11

Contours, Optic Flow, and Prior Knowledge: Cues for Capturing 3D Human Motion in Videos

Thomas Brox[1], Bodo Rosenhahn[2], and Daniel Cremers[1]

[1] CVPR Group, University of Bonn
Römerstr. 164, 53117 Bonn, Germany

[2] MPI for Computer Science,
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany

Summary. Human 3D motion tracking from video is an emerging research field with many applications demanding highly detailed results. This chapter surveys a high quality generative method, which employs the person's silhouette extracted from one or multiple camera views for fitting an a-priori given 3D body surface model. A coupling between pose estimation and contour extraction allows for reliable tracking in cluttered scenes without the need of a static background. The optic flow computed between two successive frames is used for pose prediction. It improves the quality of tracking in case of fast motion and/or low frame rates. In order to cope with unreliable or insufficient data, the framework is further extended by the use of prior knowledge on static joint angle configurations.

11.1 Introduction

Tracking of humans in videos is a popular research field with numerous applications ranging from automated surveillance to sports movement analysis. Depending on applications and the quality of video data, there are different approaches with different objectives. In many people tracking methods, for instance, only the position of a person in the image or a region of interest is sought. Extracting more detailed information is often either not necessary or very difficult due to image resolution.
In contrast to such model-free tracking methods, the present chapter is concerned with the detailed fitting of a given 3D model to video data. The model consists of the body surface and a skeleton that contains predefined joints [26, 8]. Given the video data from one or more calibrated cameras, one is interested in estimating the person's 3D pose and the joint angles. This way, the tracking becomes an extended 2D-3D pose estimation problem, where additionally to the person's rigid body motion one is interested in some restricted kind of deformation, namely the motion of limbs.
Applications of this kind of tracking are sports movement and clinical analysis, as well as the recording of motion patterns for animations in computer graphics. The state-of-the-art for capturing human motion is currently defined by large industrial

motion capture systems with often more than 20 cameras. These systems make use of markers attached to the person's body in order to allow for a fast and reliable image processing. Often the reliability of the results is further improved by manually controlling the matching of markers. Such systems are described in Chapter 16.

While results of marker-based motion capturing systems are very trustworthy, markers need to be attached, which is sometimes not convenient. Moreover the manual supervision of marker matching can be very laborious. For these reasons, one is interested in marker-less motion capturing, using the appearance of the person as a natural marker.

While markers in marker-based systems have been designed for being easy to identify, finding correct correspondences of points in marker-free systems is not as simple. A sensible selection of the right feature to be tracked is important. A typical way to establish point correspondences is to concentrate on distinctive patches in the image and to track these patches, for instance, with the KLT tracker [56] or a tracker based on the so-called SIFT descriptor [37]. However, patch based tracking typically only works reliably if the appearance of the person contains sufficiently textured areas.

An alternative feature, particularly for tracking people with non-textured clothing, is the silhouette of the person. Early approaches have been based on edge detectors and have tried to fit the 3D model to dominant edges [26]. Since image edges are not solely due to the person's silhouette, the most relevant problem of such approaches is their tendency to get stuck in local optima. Sophisticated optimization techniques have been suggested in order to attenuate this problem [62]. Nowadays, silhouette based tracking usually relies on background subtraction. Assuming both a static camera and a static background, the difference between the current image and the background image efficiently yields the foreground region. Apart from the restrictive assumptions, this approach works very well and is frequently employed for human tracking [25, 60, 1].

In [53] a contour-based method to 3D pose tracking has been suggested that does not impose such strict assumptions on the scene. Instead, it demands dissimilarity of the foreground and background region, which is a typical assumption in image segmentation. In order to deal with realistic scenarios where persons may also wear non-uniform cloths and the background is cluttered, the dissimilarity is defined in a texture feature space, and instead of homogeneous regions, the model expects only *locally* homogeneous regions. The main difference to other pose tracking methods, however, is the coupling between feature extraction and estimation of the pose parameters. In a joint optimization one seeks the pose parameters that lead to the best fit of the contour in the image. Vice-versa, one seeks a segmentation that fits the image data *and* resembles the projected surface model. Due to this coupling, the contour extraction is much more reliable than in a conventional two-step approach, where the contour is computed independently from the pose estimation task. We will survey the method in Section 11.3.

Although this way of integrating contours into pose estimation is more robust than the edge-based approach, it is still a local optimization method that can get stuck in local optima in case of fast motion. To alleviate these effects, it is common practice in tracking to predict the pose of the tracked object in the coming frame. A prediction is usually computed by simply extrapolating the motion between the last two frames to the next frame. In a more subtle way, learning based approaches incorporate auto-regressive models or nonlinear subspace methods based on training sequences to accomplish this task [57, 25, 1, 17, 65].

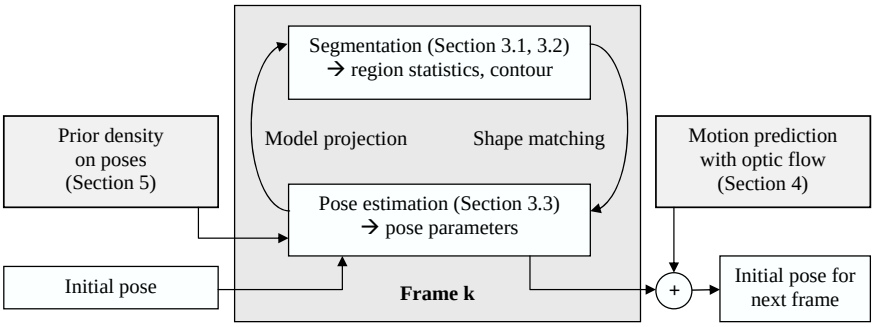


Fig. 11.1. System overview: the core is the coupled contour extraction and pose estimation. The motion between frames is predicted by optic flow in order to ensure a close initialization in the next frame. Pose configurations are constrained by a prior density estimated from training samples.

Another possibility to predict the pose parameters in the next frame is by the optic flow. Optic flow based tracking is similar to patch-based tracking, though instead of patches one tries to match single points under certain smoothness assumptions. With a reliable optic flow estimation method, one can predict rather large displacements [10]. In combination with the contour-based method, one obtains a system that can handle fast motions and is free from error accumulation, which is a severe problem for optic flow or patch-based tracking. A similar concept has been presented earlier in [21] and [38] in combination with edges instead of contours and different optic flow estimation techniques. The pose prediction by means of optic flow and how the flow can be efficiently computed is explained in Section 11.4.
Since 3D human tracking is generally an ill-posed problem with many solutions explaining the same data, methods suffer enormously from unreliable data. Therefore, in recent years, it has become more and more popular to exploit prior assumptions about typical human poses and motion patterns [59, 65, 11]. In Section 11.5 it will be described how the tracking model can be constrained to prefer solutions that are close to familiar poses. The impact of such a constraint regarding the robustness of the technique to disturbed image data is remarkable. See also Chapter 2 and Chapter 8 for learning techniques in human tracking.
Although the tracking system described in this chapter comprises many advanced methods, there is still much room for extensions or alternative approaches. In Section 11.6 we discuss issues such as running times, auto-initialization, dynamical pose priors, and cloth tracking including cursors to other chapters in this book or to seminal works in the literature. A brief summary of the chapter is given in Section 11.7.

11.2 Human Motion Representation with Twists and Kinematic Chains

A human body can be modeled quite well by means of a kinematic chain. A kinematic chain is a set of (usually rigid) bodies interconnected by joints. For example, an arm consists of an upper and lower arm segment and a hand, with the shoulders,

elbow and wrist as interconnecting joints. For a proper representation of joints and transformations along kinematic chains in the human tracking method, we use the exponential representation of rigid body motions [42], as suggested in [7, 8].
Every 3D rigid motion can be represented in exponential form

$$\boldsymbol{M} = \exp(\theta\hat{\xi}) = \exp\begin{pmatrix} \hat{\omega} & \boldsymbol{v} \\ 0_{3\times 1} & 0 \end{pmatrix} \tag{11.1}$$

where $\theta\hat{\xi}$ is the matrix representation of a twist $\xi \in se(3) = \{(\boldsymbol{v}, \hat{\omega}) | \boldsymbol{v} \in \mathbb{R}^3, \hat{\omega} \in so(3)\}$, with $so(3) = \{\boldsymbol{A} \in \mathbb{R}^{3\times 3} | \boldsymbol{A} = -\boldsymbol{A}^T\}$. The Lie algebra $so(3)$ is the tangential space of the 3D rotations at the origin. Its elements are (scaled) rotation axes, which can either be represented as a 3D vector

$$\theta\omega = \theta \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}, \text{ with } \|\omega\|_2 = 1 \tag{11.2}$$

or as a skew symmetric matrix

$$\theta\hat{\omega} = \theta \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{11.3}$$

