods. On the bottom row we show the variables involved during inference: observed variables are shown in orange and hidden variables are shown in blue.

Since the scenarios we deal with here are indoor and the video data is relatively clean, we can use local optimization. We showed that the energy term corresponding to the orientation goals can be linearized in a similar way as the energy corresponding to positional goals. Since the kinematics are constrained by positional and orientation goals we could correct tracking errors arising from rotationally symmetric limbs. Using only a small number of inertial sensors fixed at outer extremities stabilized the tracking for the entire underlying kinematic chain. Several experiments for different motion patterns demonstrate that the proposed hybrid tracker results in much more accurate pose estimation results than a pure markerless motion capture system. In particular, our tracker can capture very accurately limb orientation which is very important for many applications. In addition, since the proposed tracker is based on local optimization it is very efficient making real time motion capture possible.

Application scenarios The proposed hybrid local tracker can be used for any application where a high degree of accuracy is required. An obvious example is for character animation where the recordings are usually made in indoor setups and a lot of accuracy is required. The proposed system is much more cost efficient than commercial based motion capture systems based on optical markers. At the same time it is much less intrusive which allows subjects to move more freely resulting in more natural motions.

7.2 Inverse Kinematics Sampling Tracker

The hybrid local tracker is very efficient and results in very accurate results in indoor multi-camera setups. Some applications however, require to make the MoCap recordings outdoors. This allows to capture motions in their natural environment, *e.g.* it allows to capture the movement a soccer player on the field itself where he can move freely. Motion capture outdoors is significantly more challenging as the video data term is ambiguous and high quality segmentation is not feasible. In addition, we considered scenarios with highly dynamic motions of people interacting with objects which creates many self-occlusions. Here we again argue that combining video data with a small number of sensors can greatly improve performance and enable tracking in challenging outdoor scenarios. The additional difficulties outdoors require a tracker that can recover from errors or at least that can propagate multiple hypothesis over time. This rules out the use of local optimization that provides a single estimate per frame and is prone to get trapped in local minima. A more robust alternative is to use particle based optimization. The main problem here is that the pose search space is high dimensional and a large number of particles are needed which makes it computationally very expensive. Furthermore, it is not clear how to successfully integrate the kinematic constraints imposed by the sensors in this framework. Therefore, we introduced a novel particle-based hybrid tracker that enables robust 3D pose estimation of arbitrary human motions in outdoor scenarios. We first presented an analytic procedure based on Inverse Kinematics for efficiently sampling from the manifold of poses that fulfill orientation constraints. Notably, we show how the IK can be solved in closed form by solving smaller Paden-Kahan subproblems. By sampling from the manifold, the dimension of the search space is reduced and the constraints are naturally full-filed. This, in turn, allows us to achieve good performance with far fewer particles. Secondly, robustness to uncertainties in the orientation data was achieved by introducing a sensor noise model based on the von Mises-Fisher distribution instead of the commonly used Gaussian distribution. Our experiments on diverse complex outdoor video sequences reveal major improvements in the stability and time performance compared to other state-of-the art trackers.

Application scenarios The proposed tracker can be used to perform motion capture of highly dynamic motions in outdoor scenarios. Again, the setup requires as few as 2 consumer cameras and a set of inertial sensors (we used five) [1]. This constitutes a cost efficient alternative to marker-based systems which do not work well outdoors and are much more expensive. Pure inertial systems exist but they typically drift over time. Capturing outdoors is important for applications such as sports science or movie production. Although in this work we focused on the integration of constraints derived from IMU, the proposed sampling scheme can be used to integrate general kinematic constraints. Kinematic constraints can be derived from the environment, *e.g.* one might want to impose that the feet touch the ground or that the hands are in contact with an object.

7.3 Posebits Pose Estimator

The two previous trackers achieve a high degree of accuracy which is needed for many applications. However, they are relatively intrusive, the subject needs to wear a set of IMUs and at least two cameras are required. Certain scenarios, require to be able to extract pose information from images into the wild. That is, images of people in natural photographs we can find on Internet archives. For this task, we usually have a single image. We argue here that since the problem is extremely ill-posed and under-constrained one should first extract high level qualitative information about the pose. To this end, we have introduced posebits which constitute a semantically powerful pose descriptor. Posebits describe relationships between parts in the body, *e.g.* a posebit determines whether the right hand is above the head. We show how to select a set of good posebits using a method based on decision trees. Given a set of selected posebits we infer them from image features using a discriminative structural SVM predictor. The inferred posebits serve to constrain the pose space, *i.e.*, we sample poses that satisfy the estimated posebits. We argue that this semantic pose prior facilitates pose estimation under difficult ambiguities present in monocular images.

One of the main advantages of the proposed method is that 3D pose-image pairs are no longer needed which enables easy collection of training data. This is due to the fact that labeling images with

---

[1] To achieve good results at least 4 cameras should be used

posebits is an easy task that humans can quickly perform. The experiments show that posebits can be estimated from images with good accuracy, even when training using different datasets. This results in improved pose estimation performance. Posebits can resolve many of the monocular ambiguities and can be useful as basis for many potential applications.

Applications The pose estimation method based on posebits is suited for monocular imagery where the data is very weak and many ambiguities are present. The proposed posebits however can be used as a useful pose feature which may constitute the input for many algorithms. They could for example be used as input for action recognition, scene understanding or image retrieval. Posebits are also useful to segment mocap data in semantic classes.

