A Silhouette Based Human Motion Tracking System

Bodo Rosenhahn[1,3], Uwe G. Kersting[2], Lei He[1], Andrew W. Smith[2]
Thomas Brox[4], Reinhard Klette[1], Hans-Peter Seidel[3]

[1] Centre for Imaging Technology and Robotics (CITR),
[2]Department of Sport and Exercise Science
University of Auckland, New Zealand
[3] Max Planck Center Saarbrücken, Germany
rosenhahn@mpi-sb.mpg.de
[4]Math. Image Analysis Group, Saarland University, Germany

Abstract

This paper proposes a system for model based human motion estimation. We start with a human model generation system, which uses a set of input images to automatically generate a free-form surface model of a human upper torso. We subsequently determine joint locations automatically and generate a texture for the surface mesh. Following this, we present morphing and joint transformation techniques to gain more realistic human upper torso models. An advanced model such as this is used in a system for silhouette based human motion estimation. The presented motion estimation system contains silhouette extraction based on level set functions, a correspondence module, which relates image data to model data and a pose estimation module. This system is used for a variety of experiments: Different camera setups (between one to four cameras) are used for the experiments and we estimate the pose configurations of a human upper torso model with 21 degrees of freedom at two frames per second. We also discuss degenerated cases for silhouette based human motion estimation. Next, a comparison of the motion estimation system with a commercial marker based tracking system is performed to gain a quantitative error analysis. The results show the applicability of the system for marker-less human movement analysis. Finally we present experimental results on tracking leg models and show the robustness of our algorithms even for corrupted image data.

1 Introduction

Human motion estimation from image sequences refers to the determination of rigid body motion [34] and joint angles of a 3D human model from 2D image data. Different applications of human motion capturing exist, e.g., in the area of surveillance. Another common application aims to analyze the captured data for subsequent motion analysis purposes, e.g., clinical studies, diagnostics of orthopaedic patients or to help athletes to understand and improve their performances. The motivation for the present paper is the sport movement analysis: We will introduce a marker-less human motion tracking system for motion analysis. We will show that our algorithms are suited for this task, they are reasonably fast and fairly accurate. To demonstrate this, we present various experiments and perform a system comparison between the presented human motion capture system with a marker-based motion analysis [33] system.

For human motion analysis, a variety of approaches exist in the literature, with respect to different ways of representation for human models, different image processing techniques and solution methods. The global interest in this topic is shown at workshops, conferences and special journal issues in the past and the near future just dealing with this topic [22].

Surveys on existing methods can be found in [31, 4, 16]. The existing works can be categorised into such areas as model types (cylinders, stick figures, patches, super quadrics, scaled prismatics, CAD-models etc.), abstraction levels (edges, silhouettes, texture, etc.), or object parts (varying from 2 to 17).

For simplicity, sports scientists often prefer commercial marker based tracking systems, e.g. provided by Motion Analysis [33], Vicon [52] or Simi [49]. Using markers comes along with intrinsic problems, e.g. incorrect tracking of markers, tracking failures, the need for special lab environments and lighting conditions and the fact, that people do not feel comfortable with markers attached to the body. This often leads to unnatural motion patterns. As well, marker-based systems are designed to track the motion of the markers themselves and thus it must be assumed that the recorded motion of the markers is identical to the motion of the underlying human segments. Since human segments are not truly rigid, this assumption may cause problems, especially in highly dynamic movement typically seen in sporting activities. For these reasons, marker-less tracking is an important field of research. It requires knowledge in biomechanics, computer vision and computer graphics.

Typically, researchers working in the area of computer vision often use simplified human body models, e.g., based on stick, ellipsoidal, cylindrical or skeleton models [39, 5, 30, 14, 32, 17, 21]. In computer graphics advanced object modelling and texture mapping techniques for human motions are well known [35, 1, 10, 7, 50, 53], but the image processing and pose estimation techniques are often simplified. Often model assumptions are made, and acquired manually. Automatic model generation systems are e.g. presented in [29, 23, 25], but here also manual interventions (e.g. rescaling of images or the determination of joint locations) are necessary.

In this paper, we start to close these gaps by introducing a system for silhouette based human motion estimation, which relies on an accurate and realistic human model. We further describe an automatic model generation system, which assumes four input images in a predefined setup. These images are used to reconstruct the human surface

model and to determine the joint locations on the surface meshes. We further apply local and global morphing techniques to get realistic motions of the upper torso model. This kind of model is used within the human motion estimation system. The system consists of an advanced image segmentation method based on level set functions, dynamic occlusion handling and kinematic chains of higher complexity (21 degrees of freedom). Finally, we perform a comparison of the system with a commercial marker based tracking system [33] to analyze sport movements. We perform and analyze exercises, such as push ups or sit ups. This results in a quantitative error analysis. The algorithm proves as stable, robust and fairly accurate.
The contribution is based on former works of the authors regarding pose estimation of free-form contours and surfaces [41, 43, 44] and image segmentation [6]. To be self-consistent, these works will be summarized in section 3 and 4.1. This paper comprises and extends an earlier work presented on two conferences [42, 45]. In comparison to these early presentations of the basic system, the present paper contains a much more detailed description of the approach, an automatic human model generation system, and demonstrates the generality of the method by means of additional experiments.
The contribution is organized as follows: We will start with the model generation system in Section 2. This part is self-contained, and readers who are not interested in the task to acquire a 3D model, can skip this section. Then we continue with foundations about 2D-3D pose estimation. This results in a core algorithm for silhouette based pose estimation needed in the motion estimation system. In Section 4 we will continue with the presentation of the basic modules of the motion capture system. Here we will describe image segmentation based on level sets and the morphing techniques for more realistic human models. Section 5 presents experimental results of an upper body model. Here we will also explain the dynamic occlusion handling and the used particle filter. We will demonstrate the performance of our algorithms on different image sequences and motion patterns. Finally we will also present experiments performed with a lower body model. We show that the algorithm can be applied on completely different models and can handle corrupted image data. The contribution ends with a summary.

2 Automatic human upper torso generation

In the literature model reconstruction techniques can be broadly divided into active and passive methods. Where active methods use a light pattern projected into the scene or a laser ray emitting from a transmitter, passive techniques use the image data itself. Our approach is a passive reconstruction method due to its greater flexibility in scene capturing and to being a low-cost technique. Kakadiaris et al. [25] propose a system for 3D human body model acquisition by using three cameras in mutually orthogonal views. A subject is requested to perform a set of movements according to a protocol. The body parts are identified and reconstructed incrementally from 2D deformable contours. Hiltion et al. [23] propose an approach for modelling a human body from four views. The approach uses extrema to find feature points. It is simple and efficient. However, it is not reliable for finding the neck joint and it does not provide a solution to determine elbow or wrist joints. Lee et al. [29] build a seamless human model. Their approach obtains robust and efficient results, but it cannot detect joint

positions which have to be arranged manually. Figure 1 gives an overview of the input images and the implemented modules.

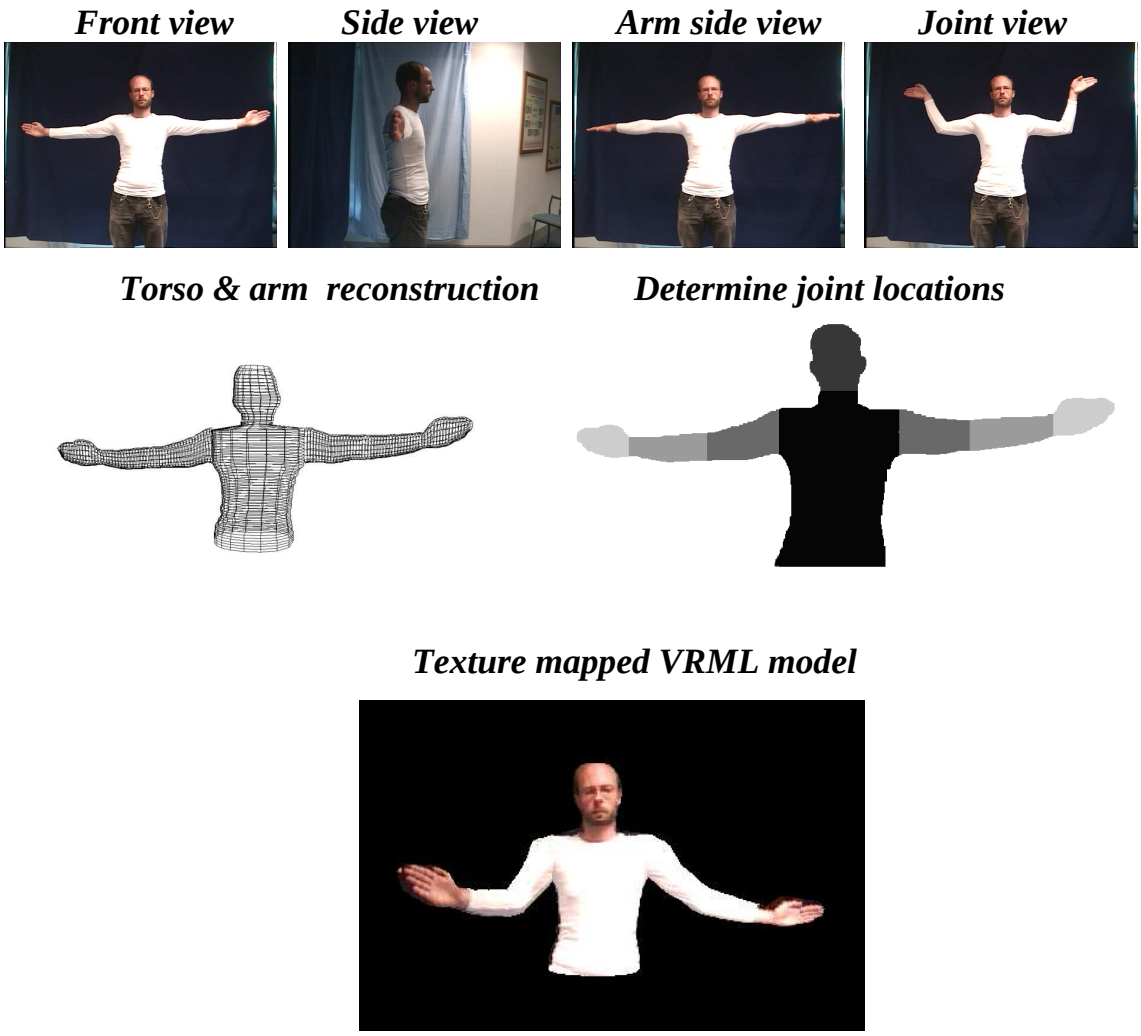

Figure 1: Steps of the implemented system. Four input images are used for model generation.

2.1 Modules for upper body model generation

The model generation system can be subdivided into different modules which will be described briefly:

Segmentation

Segmentation is the process of extracting a region of interest from an image. Accuracy and efficiency of contour detection are crucial for the final outcome. Fortunately, the task is relatively easy to solve since we assume a person in a lab environment with known uni-colored background. Here we use a modified version of [24], which proves to be fast and stable: To differentiate between object and background pixels we compare pixels of typical background characteristics with all pixels in the given image. The difference between two pixels is measured with two components, brightness and chromaticity.

Thresholds are used to segment the images. Afterwards the images are smoothed using morphological operators [27].

