Grenoble INP – ENSIMAG
École Nationale Supérieure d'Informatique et de Mathématiques Appliquées

Report of the Masters Research Project

Effectué chez Thales

Monocular Human 3D Pose Estimation

FÜHR Gustavo
M2R – Option GVR

14 février 2011 – 12 août 2011

Thales Services SAS
1 Rue du Général de Gaulle
BP 226
95523 Cergy-Pointoise Cedex

Responsables de stage
COUVET Serge
BOUFARGUINE Mourad
Tuteur de l'école
FRANCO Jean-Sébastien

Contents

Abstract

This report addresses the problem of recovering 3D human motion from a single image sequence, using model-based approaches. We review the current literature on the subject and propose improvements within an Annealed Particle Filtering framework.

A likelihood function that combines silhouette and visual appearance information is presented. Our appearance model is based on 2D color histograms computed in CIELab color space. To avoid pixel mis-attributions caused by self-occlusions, we build visibility maps that help to correctly sample the images.

A temporal prior specific for walking motions is also proposed to limit the search space and introduce temporal consistency. It is based on principal component analysis of a set of walking cycles that are extracted from motion capture data. The particle states are reformulated to include a low-dimensional point representing a walking cycle, the walking phase in this cycle and a parameter associated to the walking speed.

Finally, a set of experiments using the HumanEva-I dataset is presented. Tracking results using the proposed likelihood function show that mixing image features can increase the estimation accuracy. Additionally, the results are further improved when the PCA-based temporal prior is included: more accurate poses are recovered using a particle filter with almost 7 times fewer particles (compared to a filter with a prior based on a Gaussian random walk).

Keywords: human motion tracking, annealed particle filter, PCA-based prior, appearance model

Résumé

Ce rapport traite de l'estimation de pose 3D d'une personne observée par une caméra monoculaire. Nous effectuons un état de l'art concernant le sujet et nous proposons des améliorations dans une nouvelle méthode de suivi basée sur un filtrage particulaire.

Une fonction de vraisemblance qui mélange des informations de silhouette et apparence est présentée. Le modèle d'apparance utilisé est basée sur des histogrammes de couleur. Les occultations sont traitées avec des cartes de visibilité.

Une distribution de transition est aussi proposée pour limiter l'espace de recherche et introduire une cohérence temporelle. Elle est basée sur l'analyse en composantes principales d'un ensemble des cycles de marche obtenu à partir de données mocap. Le vecteur d'état du filtre particulaire contient un point dans l'espace réduit qui correspond à un cycle de marche, la phase de marche et un paramètre associé à la vitesse de marche.

Finalement plusieurs expériences, sur la base HumanEva, sont présentées. Des résultats du suivi qui utilise la fonction de vraisemblance proposée montre que le mélange de descripteurs peut augmenter la précision de l'estimation. De plus, les résultats sont améliorés lorsque la distribution de transition est ajoutée : on récupère des poses plus précises en utilisant un filtre avec presque 7 fois moins de particules.

Mots clefs: estimation de pose 3D, filtrage particulaire, distribution de transition, analyse en composantes principales

Chapter 1

Introduction

The purpose of monocular human pose estimation is to recover, given a single image sequence, the 3D pose of a person at each time step. A wide range of applications would benefit from a robust solution such as video surveillance systems, human-computer interfaces and video indexing and retrieval. Despite the significant amount of research that has been devoted, this is still an open problem [SB10]. There are several sources of challenges to this problem such as the high number of parameters that must be recovered, self-occlusions and depth ambiguities. Additionally, in general settings, people can wear different types of clothing and the lighting can vary greatly.

Human tracking methods can be classified into two general categories: model-based (generative) approaches and model-free (discriminative) approaches. Model-free methods learn a direct mapping from the image to the pose space by training on a labeled dataset. Once trained, discriminative models have the advantage of quickly produce a result, although the space of recovered poses can be limited to the exemplars used in learning. Model-based methods propose the use of a body model that can generate observations to evaluate pose hypotheses. Normally, these approaches require a search for the optimal pose in a very large search space, which can be slow. However, the accuracy of model-based techniques is usually better than most of model-free approaches. Because of this, in this work, we are mainly interested in model-based methods.

In generative approaches, tracking is usually formulated within a Bayesian framework. In this context, several algorithms have been proposed for human tracking [WN99, DR05, WR06]. Particle filtering has been very popular in human tracking [SBB10, DR05], due to the ability of this method to maintain multiple poses hypotheses. If one wants to employ a particle filter to track a person, two important parts of the system must be designed: the likelihood function and the temporal prior distribution.

The likelihood function is responsible for the evaluation of poses hypotheses at each frame, given the image observations. Different image features can be used in the likelihood function such as silhouettes [DR05, SBB10], edges [SBB10], optical flow [BFH10] and appearance models [GBRS07, RMR06]. We propose a likelihood function that combines silhouette and appearance information.

Temporal priors are used to propagate the particles from one frame to the next. In other

words, they represent the knowledge of the person's movement between two time steps. In simple functions, a random walk in the pose space is performed [SBB10]. However, the search space can be reduced by implementing stronger priors, such as physical-based priors [BFH10, VSJ08], or by applying dimensionality reduction techniques to the state space [SBF00, EL09]. We propose a motion model learned from mocap data that is capable of synthesizing walking cycles. This is used in tracking to constrain the search for poses to walking postures.

The rest of this report is organized as follows. Section 2 presents the problem of human pose estimation and discuss its difficulties. In Section 3, we review the literature on the subject. Section 4 presents our proposed method. Experiments using the HumanEva dataset [SB06] are shown in Section 5. Finally, we conclude and discuss further work in Section 6.

Chapter 2

Monocular human 3D pose estimation

Human pose estimation aims at recovering the 3D position and orientation for each body part of a person that is observed by a camera. When the task is performed at each frame using the previous estimates, the problem is often referred to as human tracking. Tracking is defined, in general terms, as the task of continuously identifying the position and orientation of an object with respect to the camera given a frame sequence where both the object and camera can move. Mathematically, given a rigid object defined by the set of homogeneous 3D points $\mathbf{M}$, the purpose is to find the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ of the projection equation[1]:

$$s\mathbf{m} = \mathbf{P}\mathbf{M}, \tag{2.0.1}$$

where s is a scale factor and $\mathbf{m}$ are the projections of the object 3D points. The projection matrix $\mathbf{P}$ can be decomposed as:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

As is common in the literature, the matrix of intrinsic parameters $\mathbf{K}$ *is assumed to be known*, i.e. the camera is calibrated. Finding the projection matrix $\mathbf{P}$ can be viewed as finding the object pose with respect to the camera coordinate frame. Alternatively, we can formulate the problem as to find the object pose w.r.t. the world coordinate frame — in this case, the transformation from the world coordinate frame to the camera coordinate frame must be known a priori.

A human body can be seen as an articulated object composed of several rigid parts corresponding to body parts. Thus, the Human Pose Estimation can be viewed as the process of recovering the configuration of the underlying kinematic structure of a person. The body configuration parameters are chosen beforehand and can vary from only the pose of the center mass as to a more complete model that determines the pose of each limb.

[1] More specifically, the equation represents a projection for the pinhole camera model.

Treating each limb as independent complicates the tracking as it increases the number of degrees of freedom (DoF) that are needed to be estimated (each limb is defined by 6 DoFs). Additionally, the human body has a structure that must be taken into account in the tracking, to not only simplify the pose, but also to impose realistic body configurations. The main approach to achieve this is to connect body parts using a kinematic tree.

The use of such structure makes the 3D pose of given limb be dependent of the 3D pose of the root node (for instance, the pelvis) and the rotations between body parts that are adjacent in the kinematic tree. For example, given that the root node is the pelvis, the 3D position of the foot depends on the global position and orientation of the pelvis and the angles for the knee and ankle. Formulated in this fashion, the configuration of the articulated body can be represented by a state vector $\mathbf{x}$:

$$\mathbf{x} = [\tau_x^r, \tau_y^r, \tau_z^r, \theta_x^r, \theta_y^r, \theta_z^r, \theta^{(i,j)} ...],$$

where $[\tau_x^r, \tau_y^r, \tau_z^r]$ is the position of the root node, $[\theta_x^r, \theta_y^r, \theta_z^r]$ its orientation and $\theta^{(\mathbf{i},\mathbf{j})}$ the relative angles between all pairs of limbs i and j that are connected in the kinematic tree. Each joint has up to 3 angles. For instance, the elbow joints often have only one DoF, as the shoulder joints have three.

Using this notation, the monocular human pose estimation problem can be formulated as the sequential recovering of the state vector $\mathbf{x}_t$ at instant t, using the observations $\mathbf{y}_t$ provided by the camera and the previous state $\mathbf{x}_{t-1}$. While there are robust methods for monocular tracking when planar or simple rigid models are used (see [LF05] for a good survey of these methods), estimating the motion of a person viewed by a single camera is still an open problem [SB10].

2.1 Monocular vs. multiview approaches

Human tracking is mostly a solved problem for controlled and static environments with several cameras and subjects wearing tight fitting clothes. Indeed, commercial applications for markeless motion capture in such conditions are available [SB10]. Relying on a single camera to capture human motion is a very challenging problem that is of great interest for applications where only one camera is available, which is often the case for video surveillance and human-computer interaction applications, for instance.

A major difficulty for monocular pose estimation are *depth ambiguities* created by the projection of 3D objects, in a 2D plane. Because of these ambiguities, two very distinct poses can fit equally well the same image observation, particularly if the image cue employed lacks depth information (e.g. silhouettes).

Considering that different configurations can explain one same image observation in a given frame, the estimation procedure must cope with this multimodal relation between pose and observations. Frequently, model-based approaches address this problem by employing a multiple-hypothesis trackers, such as particle filters, that can maintain different peaks in the likelihood function. Similarly, for model-free approaches, this means that the mapping from the observation to the pose must be multivalued.

Another challenge in human pose estimation are the self-occlusions between body parts. This implies that not all degrees of freedom are observable in a single monocular image (actually, it is estimated that at least a third of the DOFs are nearly unobservable [ST03]). Self-occlusions must be detected in order to avoid mis-attribution of image features to occluded regions.

Many other difficulties in human motion tracking (that are not exclusive to the monocular case) exist, this report will not attempt to review all of them, but the reader is referred to [Smi06] for a comprehensive review.

Chapter 3

State of the art in Human Motion Estimation

Human pose estimation has been an active field of research for many years and the associated literature is very extensive. Particularly, in the last two decades, vision-based approaches have received a great deal of attention since many applications, such as surveillance and Human-Computer Interaction, would benefit from a robust solution. In the effort to summarize these works, several surveys were made [Pop07b, FAR06, MHK06]. In order to organize the large amount of techniques that had been proposed, different taxonomies were employed. We chose to divide the range of methods into two classes, similar to [Pop07b]: model-based and model-free approaches.

Model-based (generative) approaches employ a model of the human body and a likelihood function (or cost function) that is used to find optima with respect to observations provided by the cameras. The model must be accurate enough, but also relatively simple not to increase the computational cost of the method to prohibited levels. We address both the model and model-based methods in section 3.1.

Model-free (discriminative) methods try to obtain a direct relation from image observations to poses. Two main classes of model-free methods can be identified: learning-based and example-based. In learning-based techniques, a mapping between pose and observation is learned from a training set consisting in pairs of images and corresponding poses. Example-based methods maintain a large database describing poses both in the image and pose space. These techniques are addressed in section 3.2.