7.4 Future Work

7.4.1 Multiple Cues

In this thesis we have used relatively simple image cues such as edges, silhouettes and color histograms. Modeling texture, clothing, shading and illumination [10] should make the problem less under-constrained. Other sensors such as depth cameras, or the inexpensive Kinect sensor can make the problem significantly easier since they provide 3D measurements. Indeed, excellent performance has been achieved using such sensors [116, 51].

7.4.2 More Detail

Although human motion is well approximated by an articulated skeleton, in reality human motion is non rigid: muscle bulging occurs when we move, and the body fat motion is very non-rigid. To achieve a high degree of realism, the human shape can also be captured. There exist models to capture human shape from images. When multiple cameras are available one can resort to performance capture [85], where every triangle in the surface can deform independently under some smoothness constraints. The real challenge however is to capture fine shape deformations from a few or a single camera. In that case, lower dimensional models of human shape learned from 3D scans such as SCAPE [82] can be very useful. However, people's shape is usually covered by clothing [153, 154]. In order to accurately estimate shape one probably needs to estimate clothing. Actually, estimating clothing shape and material properties is itself of great practical importance for the Online Shopping industry. Here again, since clothing deformations are very complex, data-driven methods as in [155] will be required to constrain the problem.

7.4.3 Multiple People and the Environment

People appear in images interacting with the environment. We think that future algorithms will have to exploit the geometry of the scene to constrain the pose of the people. The objects in the scene, such as tables, chairs and also smaller objects such as cups and cellphones can be seen as nuisance because they produce occlusions. However, they can impose strong kinematic constraints on the pose, *e.g.*, there are not many ways one can sit on a chair. Constraints derived from the scene layout could be integrated using our Inverse Kinematics sampling scheme. We assumed in this thesis a single person to be tracked at a time. Tracking multiple people simultaneously could potentially facilitate the task since people interact in predetermined way, *e.g.*, hand shakes, hugs. Indeed, pose estimation algorithms can be coupled with multiple people tracking algorithms [55, 54] in a straightforward way.

7.4.4 Monocular Pose Estimation

The problem of monocular pose estimation is heavily under-constrained since there are multiple 3D poses that could explain a given image under the likelihood models we are currently using. Performance would improve with richer likelihood models, including more realistic models. However, putting too much effort modeling little details before not knowing anything about the scene might not be the most effective approach. For example, trying to infer the configuration of the fingers in the hands without even knowing where the hands are might not be the best way to approach it. Therefore, it is often useful to extract high level information about the scene. This high level information might be whether there is a person or more, whether the scene is in an office or in an outdoor nature environment or whether the person is sitting or standing. This sort of high level information can be extracted using discriminative methods that use training data to map image features to this labels. Our posebits tracker is a first step into this direction. Actually, it is likely that successful algorithms will have a bottom-up (discriminative) process to extract high level information and a top-down (generative) process to recover and explain the little details of the scene. Discriminative models are effective because they are very task specific but they suffer from poor generalization. Generative models on the other hand generalize better but are require much more modeling effort and inference is often intractable. Hence, a promising approach is a generative model conditioned on high level information latent variables inferred discriminatingly.

7.4.5 Putting the Pieces Together

To summarize, we believe pose estimation algorithms have to become more intelligent. It is not likely that computer algorithms will see a significant improvement if they do not integrate many sources of information together. Therefore, we think pose estimation algorithms should be one more piece of a computer vision system. Indeed, pose estimation can be leveraged by jointly inferring the objects, the scene layout, people interactions and the context. All this pieces of information together should reduce the ambiguities and make the monocular pose estimation problem more complex yet more feasible.

Glossary

2D Two dimensional 120

3D Three dimensional 120

belief propagation Distributed algorithm for inference on graphical models, which is optimal for tree-structured graphs. Variants include the sum-product algorithm and max-product algorithm 27, 120

body frame The body frame is a frame of reference fixed at the moving body 24, 120

discriminative Discriminative models typically model a conditional distribution of target outputs given a set of inputs. Discriminative models differ from generatives in that they do not allow one to generate samples from the joint distribution over inputs and outputs (and/or hidden variables). Discriminative models are particularly well suited for input-output tasks such as classification or regression. 5, 120

DP Dynamic Programming. A combinatorial algorithm for optimizing decomposable objective functions recursively 27, 120

EM Acronym for Expectation-Maximization 120

Expectation-Maximization In statistics, an expectation-maximization (EM) algorithm is a method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables 43, 120

forward kinematics Given an articulated body parameterized by joint angles $\Theta \in Q$, the forward kinematics is defined as the mapping from the vector of joint angles to the position and orientation of the body segment $\mathbf{G}_{sb} : Q \rightarrow SE(3)$ 24, 120

generative Generative models are models capable of generating (synthesizing) observable data. Generative models are able to model joint probability distributions over the input, output and hidden variables in the model. During inference generative models are often used as an intermediate step in forming conditional distribution of interest. Generative models, in contrast to discriminatives, provide a full probabilistic model over all variables, whereas a discriminative provides a model over the target output variable(s) conditioned on the input variables 5, 37, 120

gimbal lock Gimbal lock is the loss of one of the three DoF of a rotation parameterized by either Euler angles or three concatenated revolute joints. It occurs when two of the axis of rotation align and therefore one degree of freedom is lost 17, 18, 25, 120