Body separation

Firstly, it is necessary to 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. 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 a rough segmentation of the human torso we detect the positions of the arm joints. Basically, we use a special reference frame (*joint view*) which allows to extract arm segments. To gain the length of the hands, upper arms, etc. we firstly apply a

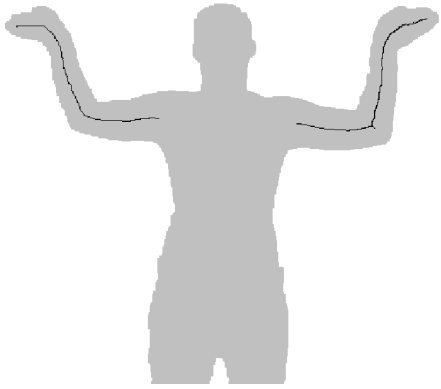

Figure 2: Skeletonization using a closed Chamfer distance transform.

skeletonization procedure. Skeletonization 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. Two approaches are common, those based on thinning and those based on distance transforms. Thinning approaches are a topological way to determine a skeleton, whereas a distance transforms results in a medial axes. Therefore, it is sometimes called medial axis transformation, MAT. We decided to work with the skeletons based on the Chamfer distance transform. Since the resulting skeletons are not connected, we close the skeleton by connecting nearest non-neighboring points. This leads to closed skeletons as shown in Figure 2. We further use the method presented in [12] to detect corners on the skeleton to identify joint positions of the arms.

Moreover, we point out that the joint localizations need to be refined since the center of the elbow joint is not at the center of the arm, but beneath. For this reason we shift the joint position aiming for correspondence with the human anatomy, see in Figure 3: Starting from the shoulder joint S and elbow joint E, we use the midpoint C between the right boundary of the silhouette B and E as new elbow position. The resulting joint locations are shown in Figure 4.

Surface Mesh Reconstruction

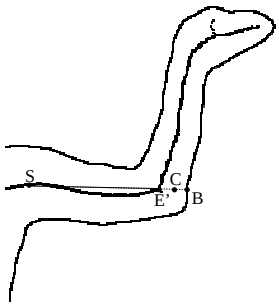

Figure 3: Adapting elbow joint locations.

Figure 4: Extracted joint segments.

For surface mesh reconstruction we assume calibrated cameras in nearly orthogonal views. Then a shape-from-silhouettes approach [28] is applied. We attempt to find control points for each slice, and by interpolating them as 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.

Figure 5: Texture fusion: The images from the stereo setup are merged to get a texture map for the head: the left texture gives a useful face, whereas the right texture gives a useful ear and side view of the face. The fusion of both textures leads to a new texture used for the 3D model.

For arm generation we use a different scheme for building a model: We use two other reference frames (input images 2 and 3 in Figure 1). Then the arms are aligned horizontally and we use the fingertip as starting point on both arms. These silhouettes are sliced vertically to gain the width and height of each arm part. The arm patches are

then connected to the mid plane of the torso.

Texture mapping

For texture mapping, we generate a texture file as a combination of the different views: We apply the multi-resolution method proposed by Burt et al. [3] for removing boundaries between different image sources. This is achieved by using a weighted average splining technique. For sake of simplicity, we adapt it to a linear weighted function. A texture resulting from a fusion of two different input views is shown on the right of Figure 5.

2.2 Results of model generation

We tested the algorithm on four different models. Model generation results from two

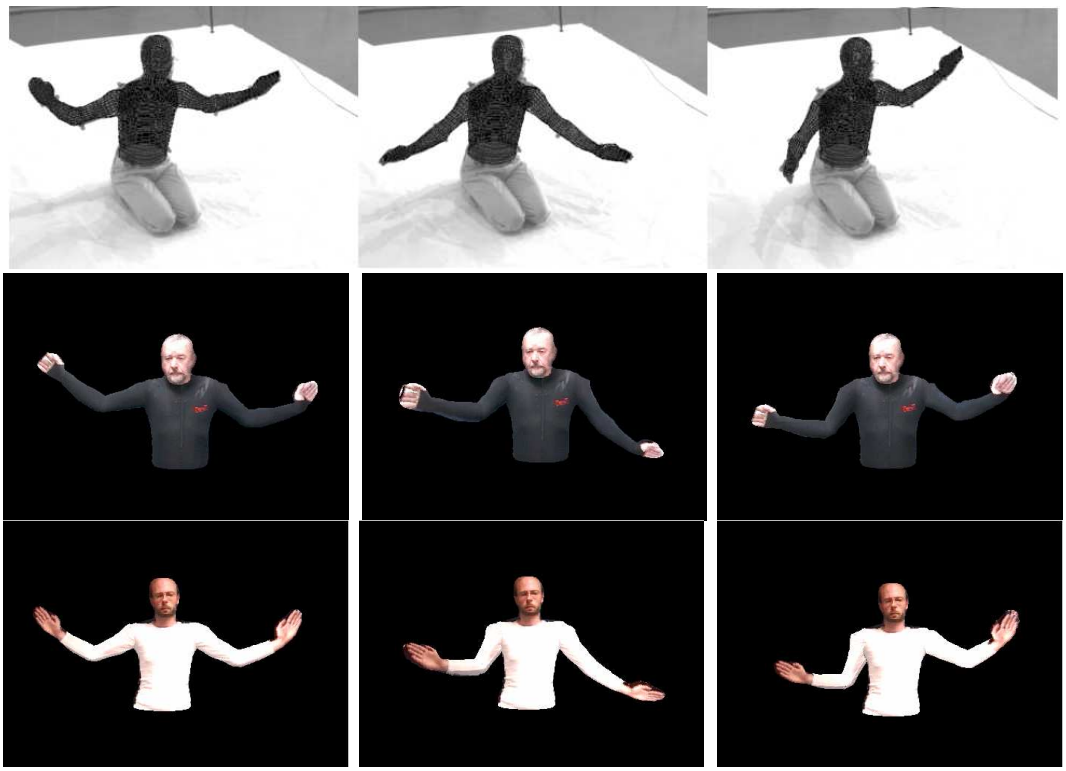

Figure 6: Pose results of the pose recognition software. Mimicking the arm configurations with the generated models.

persons are shown in the lower two rows of figure 6. An additional useful application apart from gaining the desired model for the tracking algorithm is to animate the models using motion capture data for mimicking: The top row in Figure 6 shows some capture results using a human motion estimation algorithm (described in the other sections) and below are the arm pose configurations of the generated model. This allows us to let the "Reinhard" model mimic the actors (Bodo) motions.

For a quantitative error analysis we compared reconstructed body parts with (manually) measured ones. A comparison shows a maximum deviation of 2 cm. This accuracy is sufficient for our tasks. For more information on the model generation system, see [20].

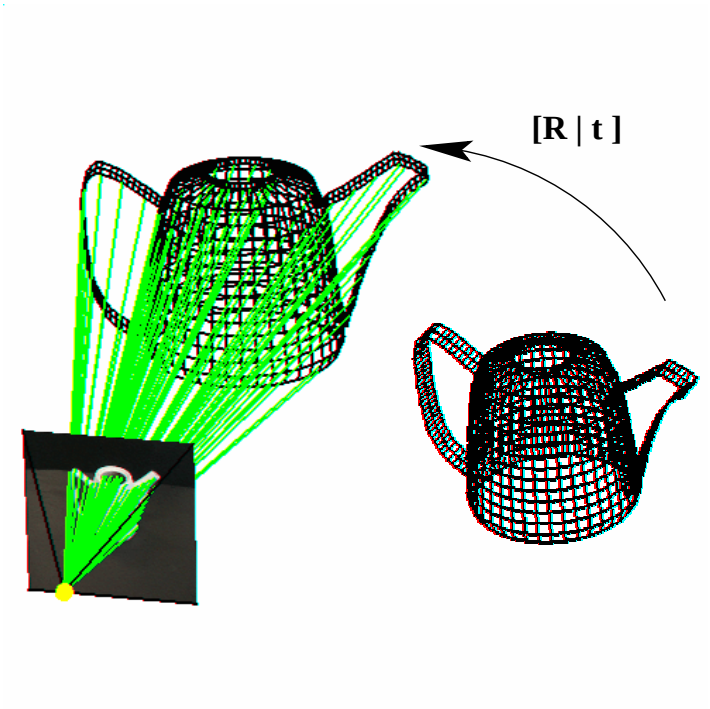

Figure 7: The pose scenario: the aim is to estimate the pose $\boldsymbol{R}$, $\boldsymbol{t}$.

3 2D-3D pose estimation

2D-3D pose estimation [19] means to estimate a rigid body motion which maps a 3D object model to an image of a calibrated camera, see Fig. 7. Depending on the camera model being used (orthographic, perspective), the object representation (e.g. point sets, line sets, contours, surface patches) and image data (e.g. corners, line segments, silhouettes) many different algorithms can be developed for different numerical estimation techniques (e.g. Kalman filters, gradient descent approaches, SVD decompositions), see [41, 18] for overviews. Before we introduce our approach for 2D-3D pose estimation of human models, we summarize basic notations, and previously developed point-based, contour-based and surface-based pose estimation algorithms, see [41].

3.1 Foundations

We start with mathematic concepts (Plücker lines and twists to model rigid body motions) needed for the pose problem. Then point-based and contour-based pose estimation algorithms are introduced.

3.1.1 Plücker lines

A 3D line $\boldsymbol{L}$ can be represented in Plücker form [41]. A 3-D Plücker line $\boldsymbol{L} = (\boldsymbol{n}, \boldsymbol{m})$ is given as 3-D (unit) vector $\boldsymbol{n}$ and 3-D moment $\boldsymbol{m} = \boldsymbol{x} \times \boldsymbol{n}$ for a given point $\boldsymbol{x}$ on the line.
An advantage of this representation is its uniqueness (apart from possible sign changes). This can be seen as follows: Let $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{L}$ with $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$. We choose $\lambda \in \mathbb{R}$ with

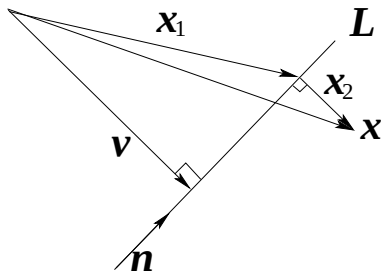

Figure 8: The comparison of the 3D point with the 3D line.

$\boldsymbol{x}_2 = \lambda\boldsymbol{n} + \boldsymbol{x}_1$. Then we have (note: $\boldsymbol{n} \times \boldsymbol{n} = 0$)

$$\boldsymbol{x}_2 \times \boldsymbol{n} \;\; = (\lambda\boldsymbol{n} + \boldsymbol{x}_1) \times \boldsymbol{n} \;\; = \lambda(\boldsymbol{n} \times \boldsymbol{n}) + \boldsymbol{x}_1 \times \boldsymbol{n} = \boldsymbol{x}_1 \times \boldsymbol{n}. \tag{3.1}$$

A second advantage is, that the incidence of a point $\boldsymbol{x}$ on a line $\boldsymbol{L} = (\boldsymbol{n}, \boldsymbol{m})$ can be expressed as

$$\boldsymbol{x} \in \boldsymbol{L} \;\; \Leftrightarrow \;\; \boldsymbol{x} \times \boldsymbol{n} - \boldsymbol{m} = 0. \tag{3.2}$$

This follows by a similar algebraic operation as in Equation 3.1.
A third advantage is, that Equation 3.2 provides us with a distance measure. Let $\boldsymbol{L} = (\boldsymbol{n}, \boldsymbol{m})$, with $\boldsymbol{m} = \boldsymbol{v} \times \boldsymbol{n}$ as shown in Figure 8, and $\boldsymbol{x} = \boldsymbol{x}_1 + \boldsymbol{x_2}$, with $\boldsymbol{x} \notin \boldsymbol{L}$ and $\boldsymbol{x}_2 \perp \boldsymbol{n}$. Then we have (note: $\boldsymbol{x}_1 \times \boldsymbol{n} = \boldsymbol{m}$, $\boldsymbol{x}_2 \perp \boldsymbol{n}$ and $\|\boldsymbol{n}\| = 1$)

$$\begin{aligned} \|\boldsymbol{x} \times \boldsymbol{n} - \boldsymbol{m}\| &= \|\boldsymbol{x}_1 \times \boldsymbol{n} + \boldsymbol{x}_2 \times \boldsymbol{n} - \boldsymbol{m}\| && (3.3) \\ &= \|\boldsymbol{x}_2 \times \boldsymbol{n}\| = \|\boldsymbol{x}_2\|. && (3.4) \end{aligned}$$

This means that $\boldsymbol{x} \times \boldsymbol{n} - \boldsymbol{m}$ in Equation 3.2 results in a (rotated) perpendicular error vector to line $\boldsymbol{L}$. This distance measure and its minimization is used for pose estimation.