In fact, $\boldsymbol{M}$ is an element of the Lie group $SE(3)$, known as the group of direct affine isometries. A main result of Lie theory is that to each Lie group there exists a Lie algebra which can be found in its tangential space by derivation and evaluation at its origin. Elements of the Lie algebra therefore correspond to infinitesimal group transformations. See [42] for more details. The corresponding Lie algebra to $SE(3)$ is denoted as $se(3)$.
A twist contains six parameters and can be scaled to $\theta\xi$ with a unit vector ω. The parameter $\theta \in \mathbb{R}$ corresponds to the motion velocity (i.e., the rotation velocity and pitch). The one-parameter subgroup $\Phi_{\hat{\xi}}(\theta) = \exp(\theta\hat{\xi})$ generated by this twist corresponds to a screw motion around an axis in space. The six twist components can either be represented as a 6D vector

$$\begin{aligned} \theta\xi &= \theta(\omega_1, \omega_2, \omega_3, v_1, v_2, v_3)^T \\ \text{with } \|\omega\|_2 &= \|(\omega_1, \omega_2, \omega_3)^T\|_2 = 1, \end{aligned} \tag{11.4}$$

or as a 4×4 matrix

$$\theta\hat{\xi} = \theta \begin{pmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{11.5}$$

To reconstruct a group action $\boldsymbol{M} \in SE(3)$ from a given twist, the exponential function $\exp(\theta\hat{\xi}) = \sum_{k=0}^{\infty} \frac{(\theta\hat{\xi})^k}{k!} = \boldsymbol{M} \in SE(3)$ must be computed. This can be done efficiently by using the Rodriguez formula [42].
In this framework, joints are expressed as special screws with no pitch. They have the form $\theta_j\hat{\xi}_j$ with known $\hat{\xi}_j$ (the location of the rotation axes as part of the model representation) and unknown joint angle θ_j. A point on the jth joint can be represented as consecutive evaluation of exponential functions of all involved joints,

$$X_i' = \exp(\theta\hat{\xi}_{RBM})(\exp(\theta_1\hat{\xi_1})\dots\exp(\theta_j\hat{\xi_j})X_i) \tag{11.6}$$

The human body motion is then defined by a parameter vector $\xi := (\xi_{RBM}, \Theta)$ that consists of the 6 parameters for the global twist ξ_{RBM} (3D rotation and translation) and the joint angles $\Theta := (\theta_1, \dots, \theta_N)$.

11.3 Contour-based Pose Estimation

In this section, we survey the coupled extraction of the contour and the estimation of the pose parameters by means of this contour. For better understanding we start in Section 11.3.1 with the simple segmentation case that is not yet related to the pose parameters. In the end, the idea is to find pose parameters in such a way that the projected surface leads to a region that is homogeneous according to a certain statistical model. The statical region model will be explained and motivated in Section 11.3.2. In Section 11.3.3 we then bend the bow to pose estimation by introducing the human model as a 3D shape prior into the segmentation functional. This leads to a matching of 2D shapes. From the point correspondences of this matching, one can derive 2D-3D correspondences and finally estimate the pose parameters from these.

11.3.1 Contour Extraction with Level Sets

Level set representation of contours. The contour extraction is based on variational image segmentation with level sets [23, 44], in particular region-based active contours [15, 64, 46, 20]. Level set formulations of the image segmentation problem have several advantages. One is the convenient embedding of a 1D curve into a 2D, image-like structure. This allows for a convenient and sound interaction between constraints that are imposed on the contour itself and constraints that act on the regions separated by the contour. Moreover, the level set representation yields the inherent capability to model topological changes. This can be an important issue, for instance, when the person is partially occluded and the region is hence split into two parts, or if the pose of legs or arms leads to topological changes of the background. In the prominent case of a segmentation into foreground and background, a level set function $\Phi \in \Omega \mapsto \mathbb{R}$ splits the image domain Ω into two regions Ω_1 and Ω_2, with $\Phi(x) > 0$ if $x \in \Omega_1$ and $\Phi(x) < 0$ if $x \in \Omega_2$. The zero-level line thus marks the boundary between both regions, i.e., it represents the person's silhouette that is sought to be extracted.

Optimality criteria and corresponding energy functional. As optimality criteria for the contour we want the data within one region to be similar and the length of the contour to be as small as possible. Later in Section 11.3.3 we will add similarity to the projected surface model as a further criterion. The model assumptions can be expressed by the following energy functional [66, 15]:

$$E(\Phi) = -\int_\Omega \big(H(\Phi)\log p_1 + (1-H(\Phi))\log p_2\big)\,dx + \nu\int_\Omega |\nabla H(\Phi)|\,dx \tag{11.7}$$

where $\nu > 0$ is parameter that weights the similarity against the length constraint, and $H(s)$ is a regularized Heaviside function with $\lim_{s\to-\infty} H(s) = 0$,

$\lim_{s\to\infty} H(s) = 1$, and $H(0) = 0.5$. It indicates to which region a pixel belongs. Chan and Vese suggested two alternative functions in [15]. The particular choice of H is not decisive. We use the error function, which has the convenient property that its derivative is the Gaussian function.
Minimizing the first two terms in (11.7) maximizes the likelihood given the probability densities p_1 and p_2 of values in Ω_1 and Ω_2, respectively. The third term penalizes the length of the contour, what can be interpreted as a log-prior on the contour preferring smooth contours. Therefore, minimizing (11.7) maximizes the total a-posteriori probability of all pixel assignments.

Minimization by gradient descent. For energy minimization one can apply a gradient descent. The Euler-Lagrange equation of (11.7) leads to the following update equation[3]:

$$\partial_t \Phi = H'(\Phi) \left(\log \frac{p_1}{p_2} + \nu \, \nabla^\top \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right) \tag{11.8}$$

where $H'(s)$ is the derivative of $H(s)$ with respect to its argument. Applying this evolution equation to some initialization Φ^0, and given the probability densities p_i, which are defined in the next section, the contour converges to the next local minimum for the numerical evolution parameter $t \to \infty$.

11.3.2 Statistical Region Models

An important factor for the contour extraction process is how the probability densities $p_i : \mathbb{R} \to [0, 1]$ are modeled. This model determines what is considered similar or dissimilar. There is on one hand the choice of the feature space, e.g. gray value, RGB, texture, etc., and on the other hand the parametrization of the probability density function.

Texture features. Since uniformly colored cloths without texture are in general not realistic, we adopt the texture feature space proposed in [12]. It comprises $M = 5$ feature channels I_j for gray scale images, and $M = 7$ channels if color is available. The color channels are considered in the CIELAB color space. Additionally to gray value and color, the texture features in [12] encode the texture magnitude, orientation, and scale, i.e., they provide basically the same information as the frequently used responses of Gabor filters[24]. However, the representation is less redundant, so 4 feature channels substitute 12-64 Gabor responses. Alternatively, Gabor features can be used at the cost of larger computation times. In case of people wearing uniform cloths and the background also being more or less homogeneous, one can also work merely with the gray value or color in order to increase computation speed.

Channel independence. The probability densities of the M feature channels are assumed to be independent, thus the total probability density can be composed of the densities of the separate channels:

$$p_i = \prod_{j=1}^{M} p_{ij}(I_j) \qquad i = 1, 2. \tag{11.9}$$

[3] As the probability densities in general also depend on the contour there may appear additional terms depending on the statistical model. For global Gaussian densities, however, the terms are zero, and for other models they have very little influence on the result, so they are usually neglected.

Though assuming channel independence is merely an approximation, it keeps the density model tractable. This is important, as the densities have to be estimated from a limited amount of image data.

Density models of increasing complexity. There are various possibilities how to model channel densities. In [15] a simple piecewise constant region model is suggested, which corresponds to a Gaussian density with fixed standard deviation. In order to admit different variations in the regions, it is advisable to use at least a full Gaussian density [66], a generalized Laplacian [28], or a Parzen estimate [54, 32]. While more complex density models can represent more general distributions, they also imply the estimation of more parameters which generally leads to a more complex objective function.

Local densities. Nevertheless, for the task of human tracking, we advocate the use of a more complex region model, in particular a Gaussian density that is estimated using only values in a local neighborhood of a point instead of values from the whole region. Consequently, the probability density is no longer fixed for one region but varies with the position. Local densities have been proposed in [31, 53]. Segmentation with such densities has been shown to be closely related to the piecewise smooth Mumford-Shah model [41, 13]. Formally, the density is modeled as

$$p_{ij}(s,x) = \frac{1}{\sqrt{2\pi}\sigma_{ij}(x)} \exp\left(\frac{(s-\mu_{ij}(x))^2}{2\sigma_{ij}(x)^2}\right). \tag{11.10}$$

The parameters $\mu_{ij}(x)$ and $\sigma_{ij}(x)$ are computed in a local Gaussian neighborhood K_ρ around x by:

$$\mu_{ij}(x) = \frac{\int_{\Omega_i} K_\rho(\zeta - x) I_j(\zeta)\, d\zeta}{\int_{\Omega_i} K_\rho(\zeta - x)\, d\zeta} \qquad \sigma_{ij}(x) = \frac{\int_{\Omega_i} K_\rho(\zeta - x)(I_j(\zeta) - \mu_{ij}(x))^2\, d\zeta}{\int_{\Omega_i} K_\rho(\zeta - x)\, d\zeta} \tag{11.11}$$

where ρ denotes the standard deviation of the Gaussian window. In order to have enough data to obtain reliable estimates for the parameters $\mu_{ij}(x)$ and $\sigma_{ij}(x)$, we choose $\rho = 12$.