GP A Gaussian Process is a continuous stochastic process defined on a real-valued domain (*e.g.*, time). It defines a Gaussian distribution over functions, and is fully characterized by a mean function and a covariance function. In addition any realization at a finite set of points in the domain (*e.g.*, time instants) form a multivariate Gaussian density 120

GPDM A Gaussian Process Dynamical Model is an extension of the GPLVM to handle high-dimensional time series data. In addition to the probabilistic generative mapping from latent positions to the observation in the GPLVM, it includes a dynamical model that models the temporal evolution of the data in terms of a latent dynamical model 49, 120

GPLVM A Gaussian Process Latent Variable Model is a probabilistic generative model that is learned from high-dimensional data. It can be used as a probabilistic dimensionality reduction, where the latent variables capture the structure (latent causes) of the high-dimensional training data. It is a generalization of probabilistic PCA to nonlinear mappings 49, 120

Hessian matrix Matrix of second-order partial derivatives of a function of several variables. It describes the local curvature of a function. Suppose a function $f : \mathbb{R}^n \mapsto \mathbb{R}$, $f(x_1,x_2,\ldots,x_n)$ then the Hessian matrix is :

$$\mathbf{H}_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n\,\partial x_1} & \frac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

48, 120

HOG Histograms of Oriented Gradients. Common gradient-based descriptor for an image. 33, 120

homogeneous coordinates Homogeneous coordinates are a system of coordinates used in projective geometry with applications in computer vision and computer graphics. Rigid transformation of points and vectors have a simpler representation in those coordinates than in the typical Euclidean coordinates. Let us define the following notation:
Homogeneous coordinates of a point $\mathbf{p} \in \mathbb{R}^3$ are $\bar{\mathbf{p}} = [\mathbf{p}\ 1]$.
Homogeneous coordinates of a vector $\mathbf{v} \in \mathbb{R}^3$ are $\bar{\mathbf{v}} = [\mathbf{v}\ 0]$.
Homogeneous coordinates of a rigid motion $\mathbf{g} = (\mathbf{R},\mathbf{t})$ are $\mathbf{G} = \begin{bmatrix} \mathbf{R}_{[3\times3]} & \mathbf{t}_{[3\times1]} \\ \mathbf{0}_{[1\times3]} & 1 \end{bmatrix}$ Thereby, a point is transformed by g with simple matrix multiplication $\bar{\mathbf{p}}_1 = \mathbf{G}\bar{\mathbf{p}}_0$ 16, 120

human pose Relative orientation and position of the human body limbs w.r.t some reference coordinate system 15, 120

ICP Iterative Closest Point is an algorithm to align or register two sets of points. The algorithm iterates the following steps: (i) collect correspondences from the two sets of points by the nearest neighbor criterion, (ii) estimate the transformation between both sets. (iii) transform the points with the computed transformation. This procedure is iterated until congergence 41, 120

image contour Image contour features are edge features that are on the outer extremity of the object. In other words, it is a sub-set of edge features on the outline of the object (as opposed to internal edge features) 38, 120

image edge Image edges are defined as pixels in the image where there exists a discontinuities in the pixel brightness. Image edges are common features used in vision as they are easy to compute and are largely invariant to lighting 38, 120

image likelihood Given an observed image (or images), I, image likelihood, $p(I|x)$, measures the probability of that image(s) being observed given a set of input parameters x 30, 43, 120

image silhouette Image silhouette corresponds to a binary image where pixels that are 0 are assumed to be part of the background and pixels that have value of 1 are part of the foreground object of interest. Binary image features are most often obtained through the process of background subtraction 38, 120

IMU Inertial Measurement Unit. Device to measure local orientation and acceleration of a moving body. 53, 120

inverse kinematics Inverse kinematics is the process of determining the parameters (joint angles) of a parameterized articulated object in order to achieve a desired goal. Goals are configurations of position and/or orientation of typically end-effector segments (such as the hands) 71, 120

Jacobian matrix Matrix of all first order derivatives of a vector or scalar-valued function with respect to another vector. Suppose $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ given by $y_1(x_1, \ldots, x_n), \ldots, y_m(x_1, \ldots, x_n)$, then the Jacobian matrix is:

$$\mathbf{J_F} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

39, 40, 120

Kalman filter Kalman filter is an algorithm for efficiently doing exact inference in a linear dynamical system (LDS), where all latent and observed variables have a Gaussian (or multivariate Gaussian) distribution 43, 120

kinematic chain Parameterization commonly used to represent the motion of articulated figures. The orientation and position of an end-efector segment is determined by the orientation and position of the previous segment in the chain and an angular rotation about a joint axis 15, 120

kinematic singularity In the context of human pose estimation a kinematic singularity refers to the fact that multiple configurations of joint angles result in the same human pose 43, 120

LDS A Linear Dynamical System is used to refer to a linear-Gaussian Markov process. In such a process the state evolution is modeled as a linear transformation plus Gaussian process noise. A first-order LDS on state $\mathbf{x}$, for matrix $\mathbf{A}$, is given by $\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \eta$ where η is a Gaussian random variable that is independent of $\mathbf{x}$ and IID through time 49, 120

MAP Acronym for maximum a posteriori estimate 37, 120

Markov Chain Monte Carlo A general framework for generating samples from a graphical model, of which is a popular special case 43, 120