3.1 Model-based approaches

3.1.1 Body models

Using a body model for tracking can simplify the evaluation of poses hypothesis and provides some flexibility in pose estimation (e.g. motion constraints can be added in a natural way). However, one must be careful in choosing the appropriate model such that it rep-

resents well a real human body but still is simple enough to make the algorithm run in reasonable time. These models can be described in 2D or 3D.

2D models are suitable to recover motion that is parallel to the image plane or to find body parts locations (these 2D locations can be "lifted" to 3D poses afterwards). An example of such models is the commonly used "cardboard" model in which body parts are represented using planar patches [JBY96]. More recently, pictorial structures [FH05] have been introduced to Pose Estimation to detect and describe body segments in 2D [LH04, ARS09] where the spatial relations between parts are described by prior distributions.

A wide range of body models were proposed to describe a person in 3D. Usually, to represent the kinematic structure of a human body, a tree is employed to represent the relations between parts [BB06]. To represent the body outer shape, each segment is described using, for instance, a tapered cylinder [BB06, SBB10]. More complex models use super-quadratics [GD96] or polygonal surfaces [BB09]. In [PF01], the authors use an even more complete model, with a layer to simulate muscles and fat tissues.

Clearly, the number of degrees of freedom (DoF) can vary for each type of body model, going from roughly 10 for simple ones to more than 50 in complex ones. When a single camera is used for tracking some of the DoFs are unobservable, so simpler models are preferred [BFH10, ARS10]. However, regardless the model that has been used, kinematic constraints are often used to prevent self intersections and to limit joint angles (which can be learned from data or be fixed using anatomical joint limits [Pop07b]).

3.1.2 Pose Estimation

The reconstruction of full-body motion can be formulated as an incremental or as a batch problem. In incremental methods, the pose is estimated (or updated) each new image observation. On the contrary, batch approaches optimize poses over a sequence of frames. Examples of batch methods are found in [ARS10] and [FDLF10]. Both methods employ Hidden Markov Models (HMMs) that are used to refine early pose detections by finding the most probable sequences of states in the Markov chain.

Incremental tracking for motion estimation is usually formulated as a Bayesian inference task, such that the objective is to estimate the current posterior distribution in the pose space given the image measurements. The approximation of this posterior can be obtained through several methods such as optimization techniques or Kalman filter [WN99]. However, the projection of a 3D object in a 2D image creates ambiguities that are best addressed by considering multiple hypothesis in the tracking. As human motion is also non-linear, particle filtering (described in Section 4.2) have been commonly used for this task [DR05, ST03, SBB10].

Even though particle filtering is a very flexible framework, the high number of DOF present in the state requires a high number of particles to accurately estimate the pose — as the likelihood of each particle must be measured, there is a limitation on the number of particles that can be processed by the system while still obtaining a reasonable computation time.

To solve this problem it is possible to spread particles in areas where the maximum likelihood is probable to happen — for instance, in [DR05] the authors introduce the Annealed Particle Filter that resamples the particles several times at each frame to gradually concentrate particles around global maxima. With the same purpose, Sminchisescu and Triggs [ST03] propose the Covariance Scaled Sampling (CSS) which guide the particles around maximas.

It has been shown that these algorithms can increase accuracy (see [SBB10, WR06] for evaluations). However, if the likelihood function and temporal prior are poorly designed, no resampling strategy will solve robustly the problem in the presence of occlusions or ambiguities. A more general way to increase reconstruction quality in such frameworks is to model temporal priors that reduce the number of particles required (by adding motion constraints) and to construct likelihood functions that are more discriminative.

3.1.3 Temporal priors

Even if the number of possible configurations in a human body model is immense in theory, the actual subset of achieved ones is much smaller for a given activity. As previously discussed, simple constraints that exclude configurations with self intersections or joint angles over thresholds can improve accuracy in tracking [SBB10]. To further reduce the number of possible configurations, physical constraints can be applied.

Vondrak et al. [VSJ08] enforce physical plausibility simulating the human body dynamics and interaction with the environment to avoid, for instance, footskate and configurations where a foot intersects the ground. To achieve this, they first extract a goal position for each frame from a mocap database using the current state as input. Then, a physical simulation takes place to detect collision and to generate new state hypothesis — including the velocities of each joint. The method increases motion realism while reducing localization error.

Brubaker and Fleet [BFH10] present a more simple physical-based model for lower-body dynamics that simulates walking by applying forces to a spring-mass system. This dynamics model is planar and the state only depends on four parameters: two angles related to the orientation of the legs and their velocities. Because the model is too simple to reconstruct the 3D pose, tracking uses a 3D body model that is, at each time step, constrained to match the simulated dynamics. The method can work well even in presence of occlusions and does not employ mocap data. Nonetheless, the approach does not easily generalize for full-body pose estimation and the likelihood function used in the experiments do not solve depth ambiguities (which are common in monocular tracking).

Physical-based models can be good to account for only plausible poses and to introduce temporal consistency, but they can be suitable for only one range of activity, such as walking. In this case, generalization can be difficult to achieve.

3.1.4 Likelihood functions

Likelihood functions are used to measure how well a hypothesis explains the current image observations. To model good likelihoods is a challenging task due to several difficulties: the problem of finding which pixels in the image correspond to a person, called data association, can be surprisingly hard [FAR06]. Moreover, the model must take into account ambiguities caused by the camera projection and frequent self-occlusions.

Silhouette is nowdays the feature that is most used to design likelihood functions [MHK06, Pop07b, BFH10]. Silhouettes can be recovered easily in controlled environments, but the lack of depth information makes estimation harder, specially for monocular tracking (it can be easily seen that two very distinctive body configurations can create the same silhouette).

Wang and Rehg [WR06] associate templates, that are initialized in the first frame, to each body part. At inference time, these patch templates are compared (using SSD) with hypothesis generated using a particle filter. Yet simple, the model is not adaptive and does not take into account self-occlusions.

Balan and Black [BB06] use an adaptive model based on the Wandering-Stable-Lost (WSL [JFEM03]) framework which is extended to cope with self-occlusions. Each pixel in their appearance model is described by an 1D WSL model, i.e. a mixture model that includes: a stable component S that adapts to slow changes, the wandering component W to deal with rapid changes and the lost component L to reject outliers. The likelihood function is defined using these WSL models with the goal of aligning coherent structures over time (the function also includes silhouette information). Results showed that performance is improved (in multi-view scenarios) with respect to the case where only silhouettes are used.

In [RMR06], the authors propose a method which builds appearance models based on color histograms that are constructed on-line from monocular images. For tracking, a given pose is evaluated by first synthesizing an appearance model using the pose and then comparing it with a model that was obtained at initialization — this initial model is also constantly updated using the means of the posterior density. Fossati et al. [FDLF10] propose a method that first detect specific postures of a walking scene using spatial-temporal templates [DLF06]. Then, from these detections, an appearance model based on a color histogram for each limb is computed in the region of the image corresponding to the projection of a 3D body part. After this learning phase, the color histograms are used to synthesize images of the appearance model given an specific pose; finally, a search for the pose that minimize the difference between the synthesize image and the current frame is performed.

3.1.5 Dimensionality reduction

Despite the fact that the pose space is high-dimensional, many of the human activities are located on low-dimensional latent spaces [EL08]. If this latent space is recovered, tracking can be performed using fewer particles. We are mostly interested in generative methods

that use dimensionality reduction, but the technique has also been applied to discriminative approaches as in [NFC07], for instance.

Tracking in this low-dimensional manifold requires a mapping from the latent space to the pose space and its inverse. Several methods to construct this mapping had been used with success for pose estimation, such as: Locally Linear Embedding [LE10], Isomap, Locally Linear Coordination (LLC) [LhYST06], Gaussian Process Latent Variable Models (GPLVM) [TLS05, UFHF05].

Li et al. propose in [LhYST06] a method to learn an embedding space for body configuration and use it for tracking. In the first step, an offline algorithm applies LLC to learn the mappings using body centered mocap data of a given activity. The points in the latent space form clusters such that, if two points are close, they will correspond to poses that are also close in the original space. In the online stage, 3D pose estimation is done using a multiple hypothesis tracker where the state is defined by a 3D global position (corresponding to the pelvis) and a point in the latent space. New hypothesis are generated using the embedding space with a dynamical model that assumes constant velocity.

The observed motion, described by silhouettes, contours or other features, also lie on low-dimensional manifolds, referred here as the visual manifold. The problem of combining a body configuration manifold with a visual manifold was addressed in [EL09] and [LE10]. Elgammal and Lee [EL09] show that the kinematic manifold and the visual manifold are related through a latent variable that represents body configurations. But, unlike the kinematic manifold, the visual manifold depends also on the viewpoint and style (the latter is related to the shape of the observed human). To construct both manifolds, they consider observations obtained from a view circle around a person performing a periodic motion. If a specific body configuration is fixed, the obtained view manifold is homeomorphic to the unit circle. The same can be noticed when a given view is fixed and we extract a body configuration manifold. This suggests that, ideally, the visual manifold should lie in a topological structure that is homeomorphic to a torus.

A mapping from the torus to the visual data (such as silhouettes) is computed in two steps. First, a mapping is built from the pairs of observations and kinematic data to the torus for several viewpoints. Second, these relations are used to fit a mapping function from the torus space ($\mathbb{R}^3$) to the visual data space using several radial basis functions. This continuous mapping is used to synthesize observations (given a body configuration and a view orientation) which are used in tracking. A particle filter is employed. The state vector consists of the coordinates in the torus and a shape parameter that is time invariant and detected in the first frames.

Dimensionality reduction is good solution to both create a motion model for tracking and reduce the computational cost of the system while achieving high accuracy — the discussed approaches show state of the art results. However, it is only possible to recover motions of the activity observed in the learning phase.

3.2 Model-free approaches

Rather than modeling a temporal prior and a likelihood function, model-free methods model a direct relation from the image to the pose space. For such methods, two main classes can be identified: example-based and learning-based. Notice that although the mapping from the image to the pose is known to be multi-valued, most methods consider as if it was single-valued [Pop07b].

3.2.1 Example-based

This class of methods keeps several exemplars of images (or features) together with the corresponding pose in a database. Then, for an input image, a search is performed to find the most similar image and the associated pose is returned (possibly, several poses from the set of most similar images can be interpolated to produce the result).

Poppe [Pop07a] uses histograms of oriented gradients (HoG) within the bounding box of silhouettes to encode each exemplar in the database. For estimation, the HoG of the input image is computed and compared to all dataset examples using the Manhattan distance. Then, the n closest entries are interpolated (in body-centered coordinates) to generate the final pose.

Mori and Malik [MM06] propose a database of 2D views of a person taken in a variety of different configurations and viewpoints. For each of these images, the body parts locations (in 2D) are also stored. To recover the pose, the input image is matched with the database using shape contexts [BMP02], i.e. the stored exemplars (including their 2D locations) are deformed to match the image observation. 2D joint locations are then "lifted" to estimate the 3D pose.