Taking advantage of local dissimilarity of foreground and background. The idea behind the local density model is the following: in realistic scenarios, the foreground and background regions are rarely globally dissimilar. For instance, the head may have a different color than the shirt or the trousers. If the same colors also appear in the background, it is impossible to accurately distinguish foreground and background by means of a standard global region distribution. Locally, however, foreground and background can be easily distinguished. Although the local density model is too complex to detect the desired contour in an image without a good contour initialization and further restrictions on the contour's shape, we are in a tracking scenario, i.e., the result from the previous frame always provides a rather good initialization. Moreover, in the next section a shape constraint is imposed on the contour that keeps it close to the projection of the surface model. Also note, that we still have a statistical region based model, which yields considerably less local optima than previous edge-based techniques. The results in Figure 11.2 and Figure 11.4 show that local region statistics provide more accurate contours and thus allow for a more reliable estimate of the 3D pose.

Optimization with EM. Estimating both the probability densities p_{ij} and the region contour works according to the *expectation-maximization principle* [22, 39]. Having the level set function initialized with some partitioning Φ^0, the probability densities in these regions can be approximated. With the probability densities, on the other hand, one can compute an update on the contour according to (11.8), leading to a further update of the probability densities, and so on. In order to attenuate the dependency on the initialization, one can apply a continuation method in a coarse-to-fine manner [6].

11.3.3 Coupled estimation of contour and pose parameters

Bayesian inference. So far, only the person's silhouette in the image has been estimated, yet actually we are interested in the person's pose parameters. They can be estimated from the contour in the image, but also vice-versa the surface model with given pose parameters can help to determine this contour. In a Bayesian setting this joint estimation problem can be written as the maximization of

$$p(\Phi, \xi | I) = \frac{p(I|\Phi, \xi) p(\Phi|\xi) p(\xi)}{p(I)} \tag{11.12}$$

where Φ indicates the contour given as a level set function, ξ the set of pose parameters, and I the given image(s). This formula imposes a shape prior on Φ given the pose parameters, and it imposes a prior on the pose parameters. For the moment we will use a uniform prior for $p(\xi)$, effectively ignoring this factor, but we will come back to this prior later in Section 11.5.

Joint energy minimization problem. Assuming that the appearance in the image is completely determined by the contour with no further (hidden) dependence on ξ, we can set $p(I|\Phi, \xi) \equiv p(I|\Phi)$. Minimizing the negative logarithm of (11.12) then leads to the following energy minimization problem:

$$\begin{aligned} E(\Phi, \theta\xi) &= -\log p(\Phi, \xi | I) \\ &= -\int_\Omega \big(H(\Phi) \log p_1 + (1 - H(\Phi)) \log p_2\big)\, dx + \nu \int_\Omega |\nabla H(\Phi)|\, dx \\ &\quad + \underbrace{\lambda \int_\Omega (\Phi - \Phi_0(\xi))^2\, dx}_{\text{Shape}} + \text{const.} \end{aligned} \tag{11.13}$$

One recognizes the energy from (11.7) with an additional term that imposes the shape constraint on the contour and relates at the same time the contour to the sought pose parameters. The parameter $\lambda \geq 0$ introduced here determines the variability of the estimated contour Φ from the projected surface model Φ_0. Φ_0 is again a level set function and it is obtained by projecting the surface to the image plane (by means of the known projection matrice) and by applying a signed distance transform to the resulting shape. The signed distance transform assigns each point x of Φ_0 the Euclidean distance of x to the closest contour point. Points inside the projected region get positive sign, points outside this region, get negative sign.

Alternating optimization. In order to minimize (11.13) for both the contour and the pose parameters, an alternating scheme is proposed. First, the pose parameters

are kept fixed and the energy is minimized with respect to the contour. Afterwards, the contour is retained and one optimizes the energy for the pose parameters. In the tracking scenario, with the initial pose being already close to the desired solution, only few (2-5) iterations are sufficient for convergence.

Optimization with respect to the contour. Since the shape term is modeled in the image domain, minimization of (11.13) with respect to Φ is straightforward and leads to the gradient descent equation

$$\partial_t \Phi = H'(\Phi) \left(\log \frac{p_1}{p_2} + \nu \, \nabla^\top \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right) + \, 2\lambda \, (\Phi_0(\xi) - \Phi). \tag{11.14}$$

One can observe that the shape term pushes Φ towards the projected surface model, while on the other hand, Φ is still influenced by the image data ensuring homogeneous regions according to the statistical region model.

Optimization with respect to the pose parameters. Optimization with respect to the pose parameters needs more care, since the interaction of the model with the contour in the image involves a projection. At the same time, the variation of the projected shape with a certain 3D transformation is quite complex. Principally, the 3D transformation can be estimated from a set of 2D-3D point correspondences in a least squares setting, as will be explained later in this section. Since we know how 2D points in Φ_0 correspond to 3D points on the surface (Φ_0 was constructed by projecting these 3D points), 2D-3D point correspondences can be established by matching points of the two 2D shapes Φ and Φ_0.

Shape matching. For minimizing the shape term in (11.13) with respect to the pose parameters, we look for a transformation in 2D that can account for the projections of all permitted transformations in 3D. Therefore, we choose a nonparametric transformation, in particular a smooth displacement field $\mathbf{w}(\mathbf{x}) := (u(\mathbf{x}), v(\mathbf{x}))$ and formulate the shape term as

$$E(u, v) = \int_\Omega (\Phi(\mathbf{x}) - \Phi_0(\mathbf{x} + \mathbf{w}))^2 + \alpha(|\nabla u|^2 + |\nabla v|^2) \, \mathbf{dx}. \tag{11.15}$$

where $\alpha \geq 0$ is a regularization parameter that steers the influence of the regularization relative to the matching criterion. The considered transformation is very general and, hence, can handle the projected transformations in 3D. The regularization ensures a smooth displacement field, which corresponds to penalizing shape deformations. Furthermore, it makes the originally ill-posed matching problem well-posed.

Optic flow estimation problem. A closer look at (11.15) reveals strong connections to optic flow estimation. In fact, the energy is a nonlinear version of the Horn-Schunck functional in [29]. Consequently, the matching problem can be solved using a numerical scheme known from optic flow estimation. We will investigate this scheme more closely in Section 11.4.

Alternative matching via ICP. Alternatively, one can match the two shapes by an iterated closest point (ICP) algorithm [4]. As Φ and Φ_0 are both Euclidean distance images, this is closely related to minimization of (11.15) for $\alpha \to 0$. In [51] it has been shown empirically that the combination of point correspondences from both methods is beneficial for pose estimation.

Inverse projection and Plücker lines. After matching the 2D contours, the remaining task is to estimate from the nonparametric 2D transformation a 3D transformation parameterized by the sought vector $\xi = (\xi_{RBM}, \Theta)$. For this purpose, the 2D points are changed into 3D entities. For the points in Φ this means that their projection rays need to be constructed. A projection ray contains all 3D points that, when projected to the image plane, yield a zero distance to the contour point there. Hence, for minimizing the distance in the image plane, one can as well minimize the distance between the model points and the rays reconstructed from the corresponding points.

There exist different ways to represent projection rays. As we have to minimize distances between correspondences, it is advantageous to use an implicit representation for a 3D line. It allows instantaneously to determine the distance between a point and a line.

An implicit representation of projection rays is by means of so-called *Plücker lines*[55, 63]. A Plücker line $\boldsymbol{L} = (\boldsymbol{n}, \boldsymbol{m})$ is given as a unit vector $\boldsymbol{n}$ and a moment $\boldsymbol{m}$ with $\boldsymbol{m} = \boldsymbol{x} \times \boldsymbol{n}$ for a given point $\boldsymbol{x}$ on the line. The incidence of a point $\boldsymbol{x}$ on a line $\boldsymbol{L} = (\boldsymbol{n}, \boldsymbol{m})$ can then be expressed as

$$\boldsymbol{x} \times \boldsymbol{n} - \boldsymbol{m} = 0. \tag{11.16}$$

Parameter estimation by nonlinear least squares. This equation provides an error vector and we seek the transformation $\xi = (\xi_{RBM}, \Theta)$ that minimizes the norm of this vector over all correspondences. For $j = \mathcal{J}(x_i)$ being the joint index of a model point x_i, the error to be minimized can be expressed as

$$\sum_i \| \Pi \Big(\exp(\hat{\xi}_{RBM}) \exp(\theta_1 \hat{\xi_1}) \dots \exp(\theta_{\mathcal{J}(x_i)} \hat{\xi}_{\mathcal{J}(x_i)}) \boldsymbol{x}_i \Big) \times \boldsymbol{n}_i - \boldsymbol{m}_i \|_2^2, \tag{11.17}$$

where Π is the projection of the homogeneous 4D vector to a 3D vector by neglecting the homogeneous component (which is 1), and the symbol $\times$ denotes the cross product.