Markov process A Markov process (or Markov chain) is a time-varying stochastic process that satisfies the Markov property. An n^{th}-order Markov process, $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots)$, satisfies $p(\mathbf{x}_t \,|\, \mathbf{x}_1, \cdots, \mathbf{x}_{t-1}) = p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \cdots, \mathbf{x}_{t-n})$. That is, conditioned on the previous n states, the current state is independent of all other previous states 43, 120

maximum a posteriori In Bayesian statistics, a maximum a posteriori probability (MAP) estimate is defined as a mode of the posterior distribution. 120

maximum likelihood Maximum likelihood estimation is a method for estimating parameters of a statistical model. For a fixed set of data and underlying statistical model, the method of maximum likelihood selects values of the model parameters that produce a distribution that gives the observed data the greatest probability (*i.e.*, parameters that maximize the image likelihood function). 120

mixture of Gaussian A method to approximate a distribution as a sum of Gaussian distributions. 32, 120

ML Acronym for maximum likelihood 120

MoCap Acronym for motion capture 120

motion capture The process of recording movement and translating that movement to a digital model. MoCap technology has numerous applications in entertainment, sports and medical applications 120

optical flow Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene 35, 41, 120

particle filter The particle filters, also known as sequential Monte Carlo methods (SMC), approximate the posterior with a set of typically weighted samples 43, 120

PCA Principal Component Analysis is a method for dimensionality reduction, wherein high-dimensional data are projected onto a linear subspace with an orthogonal matrix. It can be formulated as the orthogonal linear mapping that maximizes the variance of projection in the subspace. Probabilistic PCA is a closely related latent variable model that specifies a linear-Gaussian generative process 30, 49, 120

pep, pixel picture element, image point 120

posterior Posterior probability of a random event is the conditional probability once all the relavent evidence is taken into account. According to Bayesian statistical theory posterior can be exprezssed as a product of the the prior and likelihood, *i.e.*, $p(x|I) \propto p(I|x)p(x)$ 43, 120

prior A prior probability measures the likelihood of an event before any observable data evidence is taken into account 37, 120

QP Quadratic Program. This is a convex optimization problem for which large-scale solvers exist. A support-vector machine SVM can be trained with a quadratic program. 48, 120

rigging In computer graphics, given mesh representation of an articulated body rigging is the process of inserting a skeletton with a bone hierarchy and skinning the mesh. Rigging is the standard way to animate characters or mechanical objects 29, 120

rigid body motion refers to the motion of a set of particles that move rigidly together *i.e.*, the distance between any two points in the rigid body is constant over time. Therefore, the position of a rigid body is completely determined by the position of a reference point in the body and its orientation 16, 120

SGD Stochastic Gradient Descent. This is an online algorithm for optimizing objective functions defined on large training sets. 120

SIFT Scale-Invariant Feature Transform. Common method for generating a sparse set of invariant keypoints and descriptors from an image 35, 38, 41, 120

singularity In mathematics, a singularity is in general a point at which a given mathematical object is not defined, or a point where it fails to be well-behaved in some particular way. For a given function examples of singularities are points where the function is not defined or not differentiable 17, 120

skinning In computer graphics, given a surface mesh of the body and a skeletton, skinning is the process of binding skin to the skeleton bones. Every vertex in the mesh has to be assigned to one of the bones. It can be thought of as a segmentation of an articulated body in a number of rigid segments 29, 120

SLDS A Switching Linear Dynamical System is a collection of N LDS models along with a discrete switching variable, $s \in \{1, \cdots, N\}$. The switching variable identifies which LDS should be active at each time step. As a probabilistic generative model, each LDS is a linear-Gaussian model, and on maintains a multinomial distribution for s. SLDS models are used to approximate nonlinear dynamical processes in terms of piecewise linear state evolution 49, 120

spatial frame The spatial frame is an inertial frame of reference which is fixed to the enviroment 24, 120

statistical independence Two events are said to be statistically independent if occurrence of one event does not make the other event more or less probable 120

sum product Distributed algorithm for inference on graphical models, which is optimal for tree-structured graphs. 27, 120

SVM Support Vector Machine. This is a linear classifier that is trained using a max-margin objective function. Extensions include nonlinear classifiers trained with kernel mappings, and “soft”-margin constraints using a hinge-loss objective. 8, 120

wedge operator The wedge operator ($\wedge$) for a point or vector $\omega \in \mathbb{R}^3$ is defined as $(\omega)^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$. Thorough the book, we will often use $\widehat{\omega}$ as a replacement for $\omega^\wedge$. The notation $\omega_{[\times]}$ is also commonly used in the computer vision literature to dennote the wedge operator. Analogously the operator for the full *twist coordinates* $\xi \in \mathbb{R}^6$ is defined as $\widehat{\xi} = \begin{bmatrix} v \\ \omega \end{bmatrix}^\wedge = \begin{bmatrix} \widehat{\omega} & v \\ \vec{0} & 0 \end{bmatrix}$ and is used to construct the *twist action* $\widehat{\xi} \in se(3)$ 18, 20, 120

Index

adjoint transformation, 21
annealed particle filter (APF), 45
axis angle, 18

body frame, 24

camera model, 39, 51
correspondence, 38, 59

Euler angles, 17
expectation-maximization (EM), 43
exponential map, 19

Gauss-Newton optimization, 39, 40
gimbal lock, **17**

homogeneous coordinates, 16

image contour, 33, 59
image edge, 33
image features, 35
image likelihood, 31, 59, 82, 102
image silhouette, 82
importance sampling, 43
IMU, 36, 53
inverse kinematics, 76
iterative closest point (ICP), 41

joint
  ball (3D), **25**
  revolute (3D), **25**
  root (6D), **25**, 25
joint limits, 48

kinematic chain, 15, **24**

Levenberg-Marquadt optimization, 40
likelihood, 30, 36, 61, 75
linear dynamical systems (LDS), 49