3.1.2 Rigid motions

Every 3D rigid motion can be represented by a 4×4 matrix

$$\boldsymbol{M} \;\; = \;\; \begin{pmatrix} \boldsymbol{R}_{3\times3} & \boldsymbol{t}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 1 \end{pmatrix} \tag{3.5}$$

for a given rotation matrix $\boldsymbol{R}_{3\times3} \in SO(3)$, with $SO(n) := \{\boldsymbol{R} \in \mathbb{R}^{n\times n} : \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}, \quad det(\boldsymbol{R}) = +1\}$, and a translation vector $\boldsymbol{t}_{3\times1}$. By using homogeneous coordinates, a point $\boldsymbol{x}$ can be transformed by matrix multiplication $\boldsymbol{x}' = \boldsymbol{M}\boldsymbol{x}$. In fact, $\boldsymbol{M}$ is an element of the one-parametric 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; see [15, 34] for more details. The corresponding Lie algebra to $SE(3)$ is $se(3) = \{(\boldsymbol{v}, \omega) | \boldsymbol{v} \in \mathbb{R}^3, \omega \in so(3)\}$, with $so(3) = \{\boldsymbol{A} \in \mathbb{R}^{3\times3} | \boldsymbol{A} = -\boldsymbol{A}^T\}$. The elements in $se(3)$ are called *twists*, which can be denoted as

$$\hat{\xi} \;\; = \;\; \begin{pmatrix} \hat{\omega} & \boldsymbol{v} \\ 0_{3\times1} & 0 \end{pmatrix}, \tag{3.6}$$

with

$$\hat{\omega} = \begin{pmatrix} 0 & \text{-}\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ \text{-}\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{3.7}$$

A twist is sometimes written as vector

$$\xi = (\omega_1, \omega_2, \omega_3, v_1, v_2, v_3). \tag{3.8}$$

A twist ξ contains six parameters and can be scaled to $\theta\xi$ for a unit vector ω. To reconstruct a group action $\boldsymbol{M} \in SE(3)$ from a given twist, the exponential function $\exp(\theta\hat{\xi}) = \boldsymbol{M} \in SE(3)$ can be used. The parameter $\theta \in \mathbb{R}$ corresponds to the motion velocity (i.e., the rotation velocity and pitch). For varying θ, the motion can be identified as screw motion around an axis in space. This is also proven by Chasles Theorem [34] from 1830. Indeed, evaluating the exponential of a matrix is not trivial, but it can be calculated efficiently by using the Rodriguez formula [34],

$$\exp(\hat{\xi}\theta) \;=\; \begin{pmatrix} \exp(\theta\hat{\omega}) & (I - \exp(\hat{\omega}\theta))(\omega \times \boldsymbol{v}) + \omega\omega^T \boldsymbol{v}\theta \\ 0_{1\times 3} & 1 \end{pmatrix}, \text{for } \omega \neq 0 \tag{3.9}$$

with $\exp(\theta\hat{\omega})$ computed by calculating

$$\exp(\theta\hat{\omega}) \;=\; I + \hat{\omega}\sin(\theta) + \hat{\omega}^2(1 - \cos(\theta)). \tag{3.10}$$

Only sine and cosine functions of real numbers need to be computed.

3.1.3 Point-based pose estimation

For point-based pose estimation we combine the results of both previous subsections and introduce a gradient descent method. The idea is to reconstruct an image point to a projection ray $\boldsymbol{L} = (\boldsymbol{n}, \boldsymbol{m})$ and to claim incidence of the transformed 3D point $\boldsymbol{x}$ with the 3D ray,

$$(\exp(\theta\hat{\xi})\boldsymbol{x})_{3\times 1} \times \boldsymbol{n} - \boldsymbol{m} \;=\; 0. \tag{3.11}$$

Indeed, $\boldsymbol{x}$ is a homogeneous 4D vector, and after multiplication with the 4×4 matrix $\exp(\theta\hat{\xi})$ we neglect the homogeneous component (which is 1) to evaluate the cross product with $\boldsymbol{n}$. We now linearize the equation by using $\exp(\theta\hat{\xi}) = \sum_{k=0}^{\infty} \frac{(\theta\hat{\xi})^k}{k!} \approx \boldsymbol{I} + \theta\hat{\xi}$, with $\boldsymbol{I}$ as identity matrix. This results in

$$((\boldsymbol{I} + \theta\hat{\xi})\boldsymbol{x})_{3\times 1} \times \boldsymbol{n} - \boldsymbol{m} = 0 \tag{3.12}$$

and can be reordered into an equation of the form $\boldsymbol{A}\xi = \boldsymbol{b}$. Collecting a set of such equations (each is of rank two) leads to an overdetermined system of equations, which can be solved using, for example, the Householder algorithm. The Rodriguez formula can be applied to reconstruct the group action $\boldsymbol{M}$ from the estimated twist ξ. Then, the 3D points can be transformed and the process is iterated until the gradient descent approach converges.

Note that the projection rays only need to be reconstructed once, and can be reconstructed from orthographic, projective or even catadioptric cameras. The algorithm is very fast (e.g., it needs 2 ms on a standard Linux PC for 100 point correspondences). In [41] extensions to point-plane, line-plane constraint equations and kinematic chains are presented using Clifford algebra [46].
In this setting, the extension to multiple views is straightforward: we assume N images which are calibrated with respect to the same world coordinate system and are synchronised. For each camera the system matrices $\boldsymbol{A}_1 \dots \boldsymbol{A}_N$ and solution vectors $\boldsymbol{b}_1 \dots \boldsymbol{b}_N$ are generated. The equations are now bundled in one system $\boldsymbol{A} = (\boldsymbol{A}_1, \dots, \boldsymbol{A}_N)^T$ and $\boldsymbol{b} = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_N)^T$. Since they are generated for the same unknowns ξ, they can be solved simultaneously, i.e., the spatial errors of all involved cameras are minimized.

3.1.4 Pose estimation of free-form contours

We assume a 1-parametric closed curve in 3D space,

$$C(\phi) \;=\; (f^1(\phi), f^2(\phi), f^3(\phi))^T, \tag{3.13}$$

which is represented by a finite set of contour points

$$C_(n) = \{(f^1(n), f^2(n), f^3(n))^T : n = 0 \dots M-1\}. \tag{3.14}$$

The main idea is to interpret a 1-parametric 3D closed curve as three separate 1D signals which represent the projections of the curve along the x, y and z axis, respectively. Since the curve is assumed to be closed, the signals are periodic and can be analyzed by applying a 1D discrete Fourier transform (1D-DFT). The inverse discrete Fourier transform (1D-IDFT) enables us to reconstruct low-pass approximations of each signal. Subject to the sampling theorem, this leads to the representation of the 1-parametric 3D curve $C(\phi)$ as

$$C(\phi) = \sum_{m=1}^{3} \sum_{k=-N}^{N} \boldsymbol{p}_k^m \exp\left(\frac{2\pi k \phi}{2N+1} i\right). \tag{3.15}$$

The parameter m represents each dimension and the vectors $\boldsymbol{p}_k^m$ are phase vectors obtained from the 1D-DFT acting on dimension m. Using only a low-index subset of the Fourier coefficients results in a low-pass approximation of the object model which can be used to regularize the pose estimation algorithm.
For pose estimation we combine this parametric representation within an ICP-algorithm (iterative closest point) [54] to determine point correspondences between the image silhouette and the 3D contour.
The algorithm for pose estimation of free-form contours consists of iterating the following steps:

```
(a) Reconstruct the  projection rays from image points.
(b) Estimate the nearest point on the 3D contour to each projection ray.
(c) Estimate the contour pose by using this point/line correspondence set.
(d) Goto (b).
```

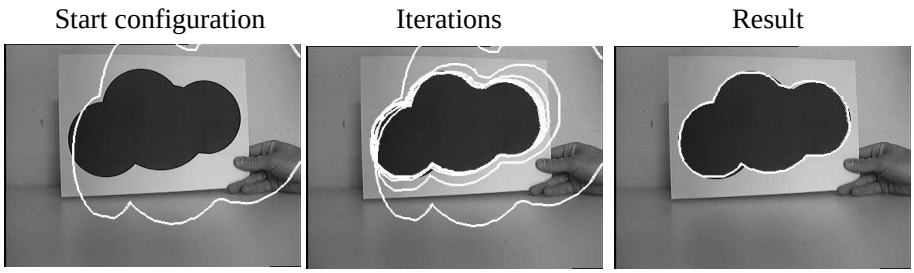

Figure 9: Pose results during iterated ICP-cycles.

For (a), we determine from the calibration the optical centre $\boldsymbol{c}$ and the image point in the world coordinate system. This is used to define the 3D Plücker line, see section 3.1.1. For (b) we use Equation 3.1 and sample along the contour. Part (c) is described in Section 3.1.3. Figure 9 shows an example for iterations. The algorithm usually converges within 10 iterations and we need 40ms on a standard Linux PC (1GHz) to estimate the pose of a free-form contour. In [41] we further use a Fourier-based object representation, which allows to use a low-pass approximation for pose estimation and for adding successively higher frequencies during the iteration. It is a multi-resolution method and helps to avoid local minima during iteration. This modification is explained for the 2D case in the next section and used as standard method. Since the constraint equations express a geometric distance measure in 3D space, it is further possible to detect and eliminate outliers, see [41].

3.1.5 Pose estimation of free-form surfaces

We assume a two-parametric surface [8] of the form

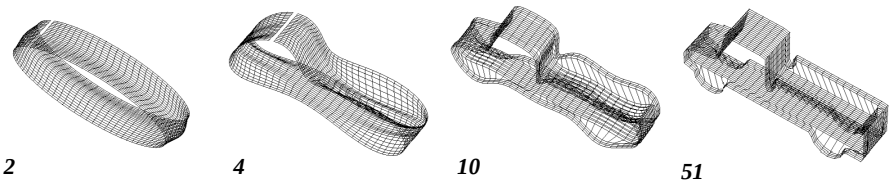

Figure 10: A sequence of different low-pass approximations of an object model.

$$F(\phi_1, \phi_2) \quad = \quad (f^1(\phi_1, \phi_2), f^2(\phi_1, \phi_2), f^3(\phi_1, \phi_2))^T \tag{3.16}$$

defined by three functions $f^i(\phi_1, \phi_2) : \mathbb{R}^2 \rightarrow \mathbb{R}$ acting on the base vectors. The idea behind a two-parametric surface is to assume two independent parameters ϕ_1 and ϕ_2 which sample the 2D surface in 3D space. In fact, we are using a mesh model of the object [8]. For a finite number of sampled points $f^i(n_1, n_2)$ ($n_1 \in [-N_1, N_1]; n_2 \in [-N_2, N_2]$; $N_1, N_2 \in \mathbb{N}$, $i = 1, \ldots, 3$) on the surface, we interpolate the surface by using a 2D discrete Fourier transform (2D-DFT) and then apply an inverse 2D discrete Fourier transform (2D-IDFT) for each base vector separately. Subject to a proper sampling,

the surface can therefore be written as a series expansion of the form

$$F(\phi_1, \phi_2) = \sum_{k_1=-N_1}^{N_1} \sum_{k_2=-N_2}^{N_2} \begin{pmatrix} F^1(k_1,k_2) \\ F^2(k_1,k_2) \\ F^3(k_1,k_2) \end{pmatrix} \exp\left(\frac{2\pi k_1 \phi_1}{2N_1+1} i\right) \exp\left(\frac{2\pi k_2 \phi_2}{2N_2+1} i\right) \quad (3.17)$$

$$F^j(k_1,k_2) = \frac{1}{(2N_1+1)(2N_2+1)} \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} f^i(n_1,n_2) \exp\left(-\frac{2\pi k_1 n_1}{2N_1+1} i\right) \exp\left(-\frac{2\pi k_2 n_2}{2N_2+1} i\right) \quad (3.18)$$

Figure 10 gives example approximation levels of a car model.