Linearization. The minimization problem in (11.17) is a least squares problem. Unfortunately, however, the equations are non-quadratic due to the exponential form of the transformation matrices. For this reason, the transformation matrix is linearized and the pose estimation procedure is iterated, i.e., the nonlinear problem is decomposed into a sequence of linear problems. This is achieved by

$$\exp(\theta \hat{\xi}) = \sum_{k=0}^{\infty} \frac{(\theta \hat{\xi})^k}{k!} \approx \boldsymbol{I} + \theta \hat{\xi} \tag{11.18}$$

with $\boldsymbol{I}$ as identity matrix.

This results in

$$((\boldsymbol{I} + \theta \hat{\xi} + \theta_1 \hat{\xi_1} \dots + \theta_{\mathcal{J}(x_i)} \hat{\xi}_{\mathcal{J}(x_i)}) X_i)_{3\times 1} \times n_i - m_i = 0 \tag{11.19}$$

with the unknown pose parameters ξ acting as linear components. This equation can be reordered into the form $\boldsymbol{A}(\theta \xi_{RBM}, \theta_1 \dots \theta_N)^T = \boldsymbol{b}$. Collecting a set of such equations (each is of rank two) leads to an over-determined linear system of equations, which can be solved using, for example, the Householder algorithm. The Rodriguez

formula can be applied to reconstruct the group action from the estimated parameter vector ξ. The 3D points can be transformed and the process is iterated until it converges.

Multiple camera views. The method can easily be extended to make use of multiple camera views if all cameras are calibrated to the same world coordinate system. The point correspondences, obtained by projecting the surface model to all images and extracting contours there, can be combined in a joint system of equations. The solution of this system is the least squares fit of the model to the contours in all images. Due to the coupling of contour and pose estimation, also the contour extraction can benefit from the multi-view setting. This is demonstrated in the comparison depicted in Figure 11.2 and Figure 11.3.

11.4 Optic Flow for Motion Prediction

The contour-based tracking explained in the previous section demands a pose initialization that is close enough to obtain reasonable estimates of the region statistics. For high frame rates and reasonably slow motion, the result from the previous frame is a sufficiently good initialization. For very fast motion or small frame rates, however, it may happen that limbs have moved too far and the method is not able to recapture them starting with the result from the previous frame. This problem is illustrated in the first row of Figure 11.5.
A remedy is to improve the initialization by predicting the pose parameters in the successive frame. The most simple approach is to compute the velocity from the results in the last two frames and to assume that the velocity stays constant. However, it is obvious that this assumption is not satisfied at all times and can lead to predictions that are even much worse than the initialization with the latest result. Auto-regressive models are much more reliable. They predict the new state from previous ones by means of a parametric model estimated from a set of training data.

Pose estimation from optic flow. In this chapter, we focus on an image-driven prediction by means of optic flow. We assume that the pose has been correctly estimated in frame t, and we are now interested in a prediction of the pose in frame $t+1$ given the images in t and $t+1$. For this prediction, we need to compute the optic flow, which provides 2D-2D correspondences between points in the images. As the 2D-3D correspondences in frame t are known, we obtain a set of 2D-3D point correspondences between the new frame $t+1$ and the model. From these, the pose of the model in frame $t+1$ can be computed by solving a sequence of linear systems, as described by equation (11.19) in the previous section.

Accumulation of errors. The inherent assumption of knowing the correct pose of the model in frame t is in fact not exactly satisfied. In practice, there will be inaccuracies in the estimated pose. This results in the accumulation of errors when using only model-free schemes based on the optic flow or feature tracking. However, the contour-based pose estimation from the previous section, which directly derives correspondences between the image and the model, does not suffer from this problem. It is able to correct errors from the previous frame or from the estimated optic flow.

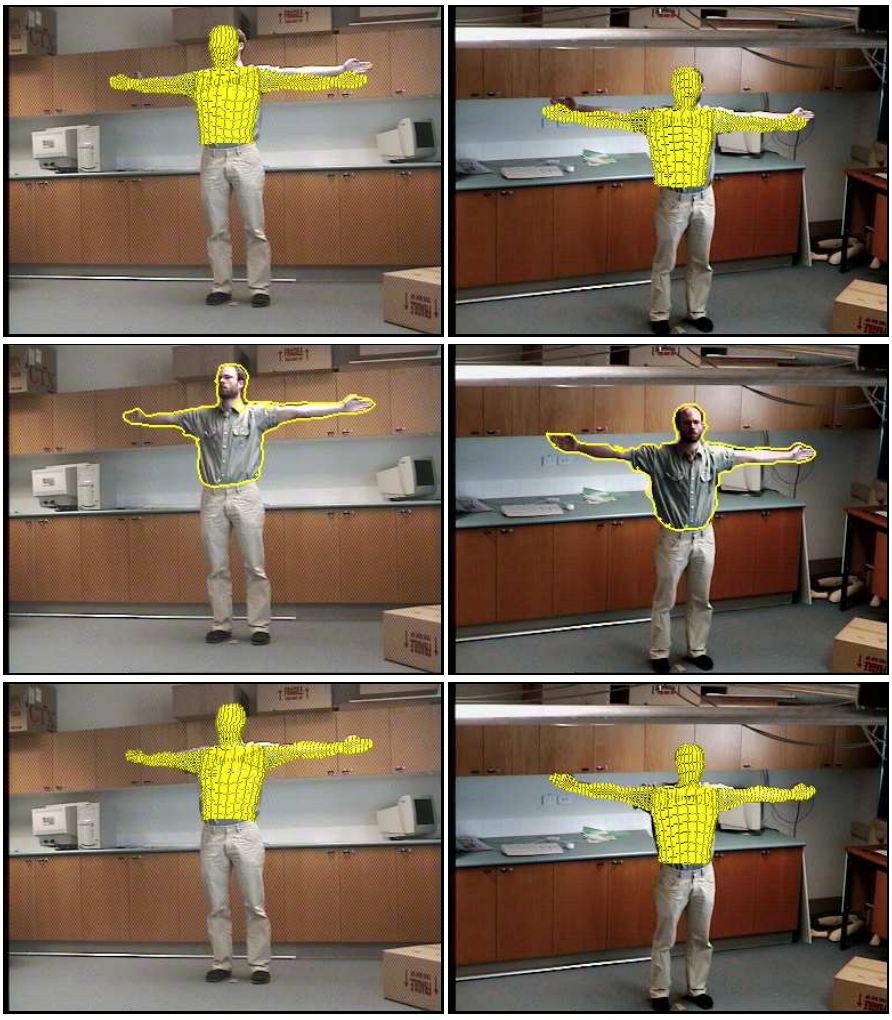


Fig. 11.2. Human pose estimation with the coupled contour-based approach given two camera views. **First row:** Initialization of the pose. The projection to the images is used as contour initialization. **Second row:** Estimated contour after 5 iterations. **Third row:** Estimated pose after 5 iterations.

For this reason, error accumulation is not an issue in the system described here. A result obtained with the optic flow model detailed below is shown in Figure 11.5.

Optic flow model. The remaining open question is how to compute the optic flow. The main goal here is to provide a prediction that brings the initialization closer to the correct pose in order to allow the contour-based method to converge to the correct solution in case of fast motion. Consequently, the optic flow method has to be able to deal with rather large displacements.

Fig. 11.3. Result with a two-step approach, i.e., extraction of the contours from the images followed by contour-based pose estimation. The same initialization as in Figure 11.2 was used. **Top row:** Estimated contour. **Bottom row:** Estimated pose. As pose and contour are not coupled, the contour extraction cannot benefit from the two camera views. Moreover, as the contour is not bound to the surface model, it can run away.

First assumption: gray value constancy. The basic assumption for optic flow estimation is the gray value constancy assumption, i.e., the gray value of a translated point does not change between the frames. With $\mathbf{w} := (u, v)$ denoting the optic flow, this can be expressed by

$$I(\mathbf{x} + \mathbf{w}, t + 1) - I(\mathbf{x}, t) = 0. \tag{11.20}$$

This equation is also called the *optic flow constraint*. Due to nonlinearity in $\mathbf{w}$, it is usually linearized by a Taylor expansion to yield

$$I_x u + I_y v + I_t = 0, \tag{11.21}$$

where subscripts denote partial derivatives. The linearization may be applied if displacements are small. For larger displacements, however, the linearization is not a good approximation anymore. Therefore, it has been suggested to minimize the original constraint in (11.20) [43] and to postpone all linearizations to the numerical scheme [2, 9], which comes down to so-called warping schemes [3, 5, 40]. These schemes can deal with rather large displacements and, therefore, are appropriate for the problem at hand.

Fig. 11.4. Result with a global Parzen estimator instead of the suggested local region statistics. The same initialization as in Figure 11.2 was used. **Top row:** Estimated contour. **Bottom row:** Estimated pose. Local differences between foreground and background are not modeled. With the global model, the right arm of the person better fits to the background.