Markov process, 43

non-linear least squares, 39

optical flow, **35**, 37, **41**

particle filter (PF), **43**, 43
pose Jacobian, 26, 38
pose parameters, 25
principal component analysis (PCA), 30

quaterion, **17**
quaternion product, 17

rigging, 29, 30
rigid body motion, 16
Rodriguez formula, **19**
rotation matrix, 15

simulated annealing, 45
skinning, 29
spatial frame, 24
switching linear dynamical systems (SLDS), 49

twist action, 20
twist coordinates, 20

wedge operator, **25**

Bibliography

[1] T.B. Moeslund and E. Granum. A survey of computer vision based human motion capture. *Computer Vision and Image Understanding (CVIU)*, 81(3), 2001.

[2] G. Pons-Moll and B. Rosenhahn. Model-based pose estimation. In *Visual Analysis of Humans: Looking at People*, pages 139–170. Springer, 2011.

[3] David J Fleet. Motion models for people tracking. In *Visual Analysis of Humans: Looking at People*, pages 171–198. Springer, 2011.

[4] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 126–133, 2000.

[5] J. Gall, B. Rosenhahn, T. Brox, and H. Seidel. Optimization and filtering for human motion capture. *International Journal on Computer Vision (IJCV)*, 87:75–92, 2010.

[6] M. Vondrak, L. Sigal, and O. C. Jenkins. Physical simulation for probabilistic motion tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[7] C. Sminchisescu. Consistency and coupling in human model likelihoods. In *IEEE Conference on Automatic Face and Gesture Recognition*, pages 22–27, 2002.

[8] G. Pons-Moll, L. Leal-Taixe, T. Truong, and B. Rosenhahn. Efficient and robust shape matching for model based human motion capture. In *German Conference on Pattern Recognition (GCPR)*, September 2011.

[9] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker. Detailed human shape and pose from images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[10] 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)*, pages 1381–1388, 2009.

[11] C. Wu, K. Varanasi, Y. Liu, H-P Seidel, and C. Theobalt. Shading-based dynamic shape refinement from multi-view video under general illumination. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1108–1115, 2011.

[12] H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision (ECCV)*, volume 1843 of *Lecture Notes in Computer Science*, pages 702–718. Springer Berlin / Heidelberg, 2000.

[13] C. Stoll, N. Hasler, J. Gall, H-P. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *IEEE International Conference on Computer Vision (ICCV)*, pages 951–958. IEEE, 2011.

[14] C.S. Lee and A. Elgammal. Coupled visual and kinematic manifold models for tracking. *International Journal on Computer Vision (IJCV)*, 87(1-2):118–139, 2010.

[15] M. Salzmann and R. Urtasun. Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 647–654, 2010.

[16] R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with gaussian process dynamical models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[17] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):283–298, 2008.

[18] D.J. Brubaker M. A., Fleet and A. Hertzmann. Physics-based person tracking using the anthropomorphic walker. In *International Journal on Computer Vision (IJCV)*, volume 87, pages 140–155, 2010.

[19] A. Baak, B. Rosenhahn, M. Müller, and H.P. Seidel. Stabilizing motion tracking using retrieved motion priors. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1428–1435, 2009.

[20] J. Chen, M. Kim, Y. Wang, and Q. Ji. Switching gaussian process dynamic models for simultaneous composite motion tracking and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2655–2662. IEEE, 2009.

[21] G. W. Taylor, L. Sigal, D. J. Fleet, and G. E. Hinton. Dynamical binary latent variable models for 3d human pose tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 631–638, 2010.

[22] J. Gall, A. Yao, and L. Van Gool. 2D action recognition serves 3D human pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 425–438, 2010.

[23] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3d human tracking. In *IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2003.

[24] M. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2004.

[25] G. Pons-Moll, A. Baak, Gall J., L. Leal-Taixé, M. Mueller, H-P. Seidel, and Bodo Rosenhahn. Outdoor human motion capture using inverse kinematics and von mises-fisher sampling. In *IEEE International Conference on Computer Vision (ICCV)*, nov 2011.

[26] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 677–684, 2000.

[27] C. Sminchisescu and A. Jepson. Variational mixture smoothing for non-linear dynamical systems. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–608. IEEE, 2004.

[28] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *IEEE International Conference on Computer Vision (ICCV)*, pages 750–757, 2003.

[29] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1052–1062, 2006.

[30] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(1):44–58, 2006.

[31] C. Sminchisescu, A. Kanaujia, and D. N. Metaxas. Bm3e : Discriminative density propagation for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(11):2030–2044, 2007.

[32] L. Bo and C. Sminchisescu. Twin Gaussian Processes for Structured Prediction. *International Journal of Computer Vision (IJCV)*, 87:28–52, 2010.

[33] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008.

[34] Roland Memisevic, Leonid Sigal, and David J. Fleet. Shared kernel information embedding for discriminative inference. *IEEE Transactions on Pattern Analysis and Machine Vision (TPAMI)*, 34(4):778–790, 2012.