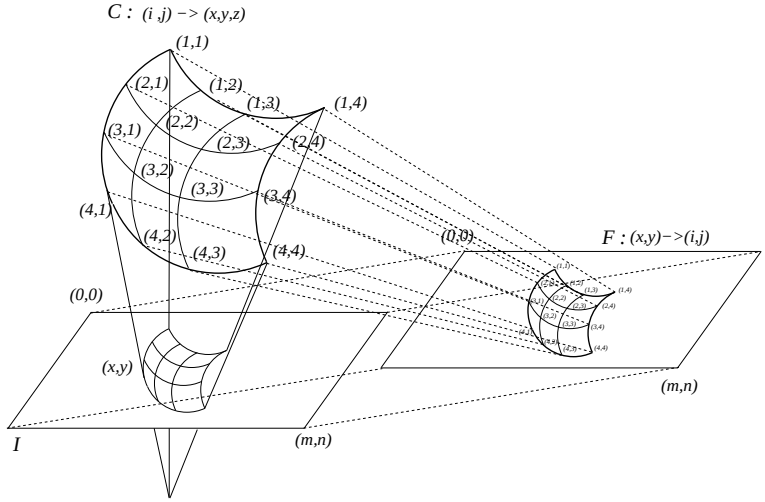

Figure 11: To determine the 3D position of a 2D image node, a field F is used as look-up table, which stores the relationship between pixels and the 3D mesh. $C(F(x, y))$ gives the 3D coordinates of a node at pixel position (x, y) in the image.

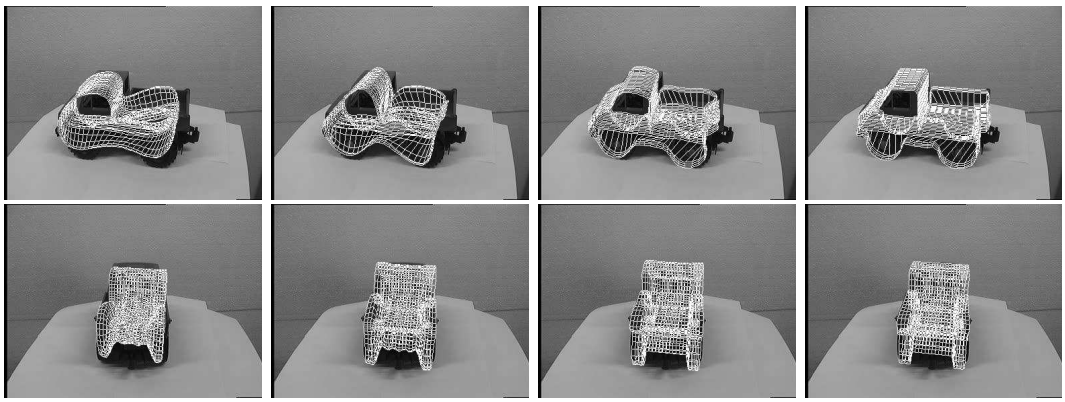

Figure 12: Pose results of low-pass contours during the ICP cycle.

We assume a properly extracted silhouette of an object in a given image, using a segmentation algorithm based on level-set functions, which is explained in section 4.1. There

is a need to express tangentiality between the surface and the reconstructed projection rays, and there is also a need to express a distance measure within our description. For this requirement, we decided to use the 3D rim of the surface model which is tangential with respect to the camera coordinate system. Here our tracking assumption comes into account: We choose our initial projection matrix (or the projection matrix from the last frame) as view-port and consider the rim of the surface model (i.e., the occluding boundary of the object) with respect to this view-port. To obtain this, we project the 3D surface onto a virtual image. Then the 2D contour is calculated and from the image contour the 3D rim of the surface model is reconstructed. To get the 3D rim, there is a need to get from the image of a node point to its 3D value. This is done with the help of a look-up table F, see Figure 11. First we assume a mesh $C(i,j) \rightarrow (x,y,z)$ which gives the 3D coordinates of the surface node for the two sample parameters (i,j). This mesh is projected with the projection matrix into a virtual image I. The model is projected in the virtual image with connecting line segments between points on the surface nodes and the nodes in another gray-scale value. This virtual image is used as a look-up table: we can detect the 3D surface point for a given surface node on the image with the help of a 2D field F and function C, since $C(F(x,y))$ gives the 3D coordinates of the node's image point (x,y). To obtain the 3D rim points we use a contour algorithm which follows the image of the mesh model by a recursive procedure. Then the nodes of the mesh model are collected (this is easy, since they are projected into another gray-scale value than the connecting line segments) from which the corresponding 3D rim is calculated with the help of F. The rim model is then applied to our contour-based pose estimation algorithm, see section 3.1.4. Since the aspects of the surface model are changing during the ICP-cycles, a new rim will be estimated after each cycle to deal with occlusions within the surface model. The convergence behavior of the silhouette-based pose estimation algorithm is shown in Figure 12. It can be seen that we refine pose results by adding successively higher frequencies during the iteration.

3.2 Pose estimation of human models

We now introduce how to couple kinematic chains within the surface model and present a pose estimation algorithm which estimates the pose and angle configurations simultaneously. A surface is given in terms of three 2-parametric functions with respect to

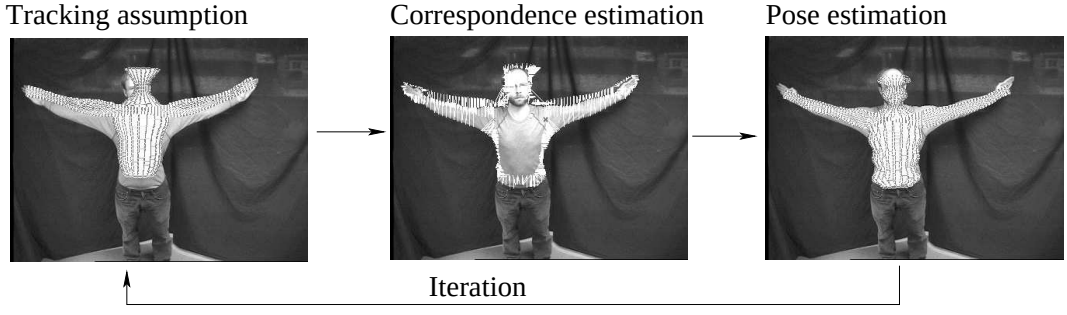

Figure 13: The basic algorithm: Iterative correspondence and pose estimation.

the parameters ϕ_1 and ϕ_2. Furthermore, we assume a set of joints J_i. By using an addi-

tional function $\mathcal{J}(\phi_1, \phi_2) \rightarrow [J_i | J_i : i\text{th. joint}]$, we are able to give every node a joint list along the kinematic chain. Note, that we use $[,]$ and not $\{,\}$ since the joints are given in order along the kinematic chain. Since the arms contain two kinematic chains (for the left and right arm separately), we introduce a further index to separate the joints on the left arm from the ones on the right arm. The joints themselves are represented as objects in an extra field (a look-up table) and their parameters can be accessed immediately from the joint index numbers. Furthermore, it is possible to transform the location of the joints in space (as clarified in section 2). For pose estimation of a point X_i^n attached to the nth joint, we generate a constraint equation of the form

$$(\exp(\theta_n \hat{\xi_n}) \dots \exp(\theta_1 \hat{\xi_1}) \exp(\theta \hat{\xi}) X_i^n)_{3\times 1} \times \boldsymbol{n}_i - \boldsymbol{m}_i \;\; = \;\; 0$$

which is linearized in the same way as the rigid body motion itself. It leads to three linear equations with the six unknown pose parameters and n unknown joint angles. Collecting a sufficient number of equations leads to an overdetermined system of equations.
The basic pose estimation algorithm is visualized in figure 13: The image processing method to gain the silhouette information of the person is described in section 4.1. Then we project the surface mesh into a virtual image and estimate its 3D contour. Each point on the 3D contour carries a given joint index. Then we estimate the correspondences by using an ICP-algorithm, generate the system of equations, solve it, transform the object and its joints and iterate this procedure. During iteration we start with a low-pass object representation and refine it by using higher frequencies. This helps to avoid local minima during iteration.

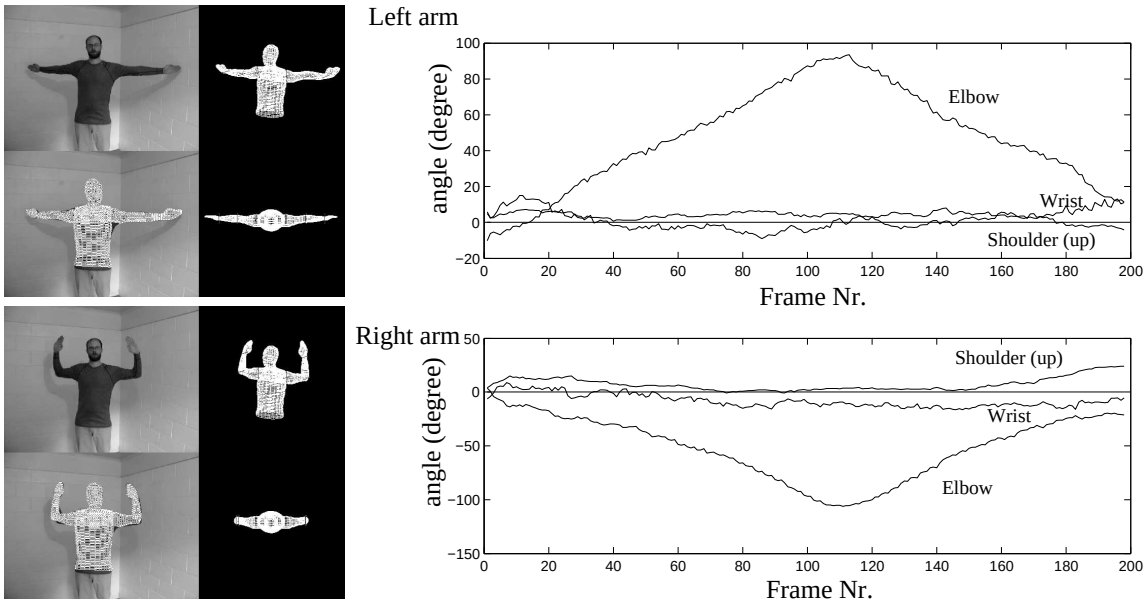

Figure 14: Left: pose results with a 6 DOF kinematic chain. Right: Angles of the left and right arm during the tracked image sequence.