Howe [How04] uses a database of artificially rendered silhouettes associated with their respective poses. The estimation for each frame is proceeded as follows: the silhouette is first extracted and compared to the database using Chamfer distance and turning angles metric. Several candidates are obtained and temporal continuity is enforced via a Markov chain that removes unlikely pose sequences. Finally, the result of the Markov chain is smoothed and an optimization takes place to increase accuracy and reduce jitter. This work is extended in [How05] to use optical flow information.

Example-based pose estimation requires a high number of exemplars that are often of high dimensionality. To efficiently search in such databases, Shakhnarovich et al. [SVD03] propose the use of several hash functions to increase inference speed. However, even if the search is quick, the pose space is limited to the examples in the database.

3.2.2 Learning-based

Learning-based methods try to learn a function that maps directly from the image features to the pose space. In other words, using the same notation as in equation (4.2.1), the goal is to estimate $p(\mathbf{x}|\mathbf{y})$ where $\mathbf{x}$ represent the pose and $\mathbf{y}$ the observation (encoded in image

features). As the mapping from the observation to the pose space is multi-valued, $p(\mathbf{x}|\mathbf{y})$ must be multi-modal.

Agarwal and Triggs [AT06] use shape contexts to encode artificially created silhouettes from a single view. A non-linear regression is used to model the mapping between histograms of shape contexts and poses. To address ambiguities, created by the use of silhouettes, dynamics are applied at each time step.

The dimensionality of both pose and feature spaces is usually high, hence, to learn $p(\mathbf{x}|\mathbf{y})$ requires a large training dataset. Navaratnam et al.[NFC07] provide a technique to use unlabeled data to train a regression model by including in the learning phase the marginal distributions $p(\mathbf{x})$ and $p(\mathbf{y})$, which can be easily obtained. To this end, they extend the GPLVM algorithm such that it can also learn from marginal distributions. When used for tracking, the regression gives multiple hypotheses that are treated as states in a Markov Chain model (transition probabilities are learned from data). The Viterbi algorithm is then used to compute state sequences.

Bo et al. [BSKM08] also address the problem of large training dataset. They propose an extension to the Mixture of Experts algorithm that can be learned from a large dataset (more than $100,000$ data points) in reasonable time. Different image features can be used with this method, they present three: histograms of shape context descriptors, histograms of SIFT descriptors sampled on the silhouette and hierarchical image descriptors (described in [KSM07]). Experiments show good performances, specially when context descriptors are included.

Chapter 4

Proposed method

In this section, we propose a model-based human tracking method that estimates the body configuration at each time step, i.e. the approach is incremental. The initialization pose is assumed to be given and close to the ground truth configuration. The tracking is formulated as a Bayesian inference problem, which is discussed in Section 4.2. Next, in Section 4.3, different likelihood functions are presented and in Section 4.4, temporal priors are addressed.

4.1 Body model

The body model used for tracking is based on the one proposed in [SBB10]. It is composed of 15 body parts represented by truncated cones, as shown in Figure 4.1.1. The joints corresponding to the shoulders, hips, thorax and neck have 3 DoFs each, while clavicles have 2 DoFs. The remaining joints (knees, ankles, elbows and wrists) are modeled with only one DoF. We assume that the length and width of body parts are known and fixed. Therefore, a body configuration can be completely described with *36 parameters*, comprising 6 parameters related to the global position and orientation of the pelvis and 30 values for the relative joint angles between limbs.

In order to recover the position and orientation of each limb with respect to the world coordinate frame, direct kinematics are applied using the hierarchy of the kinematic tree depicted in Figure 4.1.2. This tree is used to describe the body parts hierarchy, in such a way that each non-root node can be placed with respect to its parent by a local transformation. In other words, a given limb position and orientation with respect to the root node can be easily recovered by traversing the tree starting from the root and, at each node, concatenating the local transformation matrices.

As discussed in Section 3.1.1, the choice of body model must take into account several aspects because it has a great impact in tracking performance. In one hand, if the model is too simple, we limit the range of activities that can be tracked and the information recovered is less informative (because few DoFs were used). On the other hand, if the body model is too complex, such as a model that represents body parts using triangular

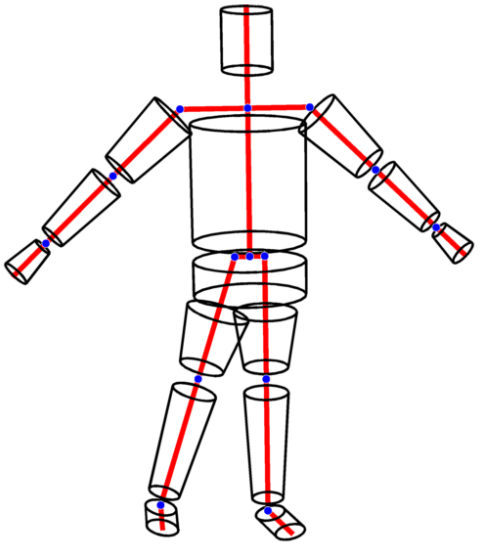
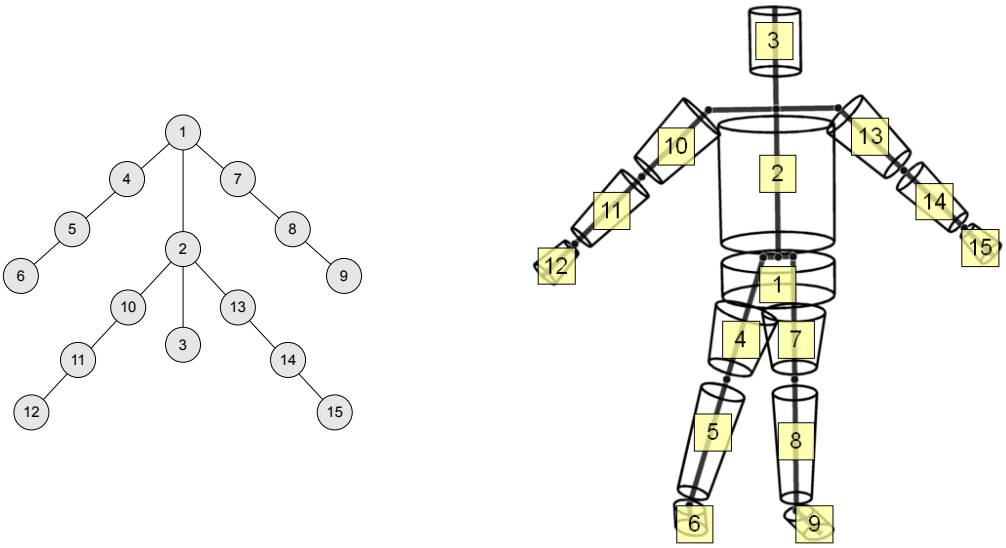


Figure 4.1.1: The body model is composed of 15 body parts and 36 degrees of freedom. Blue points represent joint locations.



(a) Kinematic tree

(b) Body model with indexed parts

Figure 4.1.2: Kinematic tree linking body parts that was used in the proposed method.

meshes, it can be difficult to adapt the model for different subjects and the computational cost of tracking can be increased to prohibitive levels. In this work, we tried to reach a middle ground in the complexity aspect by using a highly articulated model with body parts represented by simple primitives that can be easily adapted.

4.2 Bayesian Framework

As it is common in the literature, tracking is formulated as a Bayesian inference task. More precisely, let $\mathbf{x}_t$ denote the body parameters at time t, then the objective is to estimate the posterior distribution of $\mathbf{x}_t$, given the posterior from the previous time step $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$, a temporal prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and a likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$. This is done recursively by solving the Bayes equation:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y_t}|\mathbf{x_t}) \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}). \tag{4.2.1}$$

The Bayesian formulation is very popular in human tracking, partially because it provides a principled way of integrating prior knowledge on human motion and also because it allows to easily mix different image cues. Different methods, such as Kalman filters, have been proposed to solve the equation (4.2.1)[MHK06]. However, the multimodal aspect of the posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is better addressed in approaches that are able to maintain multiple hypothesis, such as particle filters.

4.2.1 Particle Filter

The purpose of particle filters is to approximate the posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ in equation (4.2.1) with a set of N samples, called particles. Each particle is composed of a state value $\mathbf{x}_t^{(i)}$ representing a body configuration and a weight $\pi_t^{(i)}$ that is proportional to the likelihood evaluated at the particle state, $\pi_t^{(i)} \propto p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$. Given an initialization, particle filter algorithms can be roughly described by three steps:

- Resampling: A new set of particles is drawn with replacement from the previous set $\{\mathbf{x}_t^{(i)}; \pi_t^{(i)}\}_{i=1}^N$. The probability of a given particle being chosen is proportional to its normalized weight. Resampling concentrates particles around posterior modes and eliminates the ones that are unlikely to be the truth state.

- Prediction: particle states are propagated according to the temporal prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. This step represents the grown of uncertainty due to the body movement between frames. Temporal priors try to model the knowledge about this movement. The simplest model for temporal prior consists in just adding a normally distributed noise to the previous state:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_{t-1}, \sigma) \tag{4.2.2}$$

- Filtering: the likelihood function is evaluated for each particle. The weights are then normalized, such that $\sum_{i=1}^N \pi_t^{(i)} = 1$, and the resulting set of particles approximates the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. Therefore, both the expected value $\hat{\mathbf{x}}_t$ and the maximum a posteriori $\hat{\mathbf{x}}_t^{MAP}$ of the posterior can be approximated from the particle set:

$$\hat{\mathbf{x}}_t = \sum_{i=1}^{N} \pi_t^{(i)} \mathbf{x}_t^{(i)} \tag{4.2.3}$$

$$\hat{\mathbf{x}}_t^{MAP} = \mathbf{x}_t^{(j)}, \pi_t^{(j)} = \max_i(\pi_t^{(i)}) \tag{4.2.4}$$

Although particle filters are very flexible and can cope with multimodal posteriors, the computational cost becomes very high when a large number of particles is needed. Unfortunately, that is the case in human tracking due to the high number of degrees of freedom that must be recovered. Many approaches, already discussed in Section 3, attempt to reduce the number of particles by better sampling the state space. We choose to employ the so-called Annealed Particle Filter (APF), proposed by Deutscher in [DR05], that has been shown to improve tracking results while maintaining the same computational cost [DR05, SBB10].

4.2.2 Annealed Particle Filter

The main idea behind the Annealed Particle Filter (APF) is to perform, at each inference time, several iterations (layers) that gradually concentrates more particles around the peaks of the posterior density distribution. The algorithm works by propagating the set of particles across layers using slightly different weighting functions.

Let $S_{t,m} = \{\mathbf{x}_{t,m}^{(i)}; \pi_{t,m}^{(i)}\}_{i=1}^{N}$ denotes the particle set at time t and layer m, then the particle weights are evaluated from layer M to 0 using a set of temperature parameters β_m:

$$\pi_{t,m}^{(i)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_{t,m}^{(i)})^{\beta_m}}{\sum_{j=1}^{N} p(\mathbf{y}_t|\mathbf{x}_{t,m}^{(j)})^{\beta_m}}. \tag{4.2.5}$$

A large β_m in equation (4.2.5) produces a peaked weighting function while a small value has a smoothing effect on the weights as illustrated in Figure 4.2.1 [DR05].