Second assumption: smooth flow field. The gray value constancy assumption alone is not sufficient for a unique solution. Additional constraints have to be introduced. Here we stick to the constraint of a smooth flow field, as suggested in [29]. It leads to the following energy minimization problem

$$E(u,v) = \int_\Omega (I(\mathbf{x},t) - I(\mathbf{x}+\mathbf{w},t+1))^2 + \alpha(|\nabla u|^2 + |\nabla v|^2)\,\mathbf{dx} \;\rightarrow \min \quad (11.22)$$

that can be solved with variational methods. Note that exactly the same problem appeared in Section 11.3.3 for matching two contours via (11.15). Thus we can use almost the same scheme for computing the optic flow between images and for shape matching.

Noise and brightness changes. When matching two images, one has to expect noise and violations of the gray value constancy assumption. These effects have to be taken into account in the optic flow model. In order to deal with noise, one can apply a robust function $\Psi(s^2) = \sqrt{s^2 + 0.001^2}$ to the first term in (11.22) [5, 40]. This has the effect that outliers in the data have less influence on the estimation result.

Robustness to brightness changes can be obtained by assuming constancy of the gradient [9]:

$$\nabla I(\mathbf{x}+\mathbf{w},t+1) - \nabla I(\mathbf{x},t) = 0. \tag{11.23}$$

With both assumptions together, one ends up with the following energy:

$$\begin{aligned} E(u,v) = &\int_{\Omega_1} \Psi\big((I(\mathbf{x},t) - I(\mathbf{x}+\mathbf{w},t+1))^2\big)\,\mathbf{dx} \\ &+\gamma \int_{\Omega_1} \Psi\big((\nabla I(\mathbf{x},t) - \nabla I(\mathbf{x}+\mathbf{w},t+1))^2\big)\,\mathbf{dx} \\ &+\alpha \int_{\Omega_1} (|\nabla u|^2 + |\nabla v|^2)\,\mathbf{dx}. \end{aligned} \tag{11.24}$$

Note that the domain is restricted to the foreground region Ω_1, since we are only interested in correspondences within this region anyway. This masking of the background region has the advantage that it considers the most dominant motion discontinuities, which would otherwise violate the smoothness assumption of the optic flow model. Moreover, it allows for cropping the images to reduce the computational load.

Euler-Lagrange equations. According to the calculus of variations, a minimizer of (11.24) must fulfill the Euler-Lagrange equations

$$\begin{aligned} \Psi'(I_z^2)I_xI_z + \gamma\,\Psi'(I_{xz}^2 + I_{yz}^2)(I_{xx}I_{xz} + I_{xy}I_{yz}) - \alpha\,\Delta u = 0 \\ \Psi'(I_z^2)I_yI_z + \gamma\,\Psi'(I_{xz}^2 + I_{yz}^2)(I_{yy}I_{yz} + I_{xy}I_{xz}) - \alpha\,\Delta v = 0 \end{aligned} \tag{11.25}$$

with reflecting boundary conditions, $\Delta := \partial_{xx} + \partial_{yy}$, and the following abbreviations:

$$\begin{aligned} I_x &:= \partial_x I(\mathbf{x}+\mathbf{w},t+1), \\ I_y &:= \partial_y I(\mathbf{x}+\mathbf{w},t+1), \\ I_z &:= I(\mathbf{x}+\mathbf{w},t+1) - I(\mathbf{x},t), \\ I_{xx} &:= \partial_{xx} I(\mathbf{x}+\mathbf{w},t+1), \\ I_{xy} &:= \partial_{xy} I(\mathbf{x}+\mathbf{w},t+1), \\ I_{yy} &:= \partial_{yy} I(\mathbf{x}+\mathbf{w},t+1), \\ I_{xz} &:= \partial_x I(\mathbf{x}+\mathbf{w},t+1) - \partial_x I(\mathbf{x},t), \\ I_{yz} &:= \partial_y I(\mathbf{x}+\mathbf{w},t+1) - \partial_y I(\mathbf{x},t). \end{aligned} \tag{11.26}$$

Numerical scheme. The nonlinear system of equations in (11.25) can be solved with the numerical scheme proposed in [9]. It consists of two nested fixed point iterations for removing the nonlinearities in the equations. The outer iteration is in $\mathbf{w}^k$. It is combined with a downsampling strategy in order to better approximate the global optimum of the energy. Starting with the initialization $\mathbf{w} = 0$, a new estimate is computed as $\mathbf{w}^{k+1} = \mathbf{w}^k + (du^k, dv^k)^\top$. In each iteration one has to solve for the increment (du^k, dv^k). Ignoring here the term for the gradient constancy, which can be derived in the same way, the system to be solved in each iteration is

$$\begin{aligned} \Psi'(I_z^k)\left(I_x^k du^k + I_y^k dv^k + I_z^k\right) I_x^k - \alpha\Delta(u^k + du^k) = 0 \\ \Psi'(I_z^k)\left(I_x^k du^k + I_y^k dv^k + I_z^k\right) I_y^k - \alpha\Delta(v^k + dv^k) = 0 \end{aligned} \tag{11.27}$$

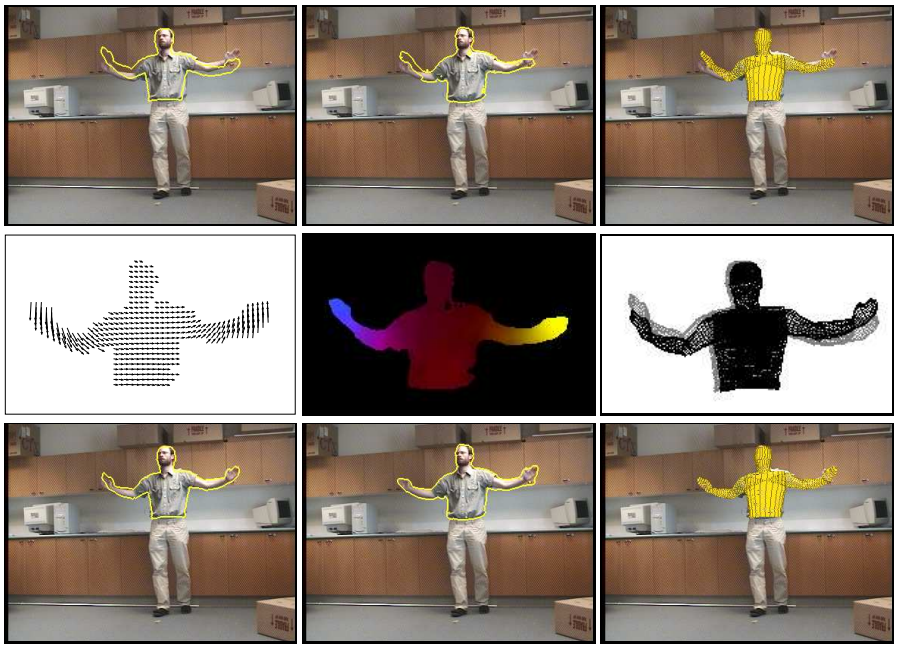


Fig. 11.5. Motion prediction by optic flow and its relevance. **First row:** Initialization with the pose from the previous frame (left). Due to fast motion, the initialization is far from the correct contour. Consequently, contour extraction (center) and tracking (right) fail. **Second row:** Optic flow field as arrow (left) and color plot (center) and prediction computed from this flow field (right). The brighter mesh shows the old pose, the dark mesh the predicted one. **Third row:** Like first row, but now the initialization is from the pose predicted by the optic flow.

If Ψ' is constant, this is the case for the shape matching problem in (11.15), (11.27) is already a linear system of equations and can be solved directly with an efficient iterative solver like SOR. If Ψ' depends on (du, dv), however, we have to implement a second fixed point iteration, now in $(du^{k,l}, dv^{k,l})$ to remove the remaining nonlinearity. Each inner iteration computes a new estimate of Ψ' from the most recent $(du^{k,l}, dv^{k,l})$. As Ψ' is kept fixed in each such iteration, the resulting system is linear in $(du^{k,l}, dv^{k,l})$ and can be solved with SOR. With a faster multigrid solver, it is even feasible to compute the optic flow in real-time [14]. However, in the scenario here, where the contour-based part is far from real-time performance, the difference to an SOR solver is probably not worth the effort.

11.5 Prior Knowledge of Joint Angle Configurations

The method surveyed in Sections 11.3 and 11.4 incorporates, apart from the input images, also prior knowledge explicitly given by the 3D shape model and the position of the joints. It has been demonstrated that this prior knowledge plays an important

role when seeking the contours. This is in accordance with findings in previous works on segmentation methods incorporating 2D shape priors [36, 18, 19]. In particular when the object of interest is partially occluded, the use of shape priors improves the results significantly.
While the method in Sections 11.3 and 11.4 includes a prior on the contour (for given pose parameters), it does not incorporate a prior on the pose parameters yet. Knowing the effects of prior shape knowledge, one expects similarly large improvements when using knowledge about familiar poses. It is intuitively clear that many poses are a-priori impossible or very unlikely, and that a successful technique for human tracking should exclude such solutions. Indeed, recent works on human pose estimation focus a lot on this issue [57, 59, 65, 11]. Their results confirm the relevance of pose priors for reliable tracking.