First results of the algorithm are shown in the left images of figure 14: The figure contains two pose results and shows the original image and the overlaid projected 3D pose on each quadrant. The other two images show the estimated joint angles in a

virtual environment to visualize the error between the real motion and the estimated pose. The tracked image sequence contains 200 images. Note, that in this sequence we use just three joints on each arm and neglect the shoulder (back) joint. The right diagram of figure 14 shows the estimated angles of the joints during the image sequence. The angles can easily be identified with the sequence. Since the movement of the body is continuous, the estimated curves are also relatively smooth.
We further extend the model to a 8DOF kinematic chain and add a joint on the shoulder that allows the arms to move backwards and forwards. Some results are shown in figure 15. The observation of the pose overlaid with the image data is also good, but in the simulation environment it is visible, that the estimated joints are very noisy. The reason

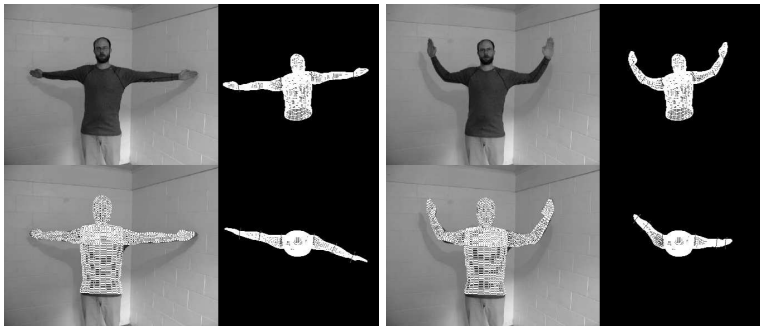

Figure 15: First pose results with a 8 DOF kinematic chain.

for the depth sensitivity lies in the used image information: Figure 16 shows two images

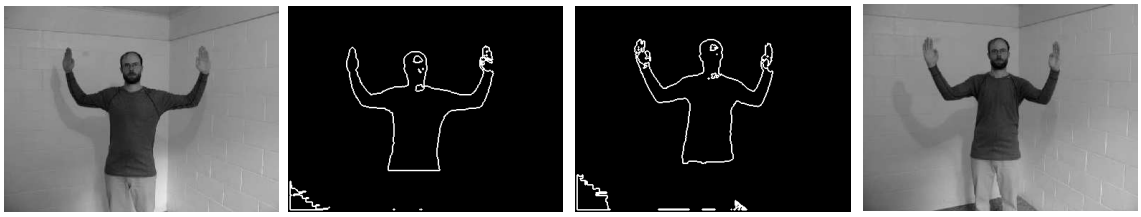

Figure 16: The silhouette for different arm poses of the kinematic chain.

of a human with different arm positions. But as can be seen, the estimated silhouettes look quite similar. This means, that the used image features are under-determined in their interpretation as 3D pose configuration. This problem can not be solved in an algorithmic manner and is of geometric nature. We are well aware that this result contradicts to some papers in the literature which deal with monocular image sequences. But our experience is, that such approaches can only work for reduced kinematic chains (containing not all degrees of freedom of a human being) or by using further a-priori knowledge (e.g., regarding the motion pattern, camera configuration, etc.).
To overcome this problem we propose a multi-view setup. The basic idea is, that the geometric non-uniquenesses can be avoided by using different cameras observing the scene from different perspectives. Since we reconstruct rays from image points, we have to calibrate the cameras with respect to the same world coordinate system. Following this, it is not important for the algorithm from which camera a ray is reconstructed and we are able to combine the equations from both cameras in one system of equations

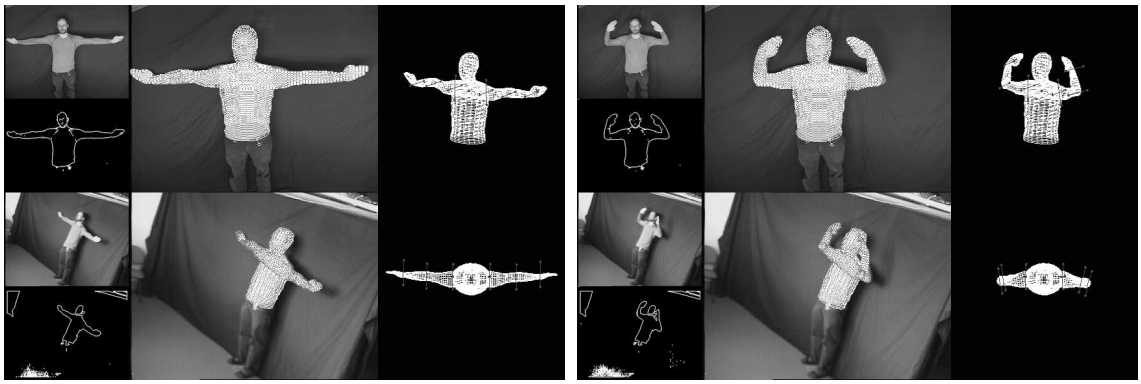

Figure 17: Example images of a stereo experiment.

and estimate the pose and arm angles simultaneously, see section 3.1.3. Figure 17 shows example images of a stereo sequence. In each segment, the images on the left show the original and filtered image of each camera. The images in the middle show pose results in both cameras and the images on the right show the pose results in a virtual environment. As can be seen, the results have been improved. Figure 18 shows estimated angles during the image sequence. As can be seen, they fit to the real motion of the human.

4 A Human motion Tracking system

At this stage we have laid the foundations for a human motion tracking system. But indeed the experiments are quite academic and improvements are necessary to obtain a system which can handle and analyse e.g. sports movements. For this reason we will now introduce image segmentation based on level-set functions to replace the previously used grey-value thresholding, and morphing approaches to gain more realistic human models and motion patterns. In section 5 these techniques will be applied together with a sampling method in a four-camera set up and a model of much higher complexity (21 degrees of freedom).

4.1 Image segmentation based on level set functions

Image segmentation usually means to estimate boundaries of objects in an image. This task can become very difficult, since noise, shading, occlusion or texture information between the object and the background may distort the segmentation or even make it impossible. Our approach is based on image segmentation based on level sets [37, 9, 11, 6]. 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. The segmentation should maximize the total a-posteriori probability given the probability densities p_1 and p_2 of Ω_1 and Ω_2, i.e., pixels are assigned to the most probable region according to the Bayes rule. Ideally, the boundary between both regions should be as small as possible. This can be

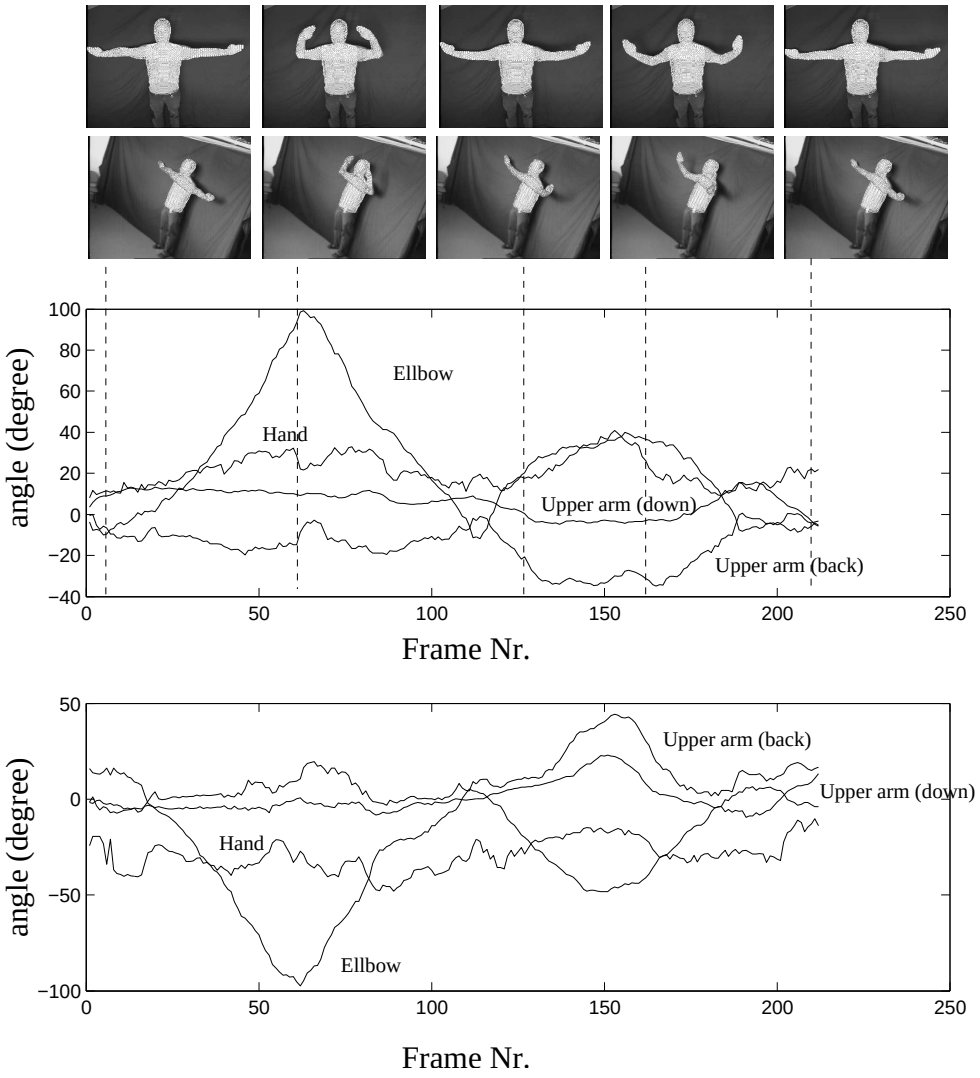

Figure 18: Estimated angles during the image sequence.

expressed by the following energy functional that is sought to be minimized:

$$E(\Phi, p_1, p_2) \quad = \quad -\int_\Omega \left(H(\Phi) \log p_1 + (1 - H(\Phi)) \log p_2 + \nu |\nabla H(\Phi)| \right) dx \qquad (4.19)$$

where $\nu > 0$ is a weighting parameter and $H(s)$ is a regularized version of the Heaviside function, e.g. the error function. Minimization with respect to the region boundary represented by Φ can be performed according to the gradient descent equation

$$\partial_t \Phi = H'(\Phi) \left(\log \frac{p_1}{p_2} + \nu \, \mathrm{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right) \qquad (4.20)$$

where $H'(s)$ is the derivative of $H(s)$ with respect to its argument. The probability densities p_i are estimated according to the *expectation-maximization principle*. Having the level set function initialized with some contour, the probability densities within the two regions are estimated by the gray value histograms smoothed with a Gaussian kernel K_σ and its standard deviation σ.

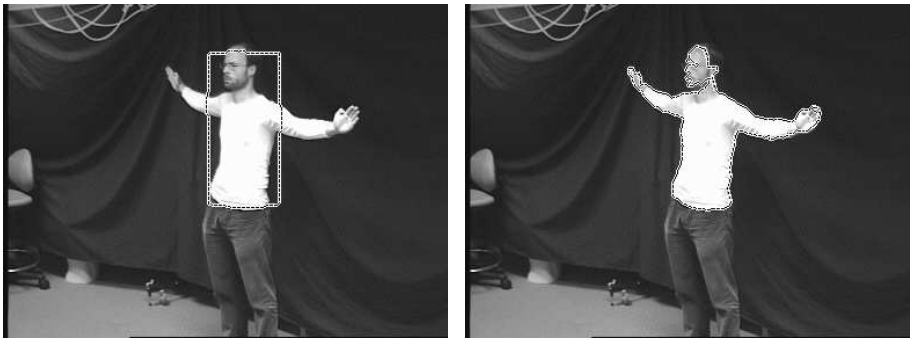

Figure 19: Silhouette extraction based on level set functions. Left: Initial segmentation. Right: Segmentation result.

This rather simple and fast approach is sufficient for our laboratory set-up, though it is also conceivable to apply more elaborated region models including texture features. Figure 19 shows an example initialization and the final contour. As can be seen, the body silhouette is well extracted, but there are some deviations in the head region, due to the dark hair. Such inaccuracies can be compensated from the pose estimation procedure. For our algorithm we can make a tracking assumption. Therefore, we initialize the silhouette with the pose of the last frame which greatly reduces the number of iterations needed. The implementation is fast; the algorithm needs 50 ms per frame and 200 ms image processing time in a four-camera setup.

4.2 Morphing approaches

The synthetic modelling of a human body and its motion simulation is mostly treated in computer graphics, e.g. for animation of avatars. It is common to model an articulated body by layers of skeletons, ellipsoidal meta-balls (to simulate muscles) and polygonal surface patches (to model the skin) [14]. It is further common to use the Denavit-Hartenberg parametrization to model joints on a skeleton by using a tree structured hierarchy of manipulators, robotic joint-link parameters and joint angle constraints (max, min, stiffness, etc.) [10, 21]. To gain a realistic motion model, joint dependent local deformation (JLD) operators are used to deform the skin surface and to model muscles and fatty tissue layers [1, 53]. Since we are using free-form surface models to represent human beings, we do not need a hierarchic representation (sticks, ellipses, etc.) since the joints are directly coupled to the surface mesh. For realistic motion modelling we introduce morphing approaches on the surface mesh itself. This results in a compact and unified representation which can be coupled immediately with the surface based pose estimation algorithm.