[35] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304, 2011.

[36] H. Dejnabadi, B.M. Jolles, E. Casanova, P. Fua, and K. Aminian. Estimation and visualization of sagittal kinematics of lower limbsorientation using body-fixed sensors. *TBME*, 53(7):1382–1393, 2006.

[37] Y. Tao, H. Hu, and H. Zhou. Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation. *International Journal on Robotics Research (IJRR)*, 26(6):607, 2007.

[38] R. Slyper and J. Hodgins. Action capture with accelerometers. In *ACM SIGGRAPH/Eurographics, SCA*, 2008.

[39] D. Roetenberg, H. Luinge, and P. Slycke. Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors.

[40] C.H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 951–958. IEEE, 2009.

[41] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. *European Conference on Computer Vision*, pages 155–168, 2010.

[42] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3337–3344. IEEE, 2011.

[43] A. Yao, J. Gall, G. Fanelli, and Van Gool L. Does human action recognition benefit from pose estimation? In *British Machine Vision Conference (BMVC)*, 2011.

[44] M. Müller, T. Röder, and M. Clausen. Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphics (TOG)*, 24(3):677–685, 2005.

[45] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1365–1372, 2009.

[46] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1014–1021, 2009.

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

[48] Yi Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1385–1392, 2011.

[49] G. Pons-Moll, A. Baak, T. Helten, M. Müller, H.-P. Seidel, and B. Rosenhahn. Multisensor-fusion for 3D full-body human motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 663–670, 2010.

[50] G. Pons-Moll, D. J. Fleet, and B. Rosenhahn. Posebits for monocular human pose estimation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, USA, June 2014.

[51] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon. Metric regression forests for human pose estimation. In *British Machine Vision Conference (BMVC)*, 2013.

[52] G. Pons-Moll and B. Rosenhahn. Ball joints for marker-less human motion capture. In *Workshop on applications on Computer Vision (WACV)*, pages 1–8, 2009.

[53] G. Pons-Moll, L. Leal-Taixé, J. Gall, and B. Rosenhahn. Data-driven manifolds for outdoor motion capture. In *Outdoor and Large-Scale Real-World Scene Analysis*, volume 7474 of *LNCS*, pages 305–328. Springer, 2012.

[54] G. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. Branch-and-price global optimization for multi-view multi-object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.

[55] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR)*, pages 120–127, 2011.

[56] L. Leal-Taixé, G. Pons-Moll, and Bodo Rosenhahn. Exploiting pedestrian interaction via global optimization and social behaviors. In *Theoretic Foundations of Computer Vision: Outdoor and Large-Scale Real-World Scene Analysis*. Springer, April 2012.

[57] A. Kuznetsova, G. Pons-Moll, and B. Rosenhahn. Pca-enhanced stochastic optimization methods. In *Pattern Recognition, German Conference on Pattern Recognition (GCPR)*, pages 377–386, 2012.

[58] A. Baak, T. Helten, M. Müller, G. Pons-Moll, B. Rosenhahn, and H.P. Seidel. Analyzing and evaluating markerless motion tracking using inertial sensors. In *Proceedings of the 3rd International Workshop on Human Motion. In Conjunction with ECCV*, volume 6553 of *Lecture Notes of Computer Science (LNCS)*, pages 137–150. Springer, 2010.

[59] G. Pons-Moll, Gilead Tadmor, Rob S. MacLeod, B. Rosenhahn, and D. H. Brooks. 4d cardiac segmentation of the epicardium and left ventricle. In *World Congress of Medical Physics and Biomedical Engineering (WC)*, 2009.

[60] G. Pons-Moll, G. Crosas, G. Tadmor, R. MacLeod, B. Rosenhahn, and D. Brooks. Parametric modeling of the beating heart with respiratory motion extracted from magnetic resonance images. In *IEEE Computers in Cardiology (CINC)*, September 2009.

[61] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Baton Rouge, 1994.

[62] K. Shoemake. Animating rotation with quaternion curves. *ACM Transactions on Graphics (SIGGRAPH)*, 19(3):245–254, 1985.

[63] S. Grassia. Practical parameterization of rotations using the exponential map. *Journal on Graphics Tools (JGT)*, 3:29–48, 1998.

[64] D. Ramanan. Finding and tracking people in images/videos. In *Visual Analysis of Humans: Looking at People*. Springer, 2011.

[65] Deva Ramanan, David A Forsyth, and Andrew Zisserman. Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(1):65–81, 2007.

[66] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2041–2048, 2006.

[67] R. Kehl, M. Bray, and L. Van Gool. Full body tracking from multiple views using stochastic sampling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 129–136, 2005.

[68] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2001.

[69] R. Plankers and P. Fua. Articulated soft objects for multiview shape and motion capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(9):1182–1187, 2003.

[70] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: a parameterized model of articulated image motion. In *IEEE conference on Automatic Face and Gesture Recognition*, pages 38–44, 1996.

[71] R. Plankers and P. Fua. Articulated soft objects for video-based body modeling. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 394–401, 2001.

[72] B. Allen, B. Curless, and Z. Popović. Articulated body deformation from range scan data. In *ACM Transactions on Graphics (SIGGRAPH)*, pages 612–619, New York, NY, USA, 2002. ACM.

[73] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. In *ACM Transaction on Graphics (TOG)*, pages 399–405, 2004.