In order to propagate the particles from the layer m to the layer $m-1$, states are drawn randomly from $S_{t,m}$ with replacement and with a probability equal to their weights. Then, a noise is added to the particles, similar as shown in equation (4.2.2):

$$\mathbf{x}_{t,m-1} = \mathbf{x}_{t,m} + \mathbf{B}_m \tag{4.2.6}$$

where $\mathbf{B}_m$ is a noise drawn from a normal distribution:

$$\mathbf{B}_m \sim \mathcal{N}(0, \alpha^{M-m}\mathbf{\Sigma}) \tag{4.2.7}$$

Usually, α is assumed to be 0.5 which decreases the covariance matrix $\mathbf{\Sigma}$ and therefore decreases the variance of particles at each layer. When the last layer ($m = 0$) is processed, the set of particles $S_{t,0}$ is used for the next time step, i.e. $S_{t+1,M} = S_{t,0}$.

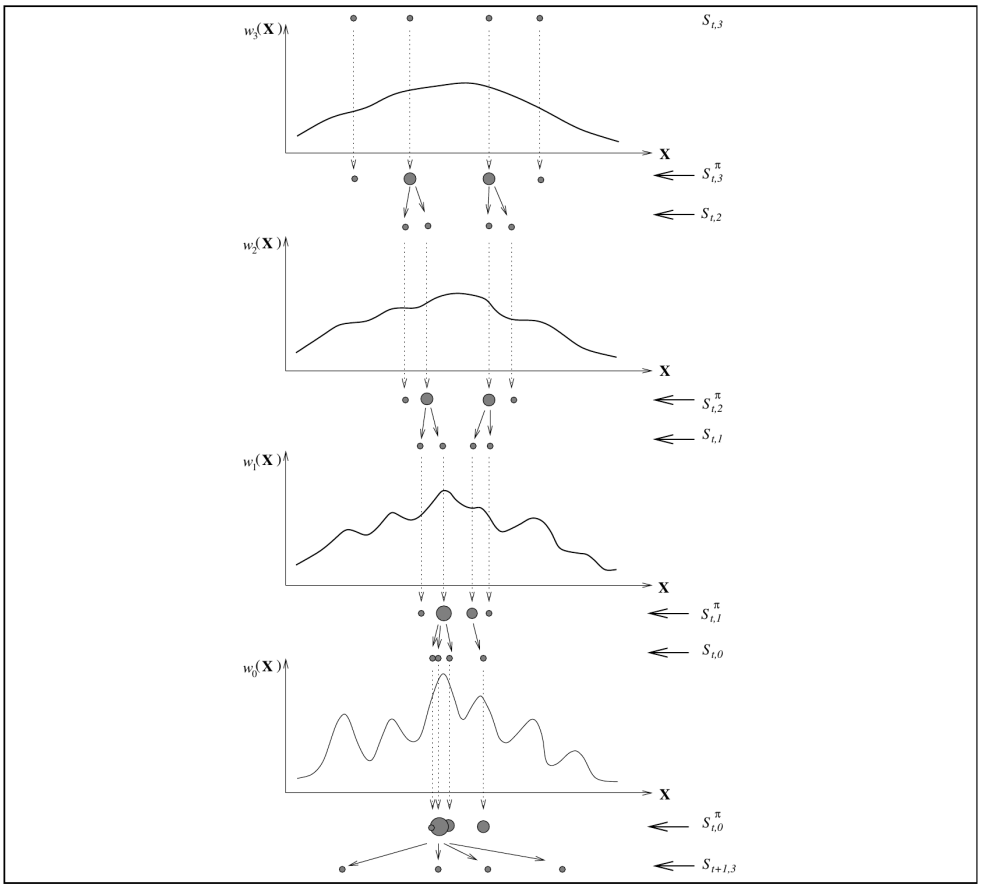


Figure 4.2.1: An illustration of the annealed particle filter with $M = 3$ [DR05].

4.3 Likelihood functions

The likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$ is an important part of a Bayesian tracker that is responsible for describing the relation between image observations and poses hypotheses. Several likelihood functions were proposed for human tracking (Section 3.1.4) and different image cues can be combined. We suggest that different likelihoods can have complementary qualities, which led us to an hybrid approach. The next sections present the likelihoods used in our proposed method.

4.3.1 Silhouette

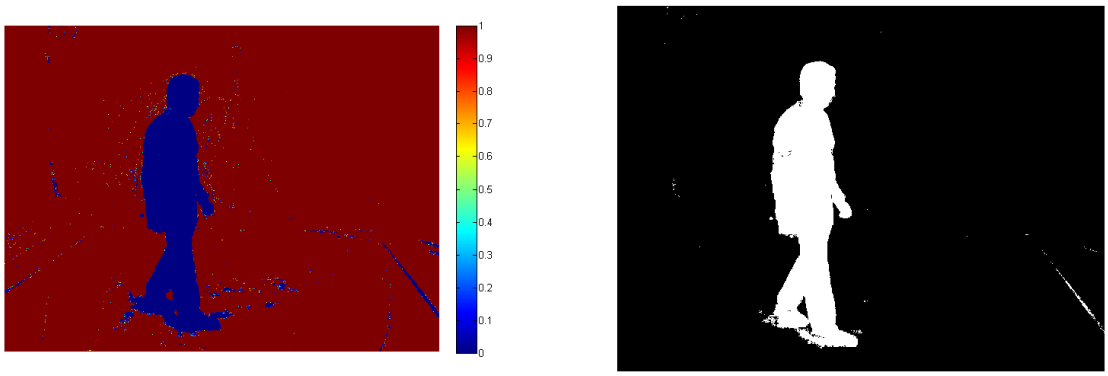
Human silhouette (also called figure-ground segmentation) is a very popular image feature that has been used to design different likelihood functions. The reason is twofold: first, silhouettes encode a great deal of information that can help tracking, specially in multi-view settings. Second, they are easy to recover from scenes with static background.

In order to extract silhouettes, a model of background must be learned from a set of static background images. The procedure extracts the mean and the variance of pixel

values from the entire set of images and for each channel separately. These values are used to parametrize a Gaussian distribution for each pixel location $\mathbf{p}$ and channel c, denoted $\mathcal{N}(\mu_p^c, \sigma_p^c)$. At runtime, the current frame is segmented by evaluating its distance from the learned distributions at each $\mathbf{p}_i$ location. These results in a probability map as the one shown in Figure 4.3.1a that is subsequently filtered to generate the silhouette map M_s(Figure 4.3.1b).

$$M_s(\mathbf{p}) = \begin{cases} 0 & \text{if } \prod_c \mathcal{N}(\mathbf{I}_c(\mathbf{p}); \mu_p^c, \sigma_p^c) > \epsilon_s \\ 1 & \text{otherwise} \end{cases} \tag{4.3.1}$$

where $\mathbf{I}_c$ is the c-channel image of the current frame and ϵ_s is a threshold that is usually determined empirically.



(a) Probability map of the background

(b) Extracted silhouette

Figure 4.3.1: An example of human silhouette extraction from the HumanEVA dataset [SB06].

To evaluate a given pose, using the body model described in Section 4.1, a silhouette can be synthesized by projecting the model w.r.t. the camera to create a binary map M_m. Then, a matching function based on the area overlap between the two silhouettes (one extracted from the frame and another based on the model) is used:

$$-\ln p_s(\mathbf{y}_t|\mathbf{x}_t) \propto \frac{1}{N}\sum_{i=1}^{N}(1 - M_s(\mathbf{m}_i)))^2, \tag{4.3.2}$$

where N is the number of sampled points $\mathbf{m}_i$ inside the model projection. Figures 4.3.2c and 4.3.2f are illustrations of the likelihood evaluation for two distinct poses — in these images, the blue area correspond to the overlap region.

The likelihood function (4.3.2) is a simple and somewhat naive measure for monocular tracking. The reason is that it only takes into account the number of foreground pixels

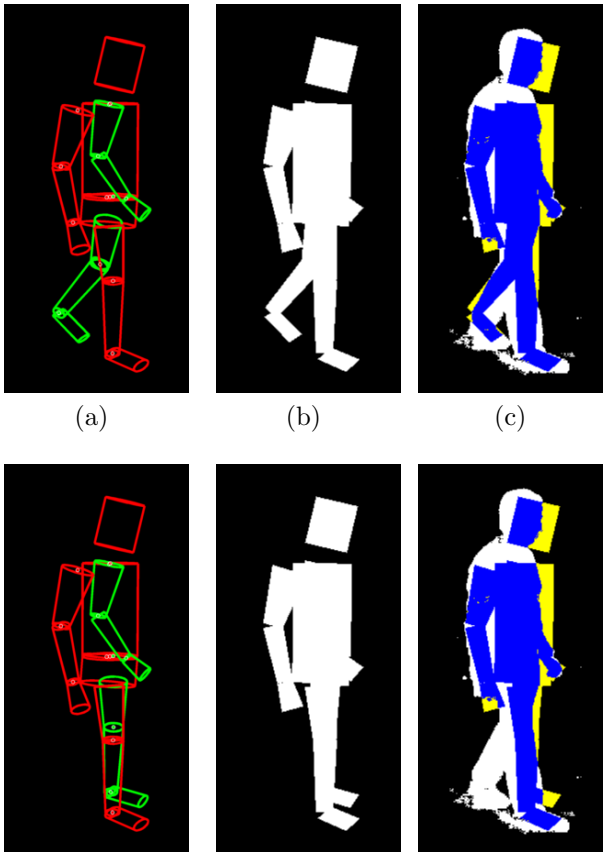


Figure 4.3.2: Two examples of (naive) likelihood evaluation based on silhouettes. First row: a pose (a) near to the correct one is used to generate a synthetic silhouette (b) and the overlap of this with the foreground extracted from the frame is computed (blue region in (c)) using equation (4.3.2). Second row: a pose (d), with a large deviation of the left leg correct pose, generates the silhouette (e) and the overlap is computed (f). Notice that even if the first pose is clearly better than the second one, the likelihood values will be similar.

covered by the model and does not penalizes the pose for the set of points in the silhouette map that were not covered. Despite this limitation, this formulation is frequently used in state of the art methods because it is computationally efficient: the generated silhouette M_m can be easily subsampled to evaluate a pose. Moreover, the problem can be mitigated in multi-view approaches by using observations provided by all cameras. However, for monocular scenarios, this problem cannot be disregarded.

To address this issue, the likelihood must be reformulated to penalize poses with body projections that do not cover the entire silhouette. Several alternatives were proposed [GBRS07, SBB10, DLC08] and we chose to describe here the so-called bi-directional silhouette, as proposed by Sigal and Balan [SBB10]. The objective is to minimize the non-

overlapping areas, such as those depicted in white and yellow in Figure 4.3.2f. The number of pixels, N_w, that are not covered by the synthetic silhouette in M_m can be computed over all image pixels $\mathbf{p}$ as follows:

$$N_w = \sum_{\mathbf{p}} \left[M_s(\mathbf{p})(1 - M_m(\mathbf{p})) \right]. \tag{4.3.3}$$

In the same manner, the number of non-zero pixels in M_m that are outside the image silhouette M_s is defined by:

$$N_y = \sum_{\mathbf{p}} \left[M_m(\mathbf{p})(1 - M_s(\mathbf{p})) \right]. \tag{4.3.4}$$

The likelihood (4.3.2) is redefined as the combination of these two regions, such that:

$$-\ln p_s(\mathbf{y}_t|\mathbf{x}_t) \propto \frac{1}{2} \left[\frac{N_w}{\sum_{\mathbf{p}} M_s(\mathbf{p})} + \frac{N_y}{\sum_{\mathbf{p}} M_m(\mathbf{p})} \right]. \tag{4.3.5}$$

The bi-directional formulation can help improve accuracy [SBB10] for both monocular and multi-view tracking. However, as previously discussed, monocular silhouettes introduce depth ambiguities, as shown in Figure 4.3.3[How04]. The image is an example where simultaneous left-right inversion of the pose yields to identical silhouettes. It is impossible to avoid right-left reversal in trackers that are solely based on silhouettes. This suggests that a more discriminant image feature must be added to the likelihood function to increase tracking results.