4.2.1 Joint motions

Joints along the kinematic chain can be modelled as screws with no pitch. In [41] it is shown by using Clifford algebras that a twist, representing a joint, corresponds to a scaled (dual) Plücker line $\Psi = \theta \underline{\boldsymbol{L}}$ in space, which gives the location of the general rotation. Since the conformal geometric algebra allows to transform higher order entities,

such as lines, circles or spheres in the same manner as points, this formulation allows for an elegant approach to move joints in space, see [42]. In matrix calculus a joint can be transformed as follows: Let $\boldsymbol{M} = \exp(\theta\hat{\nu}) = (\boldsymbol{R}, \boldsymbol{t})_{4\times4}$ a rigid body motion and $\xi = (v_1, v_2, v_3, \omega_1, \omega_2, \omega_3)^T$ a scaled twist with $\|\omega\|_2 = \|(\omega_1, \omega_2, \omega_3)^T\|_2 = 1$ representing a joint in the model. This joint needs to be transformed according to a rigid body motion. Since the vector $v = (v_1, v_2, v_3)^T$ is determined from $v = o \times \omega$ for o being a point on the joint axis, we can move the joint axis (the twist) by computing

$$\xi' \;=\; (v', \omega') = ((\boldsymbol{R}o + \boldsymbol{t}) \times \boldsymbol{R}\omega, \boldsymbol{R}\omega). \tag{4.21}$$

This leads to a transformed joint axis. By using an exponential form, $\boldsymbol{M} = \exp(\theta\hat{\nu})$, the value θ steers the amount of joint motion and allows for a continuous transformation. Note, that joints do not always undergo a linear (or circular) transformation between positions in space. They sometimes undergo a higher order trajectory, i.e. the knee joint [38]. Transforming joints along such a trajectory can be done by following the twist-representation-of-shape principle, introduced in [47].
We will now introduce two surface morphing techniques, based on a global and a local model.

4.2.2 Global surface interpolation

We assume two free-form surfaces given as two-parametric functions as follows:

$$F_1(\phi_1, \phi_2) = \sum_{i=1}^{3} f_1^i(\phi_1, \phi_2)\mathbf{e}_i \quad \text{and} \quad F_2(\phi_1, \phi_2) = \sum_{i=1}^{3} f_2^i(\phi_1, \phi_2)\mathbf{e}_i \tag{4.22}$$

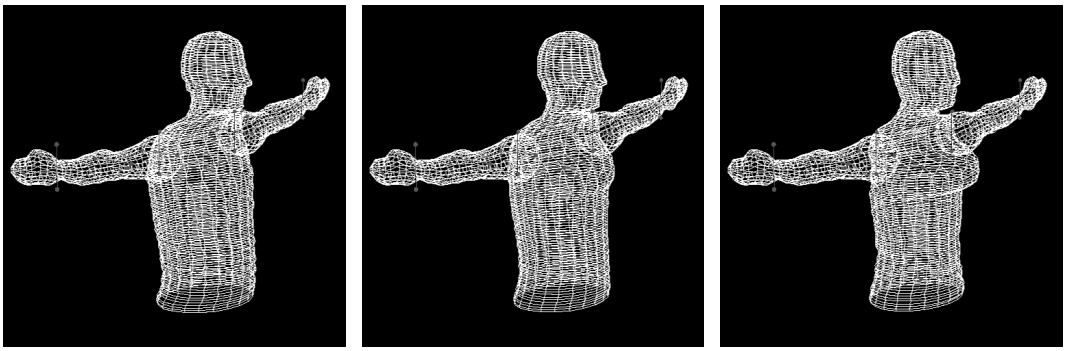

Figure 20: Morphing of a male into a female torso.

For a given parameter $t \in [0 \ldots 1]$, the surfaces can be linearly interpolated by evaluating

$$F_t(\phi_1, \phi_2) \;=\; \left(\sum_{i=1}^{3} f_1^i(\phi_1, \phi_2)\mathbf{e}_i\right) t + \left(\sum_{i=1}^{3} f_2^i(\phi_1, \phi_2)\mathbf{e}_i\right)(1-t) \tag{4.23}$$

We perform a linear interpolation along the nodes, and this results in the following:

$$F_t(\phi_1, \phi_2) \;=\; \begin{cases} \sum_{i=1}^{3} f_1^i(\phi_1, \phi_2)\mathbf{e}_i \;=\; F_1(\phi_1, \phi_2) & \text{, for} \quad t = 1 \\ \sum_{i=1}^{3} f_2^i(\phi_1, \phi_2)\mathbf{e}_i \;=\; F_2(\phi_1, \phi_2) & \text{, for} \quad t = 0 \end{cases} \tag{4.24}$$

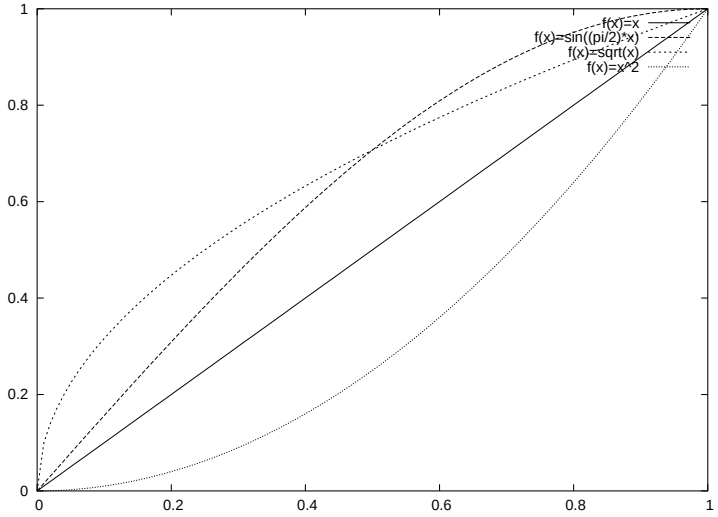

Figure 21: Different weighting functions during interpolation.

Figure 20 shows examples of morphing a male into a female torso. Note that we are only morphing surfaces with known and predefined topology. This means that we have knowledge about the correspondences between the surfaces, and morphing is realized by interpolating the corresponding nodes on the mesh.
The linear interpolation can be generalized by using an arbitrary function $\omega(t)$ with the property

$$\omega(t) = \begin{cases} 0 & \text{, for } t = 1 \\ 1 & \text{, for } t = 0. \end{cases} \tag{4.25}$$

Then, an interpolation is still possible by using

$$F_t(\phi_1, \phi_2) = \left(\sum_{i=1}^{3} f_1^i(\phi_1, \phi_2)\mathbf{e}_i \right) \omega(t) + \left(\sum_{i=1}^{3} f_2^i(\phi_1, \phi_2)\mathbf{e}_i \right) (1 - \omega(t)) \tag{4.26}$$

Figure 21 shows different possible functions which result in different interpolation dynamics. Using the square root function for weighting leads to a faster morphing at the beginning, which slows down at the end, whereas squared weighting leads to a slower start and a faster ending. Therefore, we can use non-linear weighting functions to gain a natural morphing behavior dependent on the joint dynamics.
Figure 22 shows a comparison of the non-modified model (right) with a morphed joint-transformed model (left). It can be seen that the shoulder joint is moving down and in-wards during motion, and, simultaneously, the surface of the shoulder part morphes. The amount of morphing and joint transformation is steered through the angle of the shoulder (up) joint (left and right, respectively). As can be seen, the left mesh appears more natural than the right one.

4.2.3 Local surface morphing

The use of radial basis functions for local morphing is common practice for modelling facial expressions. The basic idea is as follows: we move a node on the surface mesh and

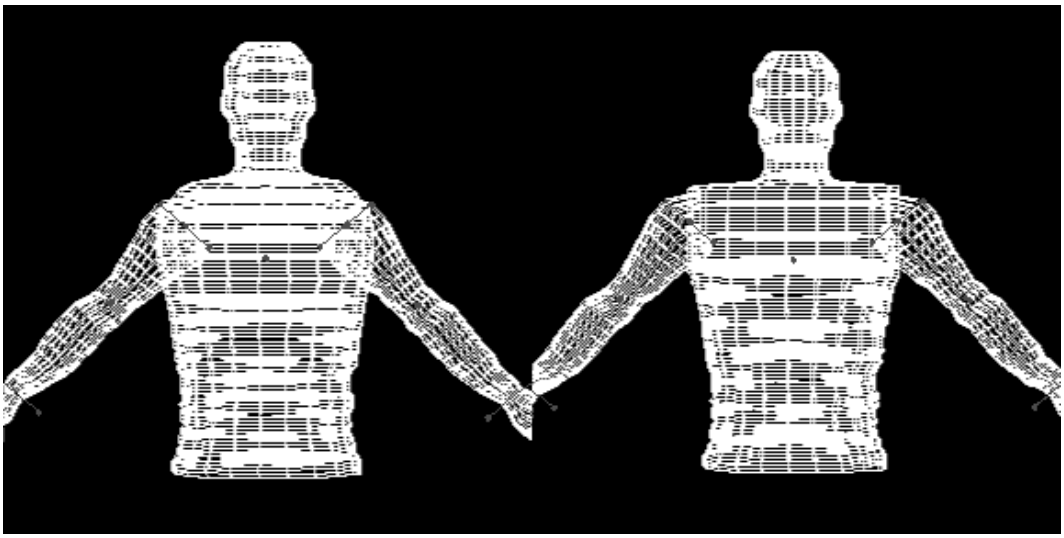

Figure 22: Different arm positions of the morphed joint-transformed model (left) and non-modified model (right).

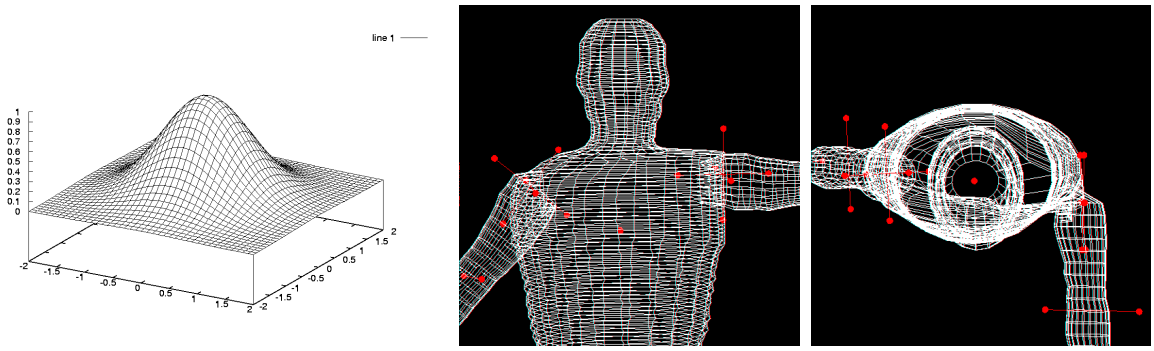

Figure 23: Left: A 2D-radial basis function. Right: Double surface morphing on the shoulder.

we move the neighboring nodes in a similar manner, but decreasingly with increasing distance to the initiating node. The classic equation for a radial basis function is

$$r(x,y) \quad = \quad \exp\left(-\frac{(x-c_x)^2}{r_x}\right)\exp\left(-\frac{(y-c_y)^2}{r_y}\right) \tag{4.27}$$

with the centre (c_x, c_y) and the radius (r_x, r_y). The values $(c_x, c_y) = (0, 0)$ and $(r_x, r_y) = (1, 1)$ lead to the classic Gaussian form as shown on the left of Figure 23.
The coupling of a radial basis function with the surface mesh leads to

$$F_R(\phi_1,\phi_2) \quad = \quad \sum_{i=1}^{3} f^i(\phi_1,\phi_2)\mathbf{e}_i + \lambda\mathbf{e}_3 r(\phi_1,\phi_2) \tag{4.28}$$

The amount of morphing is steered through the value of the radial basis function at the mesh position. It is dependent on (ϕ_1, ϕ_2) and different for each node. In equation (4.28) we model a deformation along the $\mathbf{e}_3$-axis, but it can be any orientation, and

the Gaussian function can be arbitrarily scaled. Therefore, we steer the amount of morphing through the joint angle θ_1 of the shoulder (back) joint. This means, if the shoulder is not moving forwards or backwards, we will not have any morphing, but the more the arm is moving, the larger will be the amount of morphing. It is further possible to deform the radial basis function to allow a realistic morphing in the presence of bones or ligaments.
In contrast to global morphing, local approaches have the advantage that they can be used more easily in the context of multiple morphing patches. For example, simultaneous shoulder morphing up or down and forwards or backwards is hardly possible with a global approach, but simple with a local one.
Figure 23 shows a typical radial basis function to realize local surface morphing on the left. The images on the right show a double morphing on the shoulder: moving the arms up or down and forwards or backwards leads to a suited deformation of the shoulder patch and a similar motion of the joint locations.