[74] D. Anguelov, P. Srinivasan, H.C. Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in neural information processing systems (NIPS)*, page 33. The MIT Press, 2005.

[75] Ganapathi V., Plagemann C., Thrun S., and D. Koller. Real time motion capture using a time-of-flight camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[76] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision (IJCV)*, 56(3):179–194, 2004.

[77] B. Rosenhahn and T. Brox. Scaled motion dynamics for markerless motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[78] J. P. Lewis, Matt Cordner, and N. Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[79] I. Baran and J. Popović. Automatic rigging and animation of 3d characters. In *ACM Transactions on Graphics (SIGGRAPH)*, page 72, 2007.

[80] N. Hasler, H. Ackermann, B. Rosenhahn, T. Thormaehlen, and H. Seidel. Multilinear pose and body shape estimation of dressed subjects from image sets. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1823–1830, 2010.

[81] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM Transactions on Graphics (SIGGRAPH)*, pages 587–594, New York, NY, USA, 2003. ACM.

[82] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Transactions on Graphics (SIGGRAPH)*, 24:408–416, 2005.

[83] D. A Hirshberg, M. Loper, E. Rachlin, and M.J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *European Conference on Computer Vision (ECCV)*, pages 242–255. Springer, 2012.

[84] C. Cagniart, E. Boyer, and S. Ilic. Free-form mesh tracking: A patch-based approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1339–1346, 2010.

[85] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *ACM Transactions on Graphics (SIGGRAPH)*, pages 1–10, New York, NY, USA, 2008. ACM.

[86] B. Rosenhahn, C. Schmaltz, T. Brox, J. Weickert, D. Cremers, and H.-P. Seidel. Markerless motion capture of man-machine interaction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[87] S. Dambreville, R. Sandhu, A. Yezzi, and A. Tannenbaum. Robust 3d pose estimation and efficient 2d region-based segmentation from a 3d shape prior. In *European Conference on Computer Vision (ECCV)*, volume 5303 of *Lecture Notes in Computer Science*, pages 169–182. Springer Berlin / Heidelberg, 2008.

[88] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, and T. P Andriacchi. Markerless motion capture through visual hull, articulated icp and subject specific model generation. *International Journal of Computer Vision (IJCV)*, 87(1-2):156–169, 2010.

[89] Marcus A Brubaker, Leonid Sigal, and David J Fleet. Video-based people tracking. In *Handbook of Ambient Intelligence and Smart Environments*, pages 57–87. Springer, 2010.

[90] M. Piccardi. Background subtraction techniques: a review. In *Proc. IEEE Int Systems, Man and Cybernetics Conf*, volume 4, pages 3099–3104, 2004.

[91] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision (IJCV)*, 60(2):91–110, 2004.

[92] J. Gall, B. Rosenhahn, and H.-P. Seidel. Drift-free tracking of rigid and articulated objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[93] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *FTCGV*, 1(1):1–89, 2005.

[94] Glenn Shafer. *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976.

[95] B. Scheuermann and B. Rosenhahn. Feature quarrels: The dempster-shafer evidence theory for image segmentation using a variational framework. In *Asian Conference on Computer Vision (ACCV), Part II, LNCS*, volume 6493, pages 426–439, 2010.

[96] B. Rosenhahn, T. Brox, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose tracking. *International Journal of Computer Vision*, 73(3):243–262, 2007.

[97] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.

[98] P. Besl and N. McKay. A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 12:239–256, 1992.

[99] Z. Zhang. Iterative points matching for registration of free form curves and surfaces. *International Journal on Computer Vision (IJCV)*, 13(2):119–152, 1994.

[100] KMG Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *IEEE Conferenfce on Computer Vision and Pattern Recognition*, volume 1, pages I–77, 2003.

[101] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Vision*, 16(2):150–162, 1994.

[102] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers. Combined region and motion-based 3d tracking of rigid and articulated objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(3):402–415, 2010.

[103] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 3, pages 674–679, 1981.

[104] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8–15, 1998.

[105] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11):1222–1239, 2001.

[106] T. F. Chan and L. A. Vese. Active contours without edges. *Transactions on Image Processing (TIP)*, 10(2):266–277, 2001.

[107] C. Schmaltz, B. Rosenhahn, T. Brox, D. Cremers, J. Weickert, L. Wietzke, and G. Sommer. Region-based pose tracking. In *Proc. 3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 0, pages 56–63, 2007.

[108] John Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339, 1989.

[109] L. Sigal, A.O. Balan, and M.J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal on Computer Vision (IJCV)*, 87(1):4–27, 2010.

[110] S. Kirkpatrick, D.G. Jr., and M.P. Vecchi. Optimization by simmulated annealing. *Journal of Statistical Physics*, 220(4598):671–680, 1983.

[111] J. Gall, B. Rosenhahn, and H.-P. Seidel. Clustered stochastic optimization for object recognition and pose estimation. In *German Conference on Pattern Recognition (GCPR)*, volume 4713, 2007.

[112] J. Gall, J. Potthoff, C. Schnorr, B. Rosenhahn, and H. Seidel. Interacting and annealing particle filters: Mathematics anda recipe for applications. *Journal on Mathematical Image Vision (JMIV)*, 28:1–18, 2007.

[113] Kiam Choo and D. J. Fleet. People tracking using hybrid monte carlo filtering. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 321–328, 2001.

[114] D. Gavrila and L. Davis. 3D model based tracking of humans in action: a multiview approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 73–80, 1996.