Integrating the prior via the Bayesian formula. The Bayesian formalism in (11.12) provides the basis for integrating such prior knowledge into the tracking technique. For convenience we repeat the formula:

$$p(\Phi, \xi | I) = \frac{p(I|\Phi, \xi) p(\Phi|\xi) p(\xi)}{p(I)} \rightarrow \max . \tag{11.28}$$

While the prior $p(\xi)$ has been ignored so far, the goal of this section is to learn a probability density from training samples and to employ this density in order to constrain the pose parameters.
As the prior should be independent from the global translation and rotation of the body in the training sequences, a uniform prior is applied to the global twist parameters ξ_{RBM}. Only the probability density for the joint angle vector $p(\Theta)$ is learned and integrated into the tracking framework.

Nonparametric density estimation. Figure 11.6 visualizes training data for the legs of a person from two walking sequences obtained by a marker-based tracking system with a total of 480 samples. Only a projection to three dimensions (the three joint angles of the right hip) is shown.

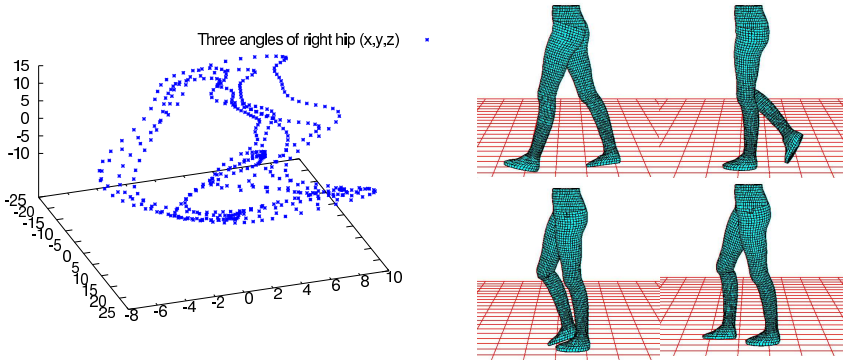


Fig. 11.6. Left: Visualization of the training data obtained from two walking sequences. Only a 3D projection (the three joint angles of the right hip) is shown. **Right:** Some training samples applied to a leg model.

There are many possibilities to model probability densities from such training samples. The most common way is a parametric representation by means of a Gaussian density, which is fully described by the mean and covariance matrix of the training samples. Such representations, however, tend to oversimplify the sample data. Although Figure 11.6 shows only a projection of the full configuration space, it is already obvious from this figure that pose configurations in a walking motion cannot be described accurately by a Gaussian density.
In order to cope with the non-Gaussian nature of the configuration space, [11] have advocated a nonparametric density estimate by means of the Parzen-Rosenblatt estimator [50, 47]. It approximates the probability density by a sum of kernel functions centered at the training samples. A common kernel is the Gaussian function, which leads to:

$$p(\Theta) = \frac{1}{\sqrt{2\pi}\sigma N} \sum_{i=1}^{N} \exp\left(-\frac{(\Theta_i - \Theta)^2}{2\sigma^2}\right) \tag{11.29}$$

where N is the number of training samples. Note that (11.29) does not involve a projection but acts on the conjoint configuration space of all angles. This means, also the interdependency between joint angles is taken into account.

Choice of the kernel width. The Parzen estimator involves the kernel width σ as a tuning parameter. Small kernel sizes lead to an accurate representation of the training data. On the other hand, unseen test samples close to the training samples may be assigned a too small probability. Large kernel sizes are more conservative, leading to a smoother approximation of the density, which in the extreme case comes down to a uniform distribution. Numerous works on how to optimally choose the kernel size are available in the statistics literature [58]. In our work, we fix σ as the maximum nearest neighbor distance between all training samples, i.e., the next sample is always within one standard deviation. This choice is motivated from the fact that our samples stem from a smooth sequence of poses.

Energy minimization. Taking the prior density into account leads to an additional term in the energy (11.13) that constrains the pose parameters to familiar configurations:

$$E_{\text{Prior}} = -\log(p(\xi)). \tag{11.30}$$

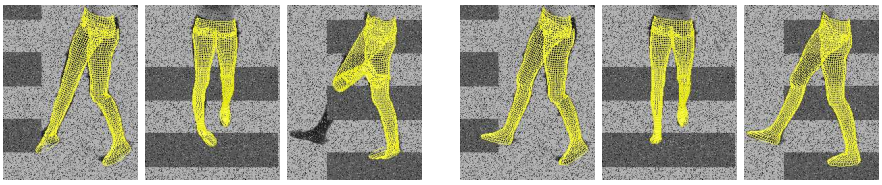
The gradient descent of (11.30) in Θ reads

$$\partial_t \Theta = -\frac{\partial E_{\text{Prior}}}{\partial \Theta} = \frac{\sum_{i=1}^{N} w_i(\Theta_i - \Theta)}{\sigma^2 \sum_{i=1}^{N} w_i} \tag{11.31}$$

$$w_i := \exp\left(-\frac{|\Theta_i - \Theta|^2}{2\sigma^2}\right). \tag{11.32}$$

Obviously, this equation draws the pose to the next local maximum of the probability density. It can be directly integrated into the linear system (11.19) from Section 11.3.3. For each joint j, an additional equation $\theta_j^{k+1} = \theta_j^k + \tau \partial_t \theta_j^k$ is appended to the linear system. In order to achieve an equal weighting of the image against the prior, the new equations are weighted by the number of point correspondences obtained from the contours. The step size parameter $\tau = 0.125\sigma^2$ yielded empirically stable results.

Regularization. The prior obviously provides a regularization of the equation system. Assume a foot is not visible in any camera view. Without prior knowledge, this



Without prior With prior

Fig. 11.7. Relevance of the learned configurations for the tracking stability. Occlusions locally disturb the image-driven pose estimation. This can finally cause a global tracking failure. The prior couples the body parts and seeks the most familiar configuration given all the image data.

would automatically lead to a singular system of equations, since there are no correspondences that generate any constraint equation with respect to the joint angles at the foot. Due to the interdependency of the joint angles, the prior equation draws the joint angles of the invisible foot to the most probable solution given the angles of all visible body parts.

Robustness to partial occlusions. Apart from providing unique solutions, the prior also increases the robustness of the tracking in case of unreliable data, as demonstrated in Figure 11.7. Instead of nonsensically fitting the bad data, the method seeks a familiar solution that fits the data best. Another example is shown in Figure 11.8 where, additionally to 25% uniform noise, 50 rectangles of random position, size, and gray value were placed in each image.

11.6 Discussion

The human tracking system described in the preceding sections is based only on few assumptions on the scene and works quite reliably, as shown for rigid bodies in [53, 10] and humans in [52] as well as in this chapter. Further experiments with the same technique are contained in the next chapter. Nevertheless, there are still lots of challenges that shall be discussed in this section.

Running time. One of these challenges is a reduction of the running time. Currently, with a 2GHz laptop, the method needs around 50 seconds per frame for 384×280 stereo images. Even though one could at least obtain a speedup of factor 4 by using faster hardware and optimizing the implementation, the method is not adequate for real-time processing. The main computational load is caused by the iterative contour and pose estimation and the approximation of region statistics involved therein. More considerable speedups may be achieved by using the parallelism in these operations via an implementation on graphics hardware.
However, most applications of 3D human motion tracking do not demand real-time performance but high accuracy. Sports movement analysis and modeling of motion patterns for computer graphics are run in batch mode anyway. Thus, improving the running time would mainly reduce hardware costs and improve user interaction.

Auto-initialization. Trying to automatically initialize the pose in the first frame is another interesting challenge. So far, a quite accurate initialization of the pose is

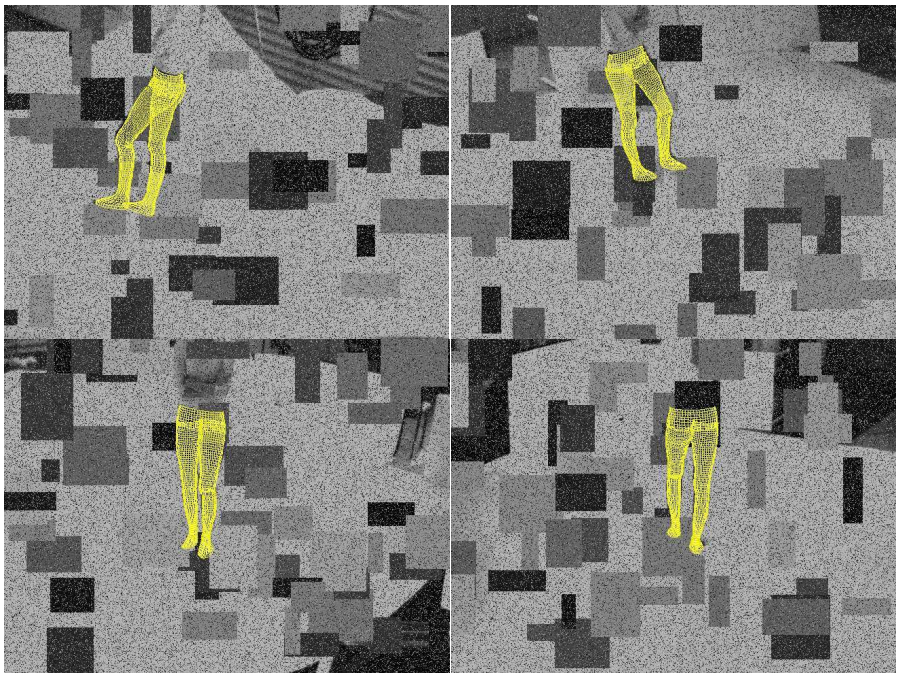


Fig. 11.8. Pose estimates in a sample frame disturbed by 50 varying rectangles with random position, size, and gray value and 25% uncorrelated pixel noise.