5 Experiments

For our experiments we use a human upper (and later lower) torso model which consists of three parametric free-form surface patches. The surface patches undergo a local deformation, depending on the joint values. The model itself consists of 3500 knots. They are attached to seven joints on each arm and one back-bone joint. Adding six unknowns for the unknown rigid body motion of the torso, we deal with 21 degrees of freedom.

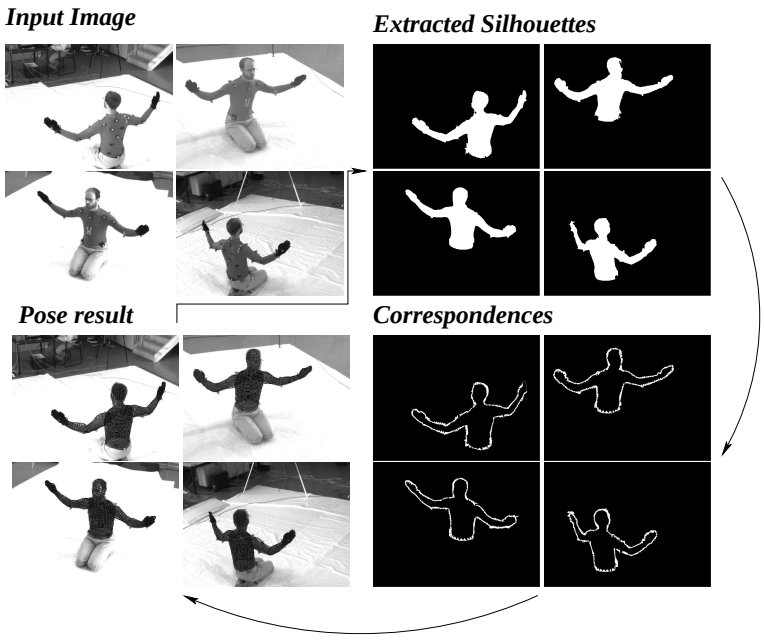

Figure 24: The capture system consists of iterating the following steps: Segmentation, correspondence estimation, pose estimation.

The capture system is shown in figure 24. The system uses as input data the object model and four calibrated (synchronised) cameras, and assumes a tracking configuration

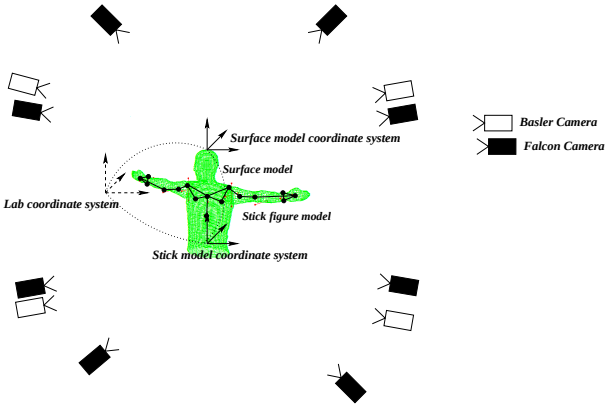

Figure 25: The coordinate systems in the lab setup.

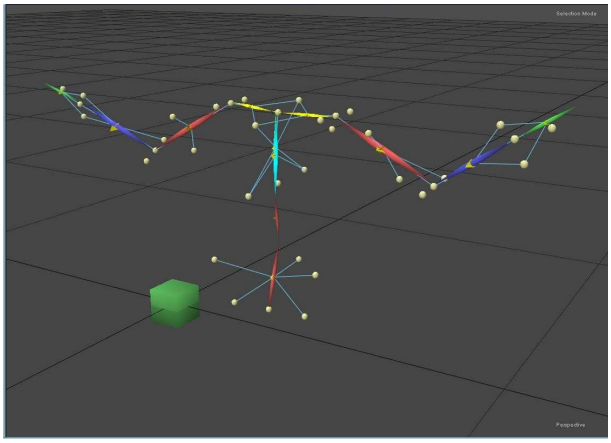

Figure 26: The coordinate systems of the markers in the lab setup. Within the Eva 3.2.1 software, the markers are (manually) connected to a skeleton model.

of the 3D model. For this pose configuration, silhouettes are extracted by using level-set functions and global region statistics. Then correspondences are established between the surface rims and the contours which are used for a pose update. This new pose leads to another starting configuration for the image segmentation and is iterated till both, segmentation and pose estimation, converge.
A disadvantage of many studies is that the only feedback one receives is visual feedback of the pose provided by overlaying the pose with the image data. To enable a quantitative error analysis, we use a commercial marker based tracking system for a comparison. We use the Motion Analysis software [33] with an 8-Falcon-camera system. For data capture we use the Eva 3.2.1 software and the Motion Analysis Solver Interface 2.0 for inverse kinematics computing [33]. In this system the subject has to have retro-reflective markers attached to specific anatomical landmarks. Around each camera is a strobe light led ring and a red-filter is in front of each lens. This gives very strong image signals of the markers in each camera. These are treated as point markers which are reconstructed in the eight-camera system. The system is calibrated by using a wand-calibration method. Due to the filter in front of the images we had to use a second camera set-up which provides *real* image data. This camera system is calibrated

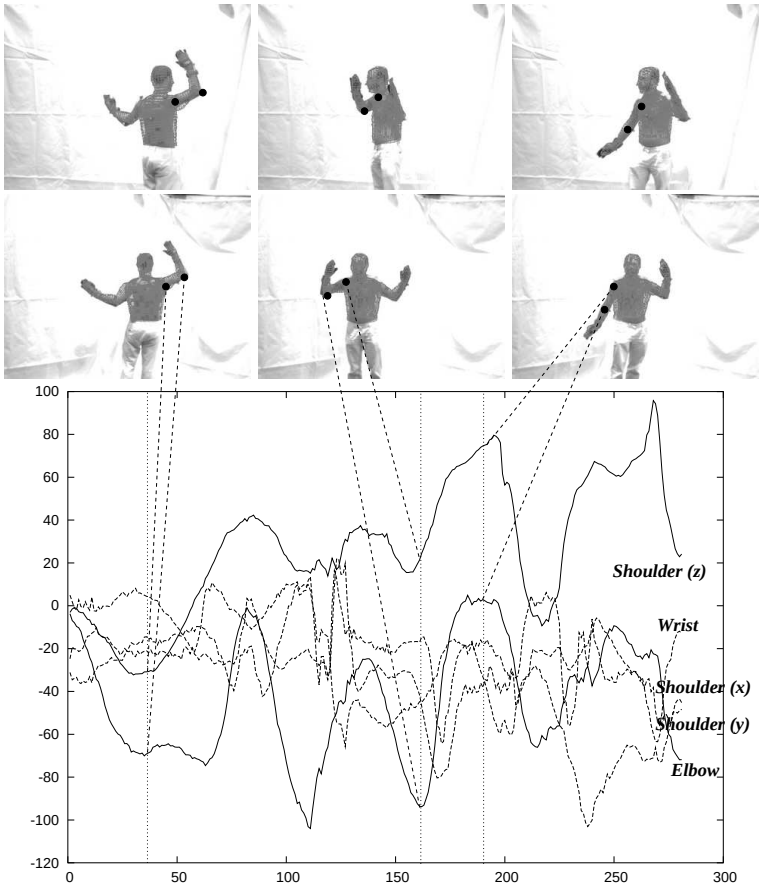

Figure 27: Subset of the estimated angles of the right arm within the four-camera sequence.

by using a calibration cube. After calibration, both camera systems are calibrated with respect to each other. Then we generate a stick-model from the point markers including joint centers and orientations. This results in a complete calibrated set-up we use for a system comparison as visualized in figure 25. some reconstructed markers of the Motion Analysis system are shown in figure 26. The skeletons are connected manually in Eva 3.2.1 by connecting the reconstructed points. Using this setup we then grabbed a series of test sequences.

Figure 27 shows the estimated angles of the right arm together with joint locations marked in some images of such a four camera sequence. In this sequence, the person performs a 180 degree rotation and moves towards the cameras while moving the arms. Furthermore there is noise on the back, face, and arms, which leads to inaccuracies during contour extraction. The algorithm is able to handle the occlusions dynamically. Clearly the estimated angles correspond well to the scene.

Figure 28 shows the second test sequence, where the person is just moving the arms forwards and backwards. The diagram on the right side shows the estimated angles of the right elbow. The marker results are given as dotted lines and the silhouette results in solid lines. The overall error between both angles diagrams is 2.3 degrees, including the tracking failure between frames 200 till 250.

Figure 29 shows the third test sequence, where the person is performing a series of push-

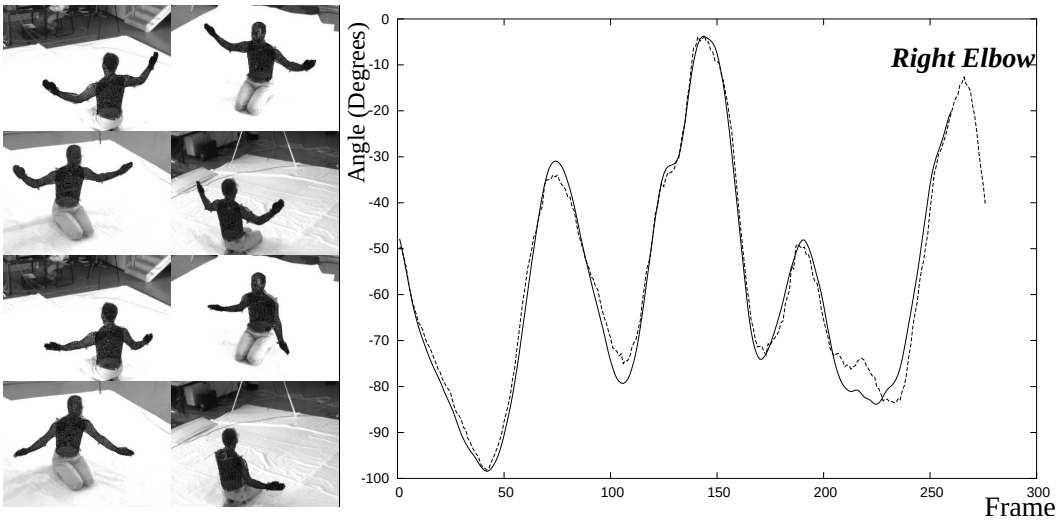

Figure 28: Tracked arms: The angle diagrams show the elbow values of the Motion Analysis system (dotted) and the silhouette system (solid).

ups. Here the elbow angles are much more characteristic and also well comparable. The overall error is 1.7 degrees. Both sequences contain partial occlusions in certain frames which can be handled by the algorithm. Figure 30 shows example images of a tracked sit-up sequence. During the sequence, parts of the arms occlude with the torso. The back bone joint is crucial for a successful tracking of the sequence.
In [40] eight biomechanical measurement systems are compared (including the Motion Analysis system). The author also performed a rotation experiment which shows, that the root mean square errors are typically within three degrees. Our error measures fit in this range quite well.
Figure 31 shows pose results of a leg model. Here the model consists of three surface patches, representing the hip and both legs. Each leg has six degrees of freedom and three segments. The person is performing walking and jumping sequences, which can be tracked from the system. Here the aim is to demonstrate that the algorithm is able to track completely different kinematic models by using the same source code, just with a modified model as input. Figure 32 shows a visualization of the captured motion data in a virtual environment.
The sequences were grabbed in 60fps, which is already the limit of the grabbing system. This is demonstrated in figure 33: sometimes during the sequence, the images are corrupted, due to the frame rate. The algorithm is able to handle such outliers by using the available information and by keeping the non-visible joints constant. A few frames later, the model is again properly fitted to the image data.