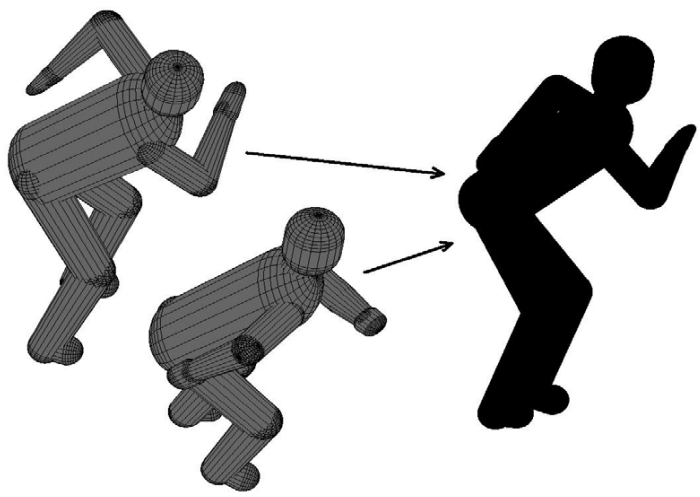


Figure 4.3.3: Illustration of two very distinctive poses that generate the same silhouette [How04].

4.3.2 Appearance

The observation that the appearance of a person remains mostly unchanged in a frame sequence has inspired several likelihood functions based on the appearance of individual body parts [RMR06, GBRS07, Pop07b], often described with color histograms or image templates. Our approach is based on color histograms compute in CIELab color space.

During initialization, an appearance model composed of several histograms is built: torso, pelvis and head are represented by one histogram each and pairs of symmetrical limbs, e.g. left and right upper arm, are grouped together in a single histogram. In other words, the model *assumes that left and right limbs have the same appearance.* Figure 4.3.4 is an illustration of the procedure employed to extract the appearance model given a pose and a frame.

In the first step, the RGB image is converted to CIELab space, which separates the lightness of the color (L-channel) from the color channels (a and b); only the a- and b-channels are used. Next, the body model is projected, with respect to the camera coordinate frame, using the pose from the input. Regions of the image that are inside a given projected cylinder are sampled to create the set of 2D histograms that compose the appearance model.

To avoid misattributions of pixels caused by self-occlusions, a visibility map is constructed as proposed in [BB06] by first sorting the model cylinders in decreasing distance from the camera. Then, body parts are rendered in order to generate a map such as the one in Figure 4.3.4. Each pixel in this map contains the index of the body model visible at that location. In the sampling step, we check if the pixel of a given limb is not occluded by another limb using the visibility map.

In order to measure a particle using this appearance model, the set of histograms built in the initialization step is compared to the histograms extracted from the particle state. Let $\hat{A}_p = \{\hat{h}_b\}_{b=1}^{B}$ be the appearance model composed of $B = 9$ histograms $\hat{h}_b$ computed at initialization and $\{h_b\}_{b=1}^{B}$ the set of histograms extracted from the current frame given the particle state, then a likelihood function can be formulated using the distance from these two models:

$$-\ln p(\mathbf{y}_t|\mathbf{x}_t) \propto \sum_{s=1}^{B} w_s \left[BC(\hat{h}_s, h_s)\right], \tag{4.3.6}$$

where w_s are normalized weights that are proportional to the body part sizes. They are defined at initialization using the ratio between the number of sampled points for a given part and total number of sampled points. $BC(\hat{h}_s, h_s)$ is called the *Bhattacharyaa* coefficient, defined as follows:

$$BC(h_1, h_2) = \sum_{i=1}^{N_{bins}} \sum_{j=1}^{N_{bins}} \sqrt{h_1(i,j)h_2(i,j)}. \tag{4.3.7}$$

The likelihood function (4.3.6) does not take into account how many pixels were sampled to generate the particle appearance. For this reason, a particle can receive a good weight

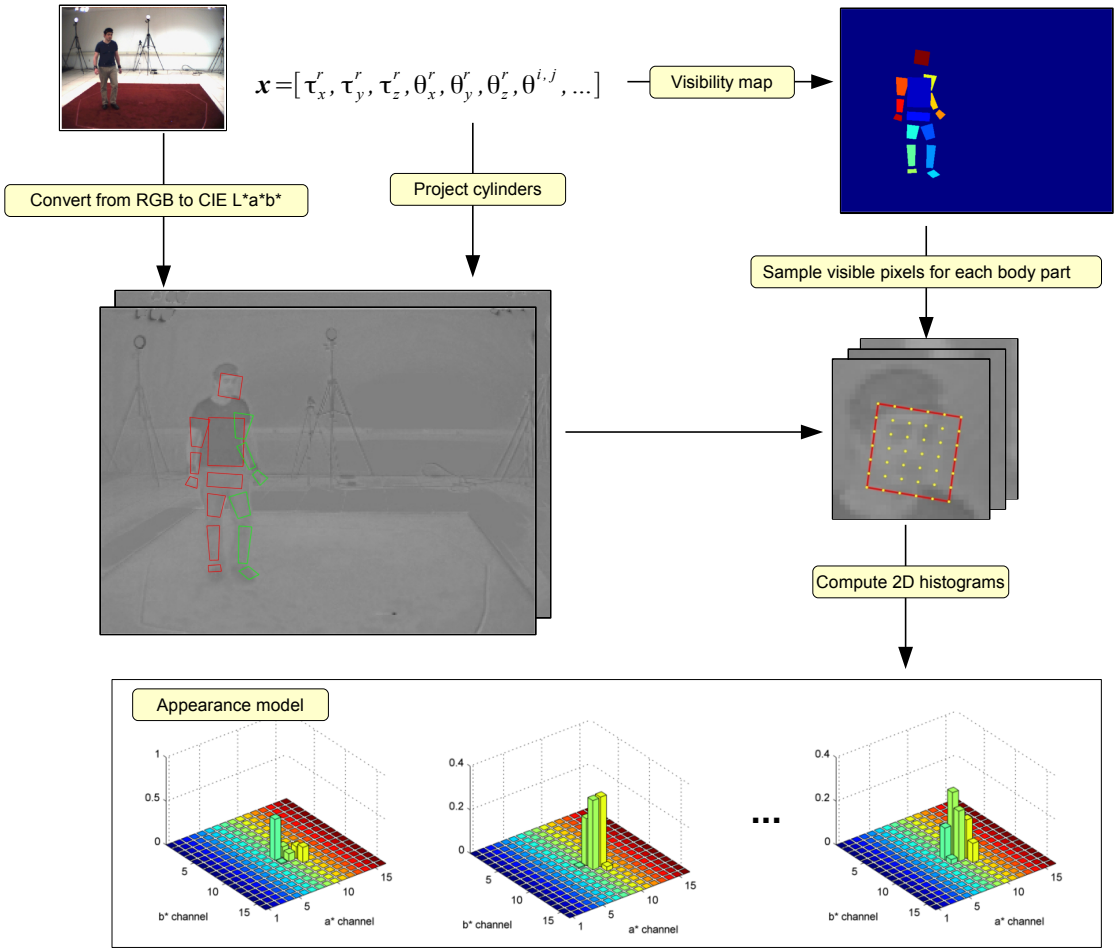


Figure 4.3.4: Illustration of the process that builds an appearance model.

even if the person silhouette is not entirely covered by the projection of the cylinders. Therefore, tracking that is based solely in this measure tends to poorly estimate limb positions. This problem can be addressed by mixing the silhouette information, as defined by equation (4.3.2), to the weighting function (4.3.6). Let $p_s(\mathbf{y}_t|\mathbf{x}_t)$ be the silhouette log-likelihood and $p_a(\mathbf{y}_t|\mathbf{x}_t)$ the appearance log-likelihood, then the mixed version of the likelihoods is:

$$-\ln p(\mathbf{y}_t|\mathbf{x}_t) \propto (1-\alpha)p_s(\mathbf{y}_t|\mathbf{x}_t) + \alpha p_a(\mathbf{y}_t|\mathbf{x}_t), \tag{4.3.8}$$

where α controls the importance of each function in the mixed likelihood. We carried out experiments, described in Section 5, that suggest that this mixed version of likelihood

significantly improves tracking results.

The appearance model extracted in the first frame represents the system approximation of the real appearance of the person. This estimation is certainly limited: in the first frame not the whole body is observed by the camera and the appearance can change along the sequence. This suggests that the model must be adapted at runtime using the previous pose estimates [RMR06, GBRS07].

Let $\bar{A}_p = \{\bar{h}_b\}_{b=1}^{B}$ be the set of histograms extracted from the previous frame using the expected value of the pose posterior, $\bar{\mathbf{x}}_t$. Then, each histogram $\hat{h}_s$ for each body part s, is updated as follows:

$$\hat{h}_s = \frac{(1-\gamma)\hat{N}_s\hat{h}_s + \gamma\bar{N}_s\bar{h}_s}{(1-\gamma)\hat{N}_s + \gamma\bar{N}_s}, \tag{4.3.9}$$

where $\hat{N}_s$ and $\bar{N}_s$ are the number of pixels that were sampled to generate $\hat{h}_s$ and $\bar{h}_s$, respectively. These values are used to lower the relevance of histograms computed from body parts that are partially occluded. The parameter γ controls the adaptation speed of the histograms.

In our experiments we found that, despite the fact that this adaptation can help the tracking, the described method is very sensible to noisy/incorrect pose estimates. This reduces the ability of the system to recover from failure because the appearance model of the person keeps being (incorrectly) updated in the frames where the tracker is lost.

4.4 Temporal priors

As previously discussed, the temporal prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is responsible for modeling the knowledge about the human motion in between two time steps. Several formulations have been proposed (see Section 3.1.3). Simple priors, such as the one defined by equation (4.2.2), assume a (Gaussian) random walk in the pose space and therefore does not encode much information about the activity that is being performed neither about the person's trajectory. However, this prior can be improved in two simple ways, as proposed in [SBB10].

First, it is possible to learn good noise parameters using motion capture (mocap) data. For instance, the standard deviation for a given angle can be defined as half of the maximum interframe change of this angle in the mocap sequences. The dataset can also be divided in different activities, in order to extract more specific variances. Second, since not all values for joint angles are anatomically possible, an anatomical range can be defined for each angle — these limits are generally defined by hand. These temporal priors have proved successful in increasing the accuracy of results and reducing the number of required particles [SBB10].

However, even if the state space of a body configuration is very large in theory, when a class of movement is observed, the actual set of achievable configurations is much smaller. So, it is possible to further limit the search space using activity-specific priors. As we are

mainly interested in tracking walking sequences, we propose the use of a *temporal prior specific for walking sequences*. The model is inspired by the fact that walking is a periodic motion which can be segmented in walking cycles. Figure 4.4.1 shows two joint angles extracted from a walking sequence (from mocap data) — it can be seen that the angles are highly correlated and that the movement is clearly periodic.