needed. For this kind of detection task, the proposed framework seems less appropriate, as it is difficult to detect silhouettes in cluttered images. For object detection, patch based methods have already proven their strength. Thus they can probably solve this task more efficiently. Auto-initialization has, for instance, been demonstrated in [45] for rigid bodies. Works in the scope of human tracking can be found in [49, 25, 60, 1, 61]. Some of these approaches even use silhouettes for the initialization. However, in these cases the contour must be easy to extract from the image data. This is feasible, for instance, with background subtraction if the background is static. The advantage of such discriminative tracking is the possibility to reinitialize after the person has been lost due to total occlusion or the person moving out of all camera views. Combinations of discriminative and generative models, as suggested in [61], are discussed in Chapter 8.

Clothed people. In nearly all setups, the subjects have to wear a body suit to ensure an accurate matching between the silhouettes and the surface models of the legs. Unfortunately, body suits may be uncomfortable to wear in contrast to loose clothing (shirts, shorts, skirts etc.). The subjects also move slightly different in body suits compared to being in clothes since all body parts (even unfavored ones) are clearly visible. The incorporation of cloth models would ease the subjects and also simplify the analysis of outdoor scenes and arbitrary sporting activities. A first approach in this direction is presented in Chapter 12.

Prior knowledge on motion dynamics. In Section 11.5, a prior on the joint angle vector has been imposed. This has lead to a significant improvement in the tracking reliability given disturbed or partially occluded input images. However, the prior is on the static pose parameters only. It does not take prior information about motion patterns, i.e. the dynamics, into account. Such dynamical priors can be modeled by regression methods such as linear regression or Gaussian processes [48]. In the ideal case, the model yields a probability density, which allows the sound integration in a Bayesian framework [17]. Recently, nonlinear dimensionality reduction methods have become very popular in the context of motion dynamics.

Subspace learning. The idea of dimensionality reduction methods is to learn a mapping between the original, high-dimensional space of pose parameters and a low-dimensional manifold in this space. Solutions are expected to lie only on this manifold, i.e., the search space has been considerably reduced. The motivation for this procedure is the expected inherent low-dimensional structure in a human motion pattern. For instance, the pattern of walking is basically a closed one-dimensional loop of poses when modeled on an adequate, however complex, manifold. Linear projection methods like PCA can be supposed to only insufficiently capture all the limb movements in motion patterns. Nonlinear methods like *Gaussian process latent variable models* (GPLVM), ISOMAP, or others have been shown to be more adequate [35, 27, 25, 65]. See also Chapter 2 and Chapter 10 for more detailed insights.
While dimensionality reduction can successfully model a single motion pattern like walking, running, jumping, etc., it is doubtful that the same concept still works if the model shall contain multiple such patterns. Even though each single pattern may be one- or two-dimensional, the combination of patterns is not. Hence, one has to employ a mixture model with all the practical problems concerning the choice of mixture components and optimization. In case of multiple motion patterns, it may thus be beneficial to define models in the original high-dimensional space, as done, e.g., in the last chapter for static pose priors. This way, one knows for sure that all different patterns can be distinguished. Dealing with the arising high dimensionality when dynamics are included, however, remains a challenging open problem.

11.7 Summary

This chapter has presented a generative Bayesian model for human motion tracking. It includes the joint estimation of the human silhouette and the body pose parameters. The estimation is constrained by a static pose prior based on nonparametric Parzen densities. Furthermore, the pose in new frames is predicted by means of optic flow computed in the foreground region. The approach demands a predefined surface model, the positions of the joints, an initialization of the pose in the first frame, and a calibration of all cameras to the same world coordinate system. In return one obtains reliable estimates of all pose parameters without error accumulation. There is no assumption of a static background involved. Instead, the foreground and background regions are supposed to be locally different. Due to the pose prior, the method can cope with partial occlusions of the person. We also discussed further extensions, in particular the use of image patches for initial pose detection and the integration of dynamical priors.

Acknowledgements

We acknowledge funding of our research by the project CR250/1 of the German Research Foundation (DFG) and by the Max-Planck Center for visual computing and communication.

References

1. Agarwal A. and Triggs B. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58, Jan. 2006.
2. Alvarez L., Weickert J. and Sánchez J. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1):41–56, Aug. 2000.
3. Anandan P. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.
4. Besl P. and McKay N. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:239–256, 1992.
5. Black M.J. and Anandan P. The robust estimation of multiple motions: parametric and piecewise smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, Jan. 1996.
6. Blake A. and Zisserman A. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.
7. Bregler C. and Malik J. Tracking people with twists and exponential maps. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8–15, Santa Barbara, California, 1998.
8. Bregler C., Malik J. and Pullen K. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, 2004.
9. Brox T., Bruhn A., Papenberg N. and Weickert J. High accuracy optical flow estimation based on a theory for warping. In T.Pajdla and J.Matas, editors, *Proc.8th European Conference on Computer Vision*, volume 3024 of *LNCS*, pages 25–36. Springer, May 2004.
10. Brox T., Rosenhahn B., Cremers D. and Seidel H.-P. High accuracy optical flow serves 3-D pose tracking: exploiting contour and flow based constraints. In A.Leonardis, H.Bischofand A.Prinz, editors, *Proc.European Conference on Computer Vision*, volume 3952 of *LNCS*, pages 98–111, Graz, Austria, May 2006. Springer.
11. Brox T., Rosenhahn B., Kersting U. and Cremers D. Nonparametric density estimation for human pose tracking. In K.F. etal., editor, *Pattern Recognition*, volume 4174 of *LNCS*, pages 546–555, Berlin, Germany, Sept. 2006. Springer.
12. Brox T. and Weickert J. A TV flow based local scale estimate and its application to texture discrimination. *Journal of Visual Communication and Image Representation*, 17(5):1053–1073, Oct. 2006.
13. Brox T. and Cremers D. On the statistical interpretation of the piecewise smooth Mumford-Shah functional. In *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *LNCS*, pages 203–213 Springer, 2007.

14. Bruhn A. and Weickert J. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *Proc.10th International Conference on Computer Vision*, pages 749–755. IEEE Computer Society Press, Beijing, China, Oct. 2005.
15. Chan T. and Vese L. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, Feb. 2001.
16. Chetverikov D. A simple and efficient algorithm for detection of high curvature points. In N.Petkov and M.Westenberg, editors, *Computer Analysis of Images and Patterns*, volume 2756 of *LNCS*, pages 746–753, Groningen, 2003. Springer.
17. Cremers D. Dynamical statistical shape priors for level set based tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1262–1273, Aug. 2006.
18. Cremers D., Kohlberger T. and Schnörr C. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, 36(9):1929–1943, Sept. 2003.
19. Cremers D., Osher S. and Soatto S. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, 69(3):335–351, 2006.
20. Cremers D., Rousson M. and Deriche R. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007.
21. DeCarlo D. and Metaxas D. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127, July 2000.
22. Dempster A., Laird N. and Rubin D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society series B*, 39:1–38, 1977.
23. Dervieux A. and Thomasset F. A finite element method for the simulation of Rayleigh–Taylor instability. In R.Rautman, editor, *Approximation Methods for Navier–Stokes Problems*, volume 771 of *Lecture Notes in Mathematics*, pages 145–158. Springer, Berlin, 1979.
24. Dunn D., Higgins W.E. and Wakeley J. Texture segmentation using 2-D Gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):130–149, Feb. 1994.
25. Elgammal A. and Lee C. Inferring 3D body pose from silhouettes using activity manifold learning. In *Proc.International Conference on Computer Vision and Pattern Recognition*, pages 681–688, Washington D.C., 2004.
26. Gavrila D. and Davis L. 3D model based tracking of humans in action: a multiview approach. In *ARPA Image Understanding Workshop*, pages 73–80, Palm Springs, 1996.
27. Grochow K., Martin S.L., Hertzmann A. and Popović Z. Style-based inverse kinematics. In *ACM Transactions on Graphics (Proc.SIGGRAPH)*, volume23, pages 522–531, 2004.
28. Heiler M. and Schnörr C. Natural image statistics for natural image segmentation. *International Journal of Computer Vision*, 63(1):5–19, 2005.
29. Horn B. and Schunck B. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
30. Horprasert T., Harwood D. and Davis L. A statistical approach for real-time robust background subtraction and shadow detection. In *International Confer-*