[115] J. Deutscher, A. Davison, and I. Reid. Automatic partitioning of high dimensional search spaces associated with articulated body motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2001.

[116] J Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 103–110. IEEE, 2012.

[117] B. North, A. Blake, M. Isard, and Jens R. Learning and classification of complex dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence Transactions*, 22(9):1016–1034, 2000.

[118] N. Hasler, B. Rosenhahn, T. Thormaehlen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 224–231, 2009.

[119] Xsens Motion Technologies. http://www.xsens.com/.

[120] L. Herda, R. Urtasun, and P. Fua. Implicit surface joint limits to constrain video-based motion capture. In *Euroepan Conference on Computer Vision (ECCV)*, volume 3022, pages 405–418, 2004.

[121] S. Ioffe and D. Forsyth. Human tracking with mixtures of trees. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 690–695, 2001.

[122] A. Elgammal and C.S. Lee. Inferring 3D body pose from silhouettes using activity manifoldlearning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2004.

[123] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3D human motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 390, 2005.

[124] L. Sigal, M. Vondrak, and O.C. Jenkins. Physical simulation for probabilistic motion tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[125] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.P. Seidel. Motion capture using joint skeleton tracking and surface estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[126] D. Roetenberg. Inertial and magnetic sensing of human motion. *These de doctorat*, 2006.

[127] Multimodal human motion database MPI08. http://www.tnt.uni-hannover.de/project/MPI08_-Database/.

[128] Thomas Moeslund, Adrien Hilton, Volker Krueger, and Leonid Sigal, editors. *Visual Analysis of Humans: Looking at People*. Springer, 2011.

[129] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal on Computer Vision (IJCV)*, 61(2):185–205, 2005.

[130] Leonid Sigal, Luca Balan, and Michael Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Conference on Neural Information Processing (NIPS)*, pages 1337–1344, 2008.

[131] R. Fisher. Dispersion on a sphere. *Proceedings of the Royal Society of London. Mathematical and Physical Sciences*, 1953.

[132] Human motion outdoor database HMODB. http://www.tnt.uni-hannover.de/papers/view_paper.php?id=901.

[133] W. Ping and J.M. Rehg. A modular approach to the analysis and evaluation of particle filters for figure tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[134] P. Azad, T. Asfour, and R. Dillmann. Robust real-time stereo-based markerless human motion capture. In *Proc. 8th IEEE-RAS Int. Conf. Humanoid Robots*, 2008.

[135] S. Hauberg, J. Lapuyade, M. Engell-Norregard, K. Erleben, and K. Steenstrup Pedersen. Three dimensional monocular human motion analysis in end-effector space. In *EMMCVPR*, 2009.

[136] H. Kjellstromm, D. Kragic, and M. J. Black. Tracking people interacting with objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 747–754, 2010.

[137] N.H. Lehment, D. Arsic, M. Kaiser, and G. Rigoll. Automated pose estimation in 3D point clouds applying annealing particle filters and inverse kinematics on a gpu. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2010.

[138] M. Fontmarty, F. Lerasle, and P. Danes. Data fusion within a modified annealed particle filter dedicated to human motion capture. In *IRS*, 2007.

[139] Fan Zhang, Edwin R. Hancock, Casey Goodlett, and Guido Gerig. Probabilistic white matter fiber tracking using particle filtering and von mises-fisher sampling. *Medical Image Analysis*, 13(1):5–18, 2009.

[140] B.E. Paden. *Kinematics and control of robot manipulators*. PhD thesis, 1985.

[141] R. Hartley and A. Zisserman. *Multiple view geometry*, volume 642. Cambridge university press Cambridge, UK, 2003.

[142] A. Wood. Simulation of the von mises-fisher distribution. *Communications in Statistics - Simulation and Computation*, 23(1):157–164, 1994.

[143] D. M. Gavrila and L. S. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 73–80, 1996.

[144] A. Kanaujia, C. Sminchisescu, and D. Metaxas. Semi-Supervised Hierarchical Models for 3D Human Pose Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[145] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *European Conference on Computer Vision (ECCV)*, pages 168–181. Springer, 2010.

[146] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6(2):1453, 2006.

[147] S. Lazebnik, Cordelia S., and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[148] Amazon mechanical turk. https://www.mturk.com/mturk/.

[149] Martin de La Gorce, David J Fleet, and Nikos Paragios. Model-based 3d hand pose estimation from monocular video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1793–1805, 2011.

[150] Kota Yamaguchi, Hadi Kiapour, and Luis E. Ortiz andTamara L. Berg. Parsing clothing in fashion photographs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[151] Cmu motion catpure database. http://mocap.cs.cmu.edu/.

[152] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. Technical report, Institute of Mathematics of the Romanian Academy and University of Bonn, September 2012.

[153] A. O Ballan and M. J Black. The naked truth: Estimating body shape under clothing. In *Euroepan Conference on Computer Vision (ECCV)*, pages 15–29. Springer, 2008.

[154] P. Guan, O. Freifeld, and M.J. Black. A 2d human body model dressed in eigen clothing. In *European Conference on Computer Vision (ECCV)*, pages 285–298. Springer, 2010.

[155] P. Guan, L. Reiss, D. A Hirshberg, A. Weiss, and M.J. Black. Drape: Dressing any person. *ACM Transactions on Graphics (TOG)*, 31(4):35, 2012.