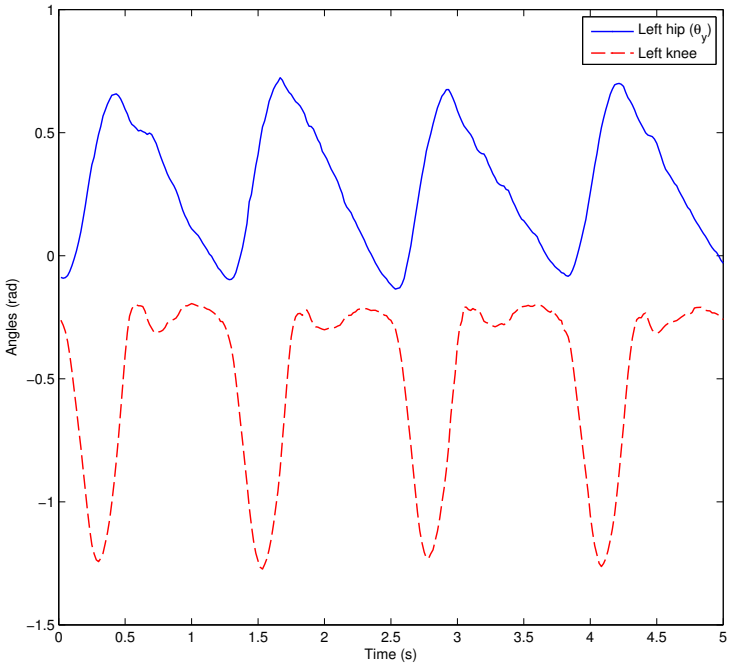
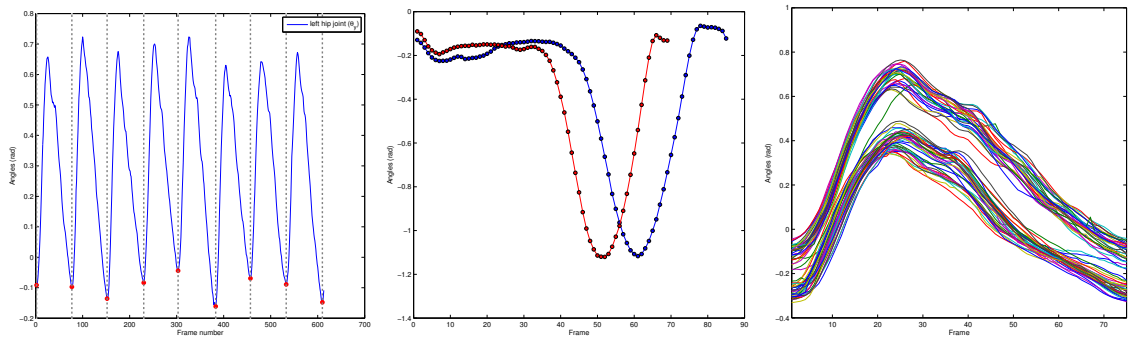


Figure 4.4.1: Joint angles extracted from motion capture data of a subject walking.

A motion model for walking can be learned from motion capture sequences using PCA [SBF00, UF04]. The data must be first segmented and scaled into walking cycles, which can be done using an automatic approach, as follows. Given a walking sequence described by joint angles over time, the minima of one angle (in our case, θ_y of the left hip joint) are extracted to detect the beginning of each cycle, as illustrated in Figure 4.4.2a. These intervals are used to segment all the angles for the whole sequence.

After segmentation, a set of walking cycles that start (and finish) with similar body postures is obtained. However, as shown in Figure 4.4.2b, the cycles have different lengths because of variations in the walking speed so they must be scaled to create cycles with equal number of samples. To scale the cycles, the data points are interpolated for all angles. We applied the above approach to the HumanEva-I [SB06] walking sequences (training subset) that is composed of three different subjects walking at different speeds. Figure 4.4.2c shows the left hip angle for all the 64 cycles extracted from the training data.

An eigenbasis can be learned using the Singular Value Decomposition (SVD) of the set of scaled walking cycles, but first the data must be arranged in a matrix. Given a cycle i, let $\mathbf{\Theta}_t^i$ be the vector formed by the n angle values of the cycle at time t, $\mathbf{\Theta}_t^i = [\theta_{1,t}^i, \theta_{2,t}^i, \ldots, \theta_{n,t}^i]^\mathsf{T}$, then a cycle can be represented by a concatenation of $\mathbf{\Theta}_t^i$ for all T time



(a) (b) (c)

Figure 4.4.2: Preparation of motion capture data before PCA analysis. The walking sequence in (a) is segmented (dashed lines) using the minima of the left hip angle (red dots); all angles are segmented using these intervals. (b) illustrates the need of scaling the data: the right knee values of two cycles have different lengths. After segmentation, all cycles are scaled to the same length, (c) shows the left hip angle for all cycles in the training set after segmentation and scaling.

steps, which results in a $nT \times 1$ row vector, called $\mathbf{a}_i$. All cycles $\mathbf{a}_i$ extracted from the training data are used to create the matrix A of dimensions $nT \times m$, where m is the number of cycle exemplars ($m = 64$ in our case).

To perform a PCA in the matrix A, the mean vector $\bar{\mathbf{a}}$ is first extracted from all columns of A. The matrix is then decomposed using SVD in order to extract the principal components of the data:

$$A = U \Sigma V^{\mathsf{T}} \tag{4.4.1}$$

where $U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m]$ are the principal components of the training set and Σ is a diagonal matrix with singular values $\lambda_1, \lambda_2, \ldots \lambda_m$ sorted in decreasing order along the diagonal. Principal components are the eigenvectors of the covariance matrix computed from the data. Each one of these eigenvectors has an associated eigenvalue such that eigenvectors with large eigenvalues are the most important to describe the data. Therefore, to reduce the dimensionality of the data, one can choose to keep only the $q < m$ first eigenvectors. The percentage of the database that q components can recover is given by,

$$g(q) = \frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{m} \lambda_i}. \tag{4.4.2}$$

When reducing the space dimensionality, computing g is useful to know how many eigenvectors are needed to keep most of the dataset variance. Figure 4.4.3 shows the value of g for different numbers of principal components.

In our experiments, we chose to use the first q components such that $g(q) \geq 0.95$, which is obtained using 11 principal components. Let $\tilde{U} = [\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_q}]$ be the reduced version

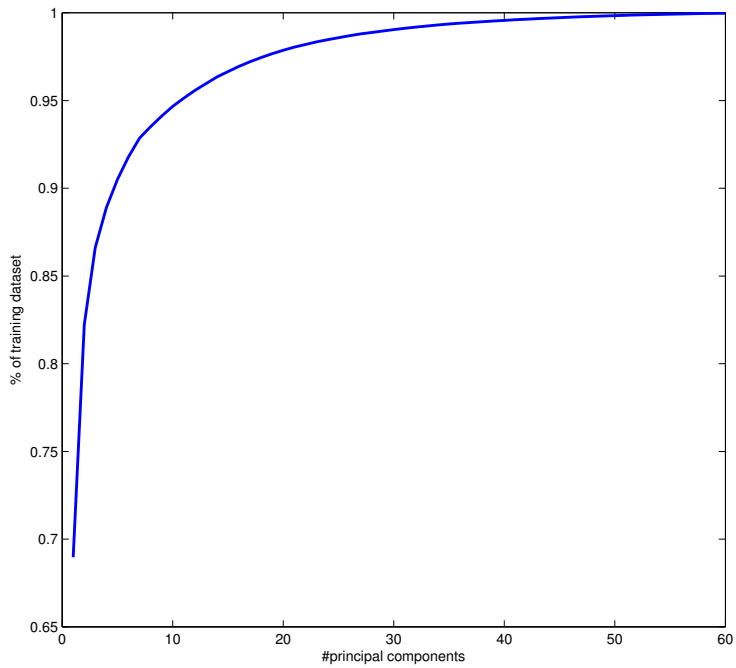


Figure 4.4.3: Percentage of the training set that can be generated using different numbers of principal components.

of U composed of the q eigenvectors with largest eigenvalues, then a walking cycle $\tilde{\mathbf{a}}$ can be *synthesized* using a subspace q-dimensional point $\mathbf{c} = [c_1, c_2, \ldots, c_q]$:

$$\tilde{\mathbf{a}} = \bar{\mathbf{a}} + \tilde{U}\mathbf{c}. \tag{4.4.3}$$

All the joint angles in our body model were included in the walking cycles with the exception of the head. The global translation and orientation were not included either because their values are generally not cyclic in a normal walking sequence. The subspace point and the phase, $\mu \in [0, 1]$, in current cycle can be added to the tracking to enforce a walking behavior to the poses and also to reduce the state dimensionality.

The state $\mathbf{x}_t$ is reformulated as the global translation and orientation of the pelvis, the joint angle of the neck (θ_y^h), the subspace point and the phase:

$$\mathbf{x}_t = [\tau_x^r, \tau_y^r, \tau_z^r, \theta_x^r, \theta_y^r, \theta_z^r, \theta_y^h, \mu, \mathbf{c}]. \tag{4.4.4}$$

The dimensionality of this state formulation is equal to 19, for $q = 11$, against the original 36-dimensional state described in Section 4.1. However, it is necessary to reconstruct from these parameters the original pose if we want to evaluate the likelihood functions in the same fashion as previously described. For a given pose $\mathbf{x}_t^{(i)}$, the walking cycle $\tilde{\mathbf{a}}^{(i)}$ is first computed using equation (4.4.3) and then the joint angles can be extracted from the cycle evaluating it at phase μ. Recall that $\tilde{\mathbf{a}}^{(i)}$ is formed by the concatenation of T vectors $\mathbf{\Theta}_t^{(i)}$. As the phase has values from 0 to 1, the index t of vector $\mathbf{\Theta}_t^{(i)}$ can be recovered by:

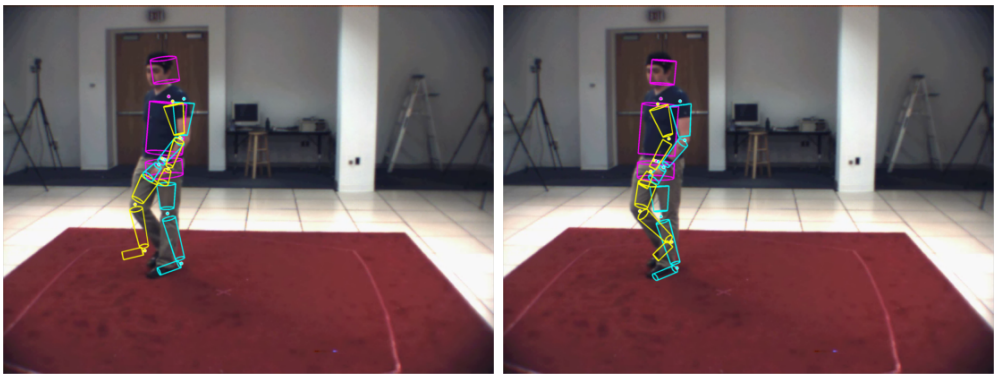
$$t = \mu(T - 1) + 1 \tag{4.4.5}$$

The state parameters, as before, are propagated in time using a Gaussian noise with the exception of the phase μ that is incremented at each time step:

$$\mu_t = \mu_{t-1} + \frac{1}{T} + B \tag{4.4.6}$$

where B is a small noise to account for speed changes. The phase is initialized manually while the PCA parameters are set to 0, i.e. the mean of the walking cycles used in the learning phase.