ence on Computer Vision, FRAME-RATE Workshop, Kerkyra, Greece, 1999. Available at `www.vast.uccs.edu/~tboult/FRAME`.
31. Kadir T. and Brady M. Unsupervised non-parametric region segmentation using level sets. In *Proc.Ninth IEEE International Conference on Computer Vision*, volume2, pages 1267–1274, 2003.
32. Kim J., Fisher J., Yezzi A., Cetin M. and Willsky A. A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Transactions on Image Processing*, 14(10):1486–1502, 2005.
33. Klette R. and Rosenfeld A. *Digital Geometry–Geometric Methods for Digital Picture Analysis.* Morgan Kaufmann, San Francisco, 2004.
34. Klette R., Schlüns K. and Koschan A. *Computer Vision. Three-Dimensional Data from Images.* Springer, Singapore, 1998.
35. Lawrence N.D. Gaussian process latent variable models for visualisation of high dimensional data. In *Neural Information Processing Systems 16.*
36. Leventon M.E., Grimson W.E.L. and Faugeras O. Statistical shape influence in geodesic active contours. In *Proc.2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume1, pages 316–323, Hilton Head, SC, June 2000.
37. Lowe D. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
38. Marchand E., Bouthemy P. and Chaumette F. A 2D-3D model-based approach to real-time visual tracking. *Image and Vision Computing*, 19(13):941–955, Nov. 2001.
39. McLachlan G. and Krishnan T. *The EM Algorithm and Extensions.* Wiley series in probability and statistics. John Wiley & Sons, 1997.
40. Mémin E. and Pérez P. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Transactions on Image Processing*, 7(5):703–719, May 1998.
41. Mumford D. and Shah J. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.
42. Murray R., Li Z. and Sastry S. *Mathematical Introduction to Robotic Manipulation.* CRC Press, Baton Rouge, 1994.
43. Nagel H.-H. and Enkelmann W. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–593, 1986.
44. Osher S. and Sethian J.A. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
45. Özuysal M., Lepetit V., Fleuret F. and Fua P. Feature harvesting for tracking-by-detection. In *Proc.European Conference on Computer Vision*, volume 3953 of *LNCS*, pages 592–605. Springer, Graz, Austria, 2006.
46. Paragios N. and Deriche R. Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, 13(1/2):249–268, 2002.
47. Parzen E. On the estimation of a probability density function and the mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.
48. Rasmussen C.E. and Williams C.K.I. *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, MA, 2006.

49. Rosales R. and Sclaroff S. Learning body pose via specialized maps. In *Proc.Neural Information Processing Systems*, Dec. 2001.
50. Rosenblatt F. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837, 1956.
51. Rosenhahn B., Brox T., Cremers D. and Seidel H.-P. A comparison of shape matching methods for contour based pose estimation. In R.Reulke, U.Eckhardt, B.Flach, U.Knauer and K.Polthier, editors, *Proc.International Workshop on Combinatorial Image Analysis*, volume 4040 of *LNCS*, pages 263–276, Berlin, Germany, June 2006. Springer.
52. Rosenhahn B., Brox T., Kersting U., Smith A., Gurney J. and Klette R. A system for marker-less motion capture. *Künstliche Intelligenz*, (1):45–51, 2006.
53. Rosenhahn B., Brox T. and Weickert J.. Three-dimensional shape knowledge for joint image segmentation and pose tracking. *International Journal of Computer Vision*, 73(3):243–262, July 2007.
54. Rousson M., Brox T.and Deriche R. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc.International Conference on Computer Vision and Pattern Recognition*, pages 699–704, Madison, WI, June 2003.
55. Shevlin F. Analysis of orientation problems using Plücker lines. In *International Conference on Pattern Recognition (ICPR)*, volume1, pages 685–689, Brisbane, 1998.
56. Shi J. and Tomasi C. Good features to track. In *Proc.International Conference on Computer Vision and Pattern Recognition*, pages 593–600, 2004.
57. Sidenbladh H., Black M. and Sigal L. Implicit probabilistic models of human motion for synthesis and tracking. In A. Heyden, G. Sparr, M. Nielsen and P. Johansen, editors, *Proc.European Conference on Computer Vision*, volume 2353 of *LNCS*, pages 784–800. Springer, 2002.
58. Silverman B.W. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, New York, 1986.
59. Sminchisescu C. and Jepson A. Generative modeling for continuous non-linearly embedded visual inference. In *Proc.International Conference on Machine Learning*, 2004.
60. Sminchisescu C., Kanaujia A., Li Z. and Metaxas D. Discriminative density propagation for 3D human motion estimation. In *Proc.International Conference on Computer Vision and Pattern Recognition*, pages 390–397, 2005.
61. Sminchisescu C., Kanaujia A. and Metaxas D. Learning joint top-down and bottom-up processes for 3D visual inference. In *Proc.International Conference on Computer Vision and Pattern Recognition*, pages 1743–1752, 2006.
62. Sminchisescu C. and Triggs B. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 22(6):371–391, 2003.
63. Sommer G., editor. *Geometric Computing with Clifford Algebra: Theoretical Foundations and Applications in Computer Vision and Robotics*. Springer, Berlin, 2001.
64. Tsai A., Yezzi A. and Willsky A. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolationand magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, 2001.
65. Urtasun R., Fleet D.J. and Fua P. 3D people tracking with Gaussian process dynamical models. In *Proc.International Conference on Computer Vision and Pattern Recognition*, pages 238–245. IEEE Computer Society Press, 2006.

66. Zhu S.-C. and Yuille A.. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, Sept. 1996.

Appendix: Semi-automatic Acquisition of a Body Model

As most model-based human tracking methods, also the approach in this chapter is based on a model that consists of multiple rigid parts interconnected by joints. Basically, this body model has to be designed manually. Thus, often one can find quite simplistic stick figures based on ellipsoidal limbs in the literature. In this subsection, we briefly describe a method that allows to construct a more accurate surface model by means of four key views of a person as shown in Figure11.9.

Body separation. After segmentation we separate the arms from the torso of the model. Since we only generate the upper torso, the user can define a bottom line of the torso by clicking on the image. Then we detect the arm pits and the neck joint from the *front view* of the input image. The arm pits are simply given by the two lowermost corners of the silhouette which are not at the bottom line and exceed a preset angle threshold. The position of the neck joint can be found when moving along the boundary of the silhouette from an upper shoulder point to the head. The narrowest x-slice of the silhouette gives the neck joint.

Joint localization. After this rough segmentation of the human torso we detect the positions of the arm joints. We use a special reference frame (*joint view* in Figure11.9) that allows to extract arm segments. To gain the length of the hands, upper arms, etc. we first apply a skeletonization procedure. Skeletonization [33] is a process of reducing object pixels in a binary image to a skeletal remnant that largely preserves the extent and connectivity of the original region while eliminating most of the original object pixels. Then we use the method presented in [16] to detect corners of the skeleton to identify joint positions of the arms.

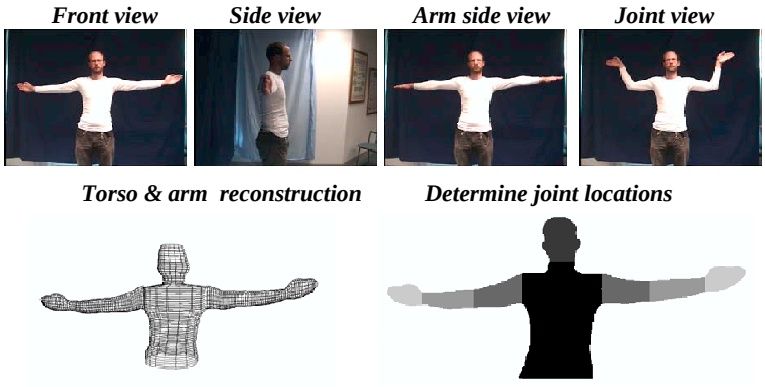


Fig. 11.9. Steps for semi-automatically deriving a body model from four input images.

Since the center of the elbow joint is not at the center of the arm but beneath, the joint localizations need to be refined. For this reason, we shift the joint position aiming at correspondence with the human anatomy. The resulting joint locations are shown in the middle right image of Figure 11.9.

Surface mesh reconstruction. For surface mesh reconstruction we assume calibrated cameras in nearly orthogonal views. Then a shape-from-silhouettes approach [34] is applied. We detect control points for each slice and interpolate them by a B-spline curve using the DeBoor algorithm. We start with one slice of the first image and use its edge points as the first two reference points. They are then multiplied with the fundamental matrix of the first to the second camera, and the resulting epipolar lines are intersected with the second silhouette resulting in two more reference points. The reference points are intersected leading to four control points in 3D space.

For arm generation we use a similar scheme for building a model: We use two other reference frames (input images 2 and 3 in Figure 11.9). Then the arms are aligned horizontally and we use the fingertip as starting point on both arms. These silhouettes are sliced vertically to obtain the width and height of each arm part. The arm patches are then connected to the mid plane of the torso.