This motion model is able to enforce walking poses for the tracking, which increases accuracy while using fewer particles. However, the phase propagation in equation (4.4.6) is not suitable for sequences where the walking speed is far from the values in the training set. This occurs because the increment in the phase parameter is a constant value and defined by the training exemplars. Figure 4.4.4a shows the 93th frame of a tracking sequence where the pose is not well estimated because the phase parameter diverged from the correct value.



(a) (b)

Figure 4.4.4: Estimating the walking phase μ parameter in tracking: (a) when only a noise is used, the phase is not well estimated and the system loses track at frame 93. (b) if a parameter corresponding to the step of the phase at each frame is included, the tracker is able to recover the correct pose.

To address this problem, a step parameter s_t was included in the state space. It is initialized with $^1/_T$ and is propagated in time using a random noise. The phase is then incremented at each time set using this parameter:

$$\mu_t = \mu_{t-1} + s_{t-1},$$
$$s_t = s_{t-1} + B.$$

Tracking the step parameter is analogous to track the speed of the walking cycle. Figure 4.4.4b is a result of a tracking using the parameter — clearly, accuracy is improved compared with tracking where phase is propagated using constant increments (Figure 4.4.4a).

Chapter 5

Evaluation

To evaluate the proposed method and its variants, discussed in the previous section, we use a dataset that provide, together with video sequences, the ground truth poses obtained by a motion capture system: the HumanEva-I dataset [SB06]. Using this kind of dataset has several advantages. The initialization can be performed automatically using the ground truth pose and the tracking error can be computed accurately and in a straightforward way. Moreover, the HumanEva-I dataset has become a standard dataset to evaluate human pose estimation techniques, which allows us to directly compare the results from our approach with state of the art methods.

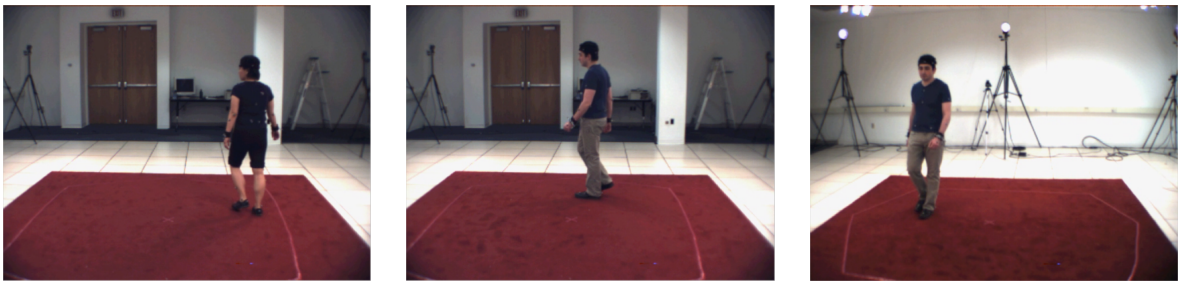
5.1 The HumanEva-I dataset

The HumanEva-I dataset [SB06] is composed of several sequences of three different subjects wearing natural clothing and performing different activities (e.g. walking, jogging, boxing). The sequences were captured at 60 Hz using 7 synchronized cameras (4 black/white cameras and 3 color) and ground truth was obtained using marker-based motion capture system. Since our appearance model is based on color information, we used the walking sequences captured by the color cameras. To learn the background statistics for silhouette extraction, as shown in Section 4.3.1, the dataset also contains a set of background images.

The walking sequences are composed of subjects walking in an elliptical path. Figure 5.1.1 shows some example frames of the walking dataset for two subjects.

5.2 Error metric

As proposed in [SBB10], the evaluation measure is defined by the distance of $N_m = 15$ virtual markers. The markers are placed in the body model joints, represented by blue circles in Figure 4.1.1, and another marker placed in the top of the head. Let $\{p_i(\mathbf{x})\}_{i=1}^{N_m}$ be the set of marker positions that are defined by the state $\mathbf{x}$, then the error is the mean Euclidean distance from all the markers of the ground truth pose $\mathbf{x}$ and the mean estimate $\bar{\mathbf{x}}$:



(a) Camera 'C1' and frame #300 (b) Camera 'C1' and frame #10 (c) Camera 'C2' and frame #10

Figure 5.1.1: Example frames in the HumanEva-I dataset of the walking sequence (subjects 'S1' and 'S2', respectively).

$$E(\mathbf{x}, \bar{\mathbf{x}}) = \frac{1}{N_m} \sum_{i=1}^{N_m} \|p_i(\mathbf{x}) - p_i(\bar{\mathbf{x}})\|. \tag{5.2.1}$$

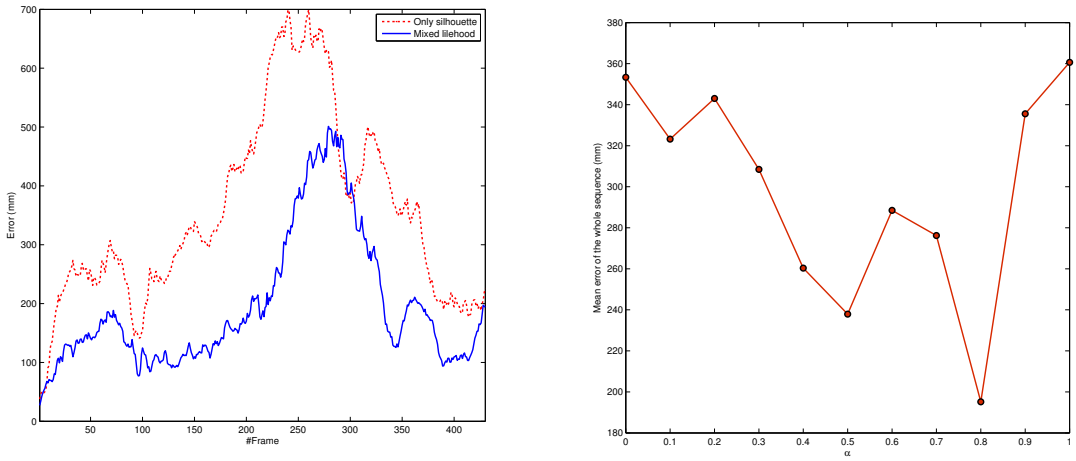
This metric is commonly used in literature, but two important aspects must be considered here. First, this metric does not directly include the global orientation of the body, such that, if two poses have the same relative angles and global positions, but opposed orientations, the computed error could be small even if, intuitively, it should be large. Second, the error can be large even if the tracking is visually accurate, i.e. the projections of the model w.r.t. the camera are aligned with the person's body. This is usually due to depth ambiguities in the pose estimation. As an alternative, several authors propose the use of a relative error measurement [BFH10, EL08] that compute the marker positions in body-centered coordinates, removing the global translation.

5.3 Likelihood experiments

Experiments were performed to test the tracking accuracy for different likelihood parameters. The tracker uses annealed filtering with 200 particles for each one of the 5 layers. The temporal prior employed is simple, it only adds a random noise in the particle states and enforces anatomical joint limits [SBB10]. The errors reported here are from the walking sequence of subject 'S2', and the camera 'C2' was used.

In the first experiment, the likelihood was based solely on silhouettes. The results are plotted in Figure 5.3.1a (dashed red curve). Large errors are mainly due to two reasons. First, after the legs are crossed, the tracking diverge from the correct value. This is caused by left-right inversions which cannot be avoided in tracking that is based only on silhouettes, as previously discussed. Second, most of the degrees of freedom are harder to recover when the person walks towards (or away from) the camera — the error increases around the 200th frame because of this. Overall, monocular trackers that are uniquely based on silhouettes are not able to properly recover the human motion in this sequence.

Similar results were obtained in [SBB10].



(a) Only silhouette vs. mixed ($\alpha = 0.8$)

(b) Mean errors for several values of α

Figure 5.3.1: Tracking errors for different likelihood parameters.

Next, the performance of the mixed likelihood, defined in equation (4.3.8), was measured. Appearance models were described by 16×16 histograms. The parameter α that controls the influence of each function was determined empirically ($\alpha = 0.8$). Figure 5.3.1a shows the error curve (in blue) for tracking using this mixed likelihood. It can be seen that the accuracy is significantly improved when appearance information is added.

We also tested the mixed likelihood for several values of α ranging from 0 to 1. For each one of these values, the mean error of the whole sequence was computed (see Figure 5.3.1b). When only silhouette ($\alpha = 0.0$) or only appearance ($\alpha = 1.0$) is used, the error is very large; good values for α lie between 0.4 and 0.8. This behavior was also observed when we changed the temporal prior to the prior based on principal component analysis.

5.3.1 Independent vs. correlated channels

The appearance model that we proposed here is similar to the model proposed by Gall et al. in [GBRS07]. However, it differs in two important aspects: (1) our approach is adapted for monocular tracking because it checks the visibility of body parts in the sampling step; and (2) in [GBRS07], the authors make the assumption, "for efficiency reasons", that the image channels are independent while we assume that the channels are correlated. We compared tracking results using appearance models consisting of 16×16 2D histograms (correlated channels) with models that use two one-dimensional histograms (the number of bins was $K = 64$ as proposed in [GBRS07]). Figure 5.3.2 shows the error curves. It is possible to see from the plot that if the image channels are taken to be correlated, as we propose, the appearance models are more discriminative, which improves tracking. Moreover, we found that the algorithm takes roughly the same computation time to build these two kinds of histograms.

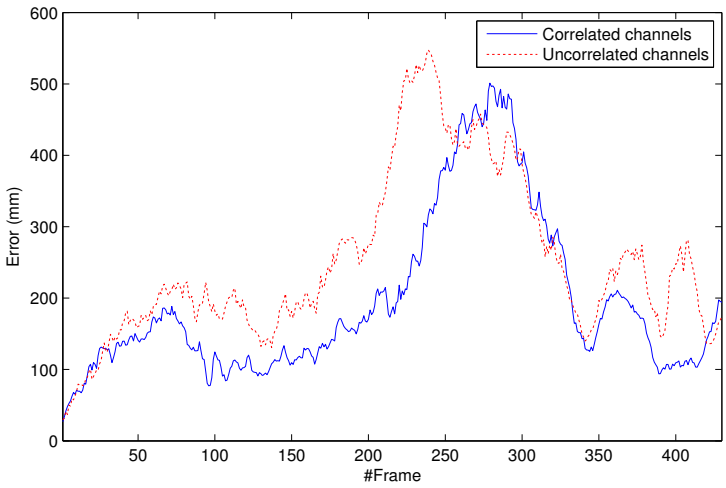


Figure 5.3.2: Tracking errors for histograms that assume that the image channels are correlated and histograms that assume independent channels.

5.4 Temporal prior

Finally, a set of experiments was performed to evaluate tracking with the proposed temporal prior (Section 4.4). In this case, we used a different camera, "C1". For the state propagation, the noise in the step parameter was defined as a Gaussian distribution with zero mean and standard deviation $\sigma = 0.01$. The likelihood combines silhouette and appearance information (with $\alpha = 0.5$). Figure 5.4.1 shows the error curve (in blue) for our proposed prior compared with prior proposed in [SBB10] (dashed red curve). As our method is a much stronger prior, less particles are needed: in our experiments we use only 50 particles and 3 layers. This is almost 7 times less than the number of particles used for the simple prior experiment (200 with 5 layers).

Using our prior, the error remains under 200mm until the tracker loses the subject around frame 250 (as opposed to the case where the simple prior is employed, in which the tracker diverges around frame 160). These results show that we are able to greatly reduce the number of particles while still maintaining low errors. Moreover, the pose trajectory that is recovered when our motion model is applied seems more realistic since the poses are reconstructed from a synthesized walking cycle.

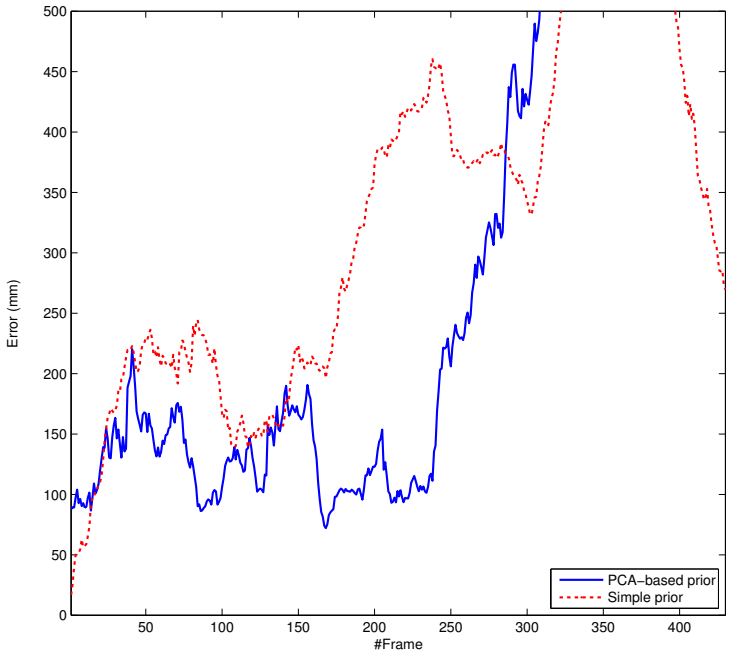


Figure 5.4.1: Error curves for tracking with our proposed prior (PCA-based) and the prior in [SBB10](Simple).

Chapter 6

Conclusion and further work

Because many applications would benefit from a robust solution, monocular motion tracking has been an active field of research in the last decade. Currently, a robust solution is only possible with strong activity-specific and subject-specific priors. In model-based approaches, such as our own, tracking is often formulated within a Bayesian framework. Therefore, a likelihood function and a temporal prior must be designed.

The two main contribution of this master thesis are:

1. A likelihood function that combines silhouette and appearance information. Our appearance model is similar to the one proposed by Gall et al. in [GBRS07], but with two important differences: (1) our approach is monocular and therefore we need to cope with self-occlusions. They are handled using visibility maps; and (2) to build histograms, we assume that image channels are correlated.

2. A temporal prior specific for walking sequences. Our method is based on PCA space reduction of walking cycles, first proposed by Sidenbladh et al. in [SBF00]. At initialization, the approach learns, from mocap data, a motion model that is able to synthesize walking cycles. However, unlike [SBF00], the walking speed is not included in the learning phase. Instead, we also track the walking speed. This makes the tracking more robust to sequences that differ greatly from the ones in the training dataset.

The algorithms have been experimentally evaluated on the HumanEva dataset [SB06]. Quantitative results have been presented which show that our methods can improve tracking. Additionally, for our temporal prior, we show that it can increase accuracy while greatly reducing the computation cost. Nonetheless, the proposed motion model have some limitations.

First, it is difficult to recover the walking phase when the person is moving towards (or away from) the camera. This is caused by the increase of depth ambiguities in such situations. To address this issue, we plan to include a dynamical noise in the estimation of the step parameter. In cases where the uncertainty is big, the noise will be small to prevent losing track of the person.

Second, even if the model is able to work for different walking speeds, it is still too limited with respect to the walking patterns used in the learning phase. For instance, if the person is walking and waving its hand at the same time, the current model cannot generalize to fully recover the arm configuration (unless a similar posture was in the training dataset). Further work will to try to generalize the model to work in such situations.

Finally, experiments suggest that recovering the person's global orientation is very important for tracking, i.e. when it is not recovered properly, the system ends up losing tracking in the subsequent frames. We are currently studying some alternatives to robustly estimate the global orientation separately from tracking and then, using the result to guide the search in the particle filter.

Bibliography

[ARS09] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1014–1021, 2009.

[ARS10] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[AT06] Ankur Agarwal and Bill Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:44–58, January 2006.

[BB06] Alexandru O. Balan and Michael J. Black. An adaptive appearance model approach for model-based articulated object tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 758–765, 2006.

[BB09] Jan Bandouch and Michael Beetz. Tracking humans interacting with the environment using efficient hierarchical sampling and layered observation models. In *IEEE International Workshop on Human-Computer Interaction*, 2009.

[BFH10] Marcus A. Brubaker, David J. Fleet, and Aaron Hertzmann. Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, 87:1–8, 2010.

[BMP02] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[BSKM08] Liefeng Bo, Cristian Sminchisescu, Atul Kanaujia, and Dimitris Metaxas. Fast algorithms for large scale conditional 3d prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[DLC08] John Darby, Baihua Li, and Nicholas Costen. Tracking a walking person using activity-guided annealed particle filtering. In *IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–6, 2008.

[DLF06] Miodrag Dimitrijevic, Vincent Lepetit, and Pascal Fua. Human body pose detection using bayesian spatio-temporal templates. *Computer Vision and Image Understanding*, 104:127–139, 2006.

[DR05] Jonathan Deutscher and Ian Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61:185–205, 2005.

[EL08] Ahmed Elgammal and Chan-Su Lee. The role of manifold learning in human motion analysis. In *Human Motion*, volume 36 of *Computational Imaging and Vision*, pages 25–56. Springer Netherlands, 2008.

[EL09] Ahmed Elgammal and Chan-Su Lee. Tracking people on a torus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:520–538, 2009.

[FAR06] David A. Forsyth, Okan Arikan, and Deva Ramanan. Computational studies of human motion: Part 1, tracking and motion synthesis. In *Foundations and Trends in Computer Graphics and Vision*, 2006.

[FDLF10] Andrea Fossati, Miodrag Dimitrijevic, Vincent Lepetit, and Pascal Fua. From canonical poses to 3d motion capture using a single camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1165–1181, 2010.

[FH05] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79, 2005.

[GBRS07] Juergen Gall, Thomas Brox, Bodo Rosenhahn, and Hans-Peter Seidel. Global stochastic optimization for robust and accurate human motion capture. Technical Report MPI-I-2007-4-008, Department 4: Computer Graphics, Max-Planck Institute für Informatik, 2007.

[GD96] Dariu M. Gavrila and Larry S. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *IEEE Conference on Conference on Computer Vision and Pattern Recognition*, 1996.

[How04] Nicholas R. Howe. Silhouette lookup for automatic pose tracking. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*, volume 1, pages 15–22, 2004.

[How05] Nicholas R. Howe. Flow lookup and biological motion perception. In *IEEE International Conference on Image Processing*, 2005.

[JBY96] Shanon X. Ju, Michael J. Black, and Yaser Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, 1996.

[JFEM03] Allan D. Jepson, David J. Fleet, and Thomas F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1296–1311, 2003.

[KSM07] Atul Kanaujia, Cristian Sminchisescu, and Dimitris Metaxas. Semi-Supervised Hierarchical Models for 3D Human Pose Reconstruction. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.

[LE10] Chan-Su Lee and Ahmed Elgammal. Coupled visual and kinematic manifold models for tracking. *International Journal of Computer Vision*, 87:118–139, 2010.

[LF05] Vincent Lepetit and Pascal Fua. Monocular model-based 3d tracking of rigid objects. *Foundations and Trends in Computer Graphics and Vision*, 1:1–89, 2005.

[LH04] Xiangyang Lan and Daniel P. Huttenlocher. A unified spatio-temporal articulated model for tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:722–729, 2004.

[LhYST06] Rui Li, Ming hsuan Yang, Stan Sclaroff, and T.-P. Tian. Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers. In *European Conference on Computer Vision*, pages 137–150, 2006.

[MHK06] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126, 2006.

[MM06] Greg Mori and Jitendra Malik. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1052–1062, 2006.

[NFC07] Ramanan Navaratnam, Andrew W. Fitzgibbon, and Roberto Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[PF01] Ralf Plänkers and Pascal Fua. Articulated soft objects for video-based body modeling. In *IEEE International Conference on Computer Vision*, volume 1, pages 394–401, 2001.

[Pop07a] Ronald Poppe. Evaluating example-based pose estimation: Experiments on the humaneva sets. Technical Report TR-CTIT-07-72, Centre for Telematics and Information Technology University of Twente, 2007.

[Pop07b] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108:4–18, 2007.

[RMR06] Timothy J. Roberts, Stephen J. McKenna, and Ian W. Ricketts. Human tracking using 3d surface colour distributions. *Image and Vision Computing*, 24(12):1332–1342, 2006.

[SB06] Leonid Sigal and Michael Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University, 2006.

[SB10] Leonid Sigal and Michael Black. Guest editorial: State of the art in image- and video-based human pose and motion estimation. *International Journal of Computer Vision*, 87:1–3, 2010.

[SBB10] Leonid Sigal, Alexandru Balan, and Michael Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87:4–27, 2010.

[SBF00] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 702–718, 2000.

[Smi06] Cristian Sminchisescu. 3d human motion analysis in monocular video techniques and challenges. *IEEE Conference on Advanced Video and Signal Based Surveillance*, 1:76, 2006.

[ST03] Cristian Sminchisescu and Bill Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 22(6):371–391, 2003.

[SVD03] Gregory Shakhnarovich, Paul Viola, and Trevor Darrell. Fast pose estimation with parameter-sensitive hashing. *IEEE International Conference on Computer Vision*, 2:750–757, 2003.

[TLS05] Tai-Peng Tian, Rui Li, and Stan Sclaroff. Tracking human body pose on a learned smooth space. Technical Report BUCS-TR-2005-029, Boston University, Computer Science Department, 2005.

[UF04] Raquel Urtasun and Pascal Fua. 3d human body tracking using deterministic temporal motion models. In *European Conference on Computer Vision*, pages 92–106, 2004.

[UFHF05] Raquel Urtasun, David J. Fleet, Aaron Hertzmann, and Pascal Fua. Priors for people tracking from small training sets. *IEEE International Conference on Computer Vision*, 1:403–410, 2005.

[VSJ08] Marek Vondrak, Leonid Sigal, and Odest C. Jenkins. Physical simulation for probabilistic motion tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[WN99] S. Wachter and H.-H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74:174–192, 1999.

[WR06] Ping Wang and James M. Rehg. A modular approach to the analysis and evaluation of particle filters for figure tracking. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 790–797, 2006.