6 Summary

This contribution presents a model based human motion estimation system. We started with a model generation system, which uses a set of input images to automatically reconstruct a free-form surface model of a human upper torso by using a shape-from-silhouettes approach. Furthermore, joint locations are determined automatically in

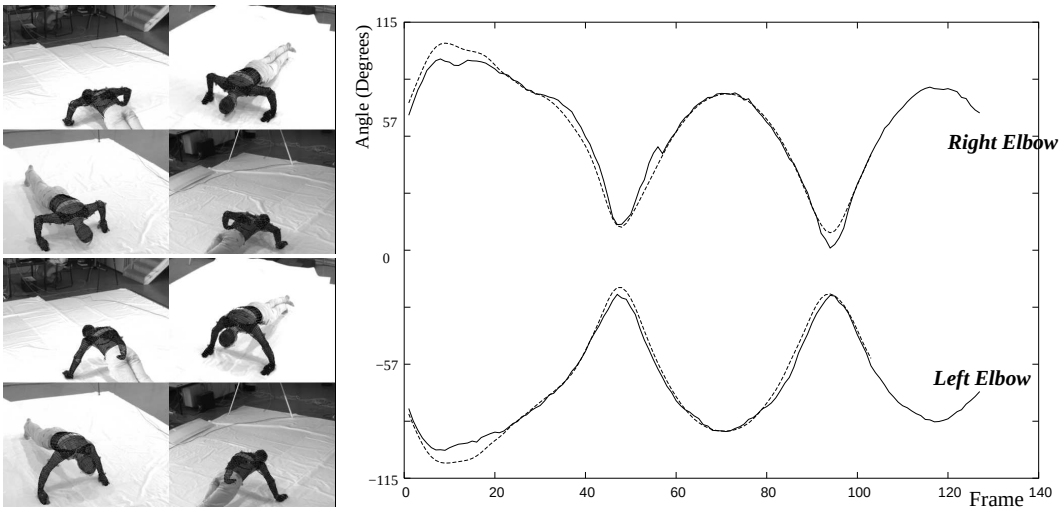

Figure 29: Tracked Push-ups: The angle diagrams show the elbow values of the Motion analysis system (dotted) and the silhouette system (solid).

Figure 30: Example images of a Tracked sit-up sequence. During the sequence, parts of the arms occlude with the torso. The back bone joint is crucial for a successful tracking of the sequence.

addition to a texture for the surface mesh. Then we present foundations on point-contour- and surface-based pose estimation and introduce the basic set-up for multi-view silhouette based human motion estimation. For applications in sport movement analysis, we further introduce morphing and joint transformation techniques to gain more realistic human upper torso models. Such an advanced model is used in a system for silhouette based human motion estimation. The presented motion estimation system contains silhouette extraction based on level sets, a correspondence module, which relates image data to model data and a pose estimation module. This system is used for a variety of experiments: Experimental results are presented for different camera setups (containing one to four cameras) and we estimate the pose configurations of a human upper torso model with 21 degrees of freedom in two frames per second. We also discuss degenerated cases for silhouette based human motion estimation (i.e. for monocular image sequences). Furthermore, a comparison of the motion estimation system with a commercial marker based tracking system is performed to gain a quantitative

Figure 31: Pose results for a leg model. The person performs walking, jumping and scissors, resulting in partial occlusions and crossings of leg parts.

Figure 32: Visualization of the captured motion data in a virtual environment. The right image shows, that the right foot, standing on the floor, is nearly perfect on the ground plane.

error analysis. The results show the applicability of the system for marker-less sport movement analysis. Finally we present experimental results on tracking leg models and show the robustness of our algorithms even for corrupted image data.

References

[1] Allen B., Curless B. and Popovic Z. Articulated body deformation from range scan data. In *Proceedings 29th Annual Conf. Computer Graphics and Interactive Techniques*, San Antonio, Texas, pp. 612 - 619, 2002.

[2] Besl P.J. The free-form surface matching problem. *Machine Vision for Three-Dimensional Scenes, Freemann H. (Ed.)*, pp. 25-71, Academic Press, 1990.

[3] Burt P.J. and Andelson E.H. A multiresolution spline with aplication to image mosaics. *ACM Trans. on Graphics*, II, No 4, pp. 217-236, 1983.

[4] Bray J. Markerless Based Human Motion Capture: A Survey Vision and VR Group, Brunel University, unpublished draft, `www.visicast.co.uk/members/publications/Publications.html` (Accessed March 2004)

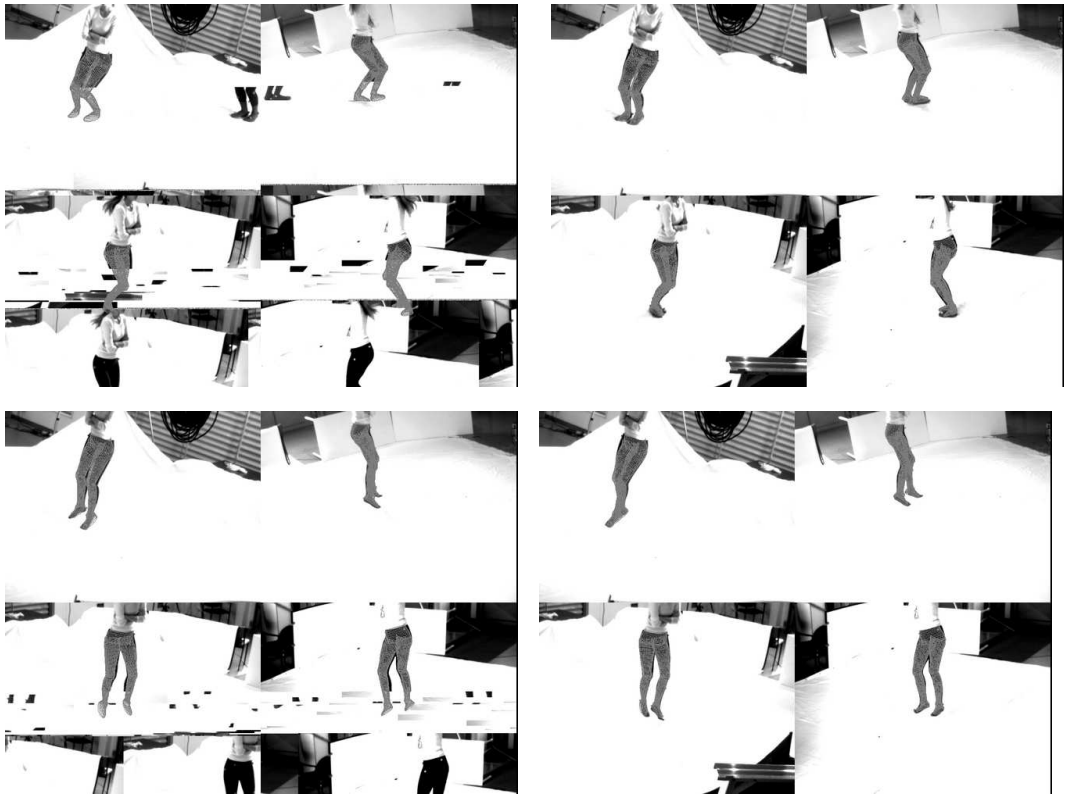

Figure 33: Left: Corrupted images during grabbing. Right: Tracking result two frames later. The algorithm is able to handle such image corruptions.

[5] Bregler C. and Malik J. Tracking people with twists and exponential maps. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, pp. 8-15, 1998.

[6] Brox T., Rousson M., Deriche R., Weickert J. Unsupervised segmentation incorporating colour, texture, and motion In *Computer Analysis of Images and Patterns*, Springer LNCS 2756, N.Petkov, M.A.Westenberg (Eds.), pp. 353-360, Proc. CAIP 2003 Conference, Groningen, The Netherlands, 2003.

[7] Carranza J., Theobalt C., Magnor M.A. and Seidel H.-P. Free-viewpoint video of human actors in *Proceedings of ACM SigGraph 2003*, San Diego, USA, pp. 569-577, 2003.

[8] Campbell R.J. and Flynn P.J. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding (CVIU)*, Vol. 81, pp. 166-210, 2001.

[9] Caselles V., Catté F., Coll T. and Dibos F. A geometric model for active contours in image processing. *Numerische Mathematik*, 66:1–31, 1993.

[10] Chadwick J.E., Haumann D.R. and Parent R.E. Layered construction for deformable animated characters. *Computer Graphics*, Vol. 23, No. 3, pp. 243-252, 1989.

[11] Chan T. and Vese L. An active contour model without edges. In M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 141–151. Springer, 1999.

[12] Chetverikov D. A simple and efficient algorithm for detection of high curvature points. *In: Computer Analysis of Images and Patterns*, N. Petkov and M.A. Westenberg (Eds.) Springer-Verlag Berlin, LNCS 2756, pp. 746-753, 2003.

[13] Cremers D., Kohlberger T., and Schnörr Ch. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, No. 36, Vol. 9, pp. 1929-1943, 2003.

[14] Fua P., Plänkers R., and Thalmann D. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding*, Vol. 81, No. 3, pp.285-302, March 2001.

[15] Gallier J. *Geometric Methods and Applications For Computer Science and Engineering.* Springer-Verlag, New York Inc., 2001.

[16] Gavrilla D.M. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, Vol. 73 No. 1, pp. 82-92, 1999.

[17] Gavrilla D.M. and Davis L. 3D model-based tracking of humans in action, a multi-view approach. In *proceedings of IEEE computer Society conference on Computer Vision and Pattern Recognition*, pp. 73-80, 1996.

[18] Goddard J.S. Pose and Motion Estimation From Vision Using Dual Quaternion-Based Extended Kalman Filtering. *University of Tennessee, Knoxville*, Ph.D. Thesis, 1997.

[19] Grimson W. E. L. Object Recognition by Computer. *The MIT Press, Cambridge, MA*, 1990.

[20] He L., Generation of Human Body Models. Master Thesis, The University of Auckland, 2005.

[21] Herda L., Urtasun R., Fua P. Hierarchical Implicit Surface Joint Limits to Constrain Video-Based Motion Capture. *Proceedings of the European Conference on Computer Vision, ECCV '04*, , Part II, T. Pajdla and J. Matas (Eds.), Springer-Verlag, Berling Heidelberg, LNCS 3022, pp. 405-418, Prague, 2004.

[22] Human Motion Conferences
- Nonrigid and articulated motion workshop, Puerto Rico, 1998.
- HuMo, Workhop on human motion, Austin, Texas, 2000.
- SigGraph 1997-2005.
- 1st-9th International symposium on 3D analysis of human movement, 1991-2006.
- 3rd Int. Biomechanics of the lower limb in health disease and rehabilitation, Salford, UK, 2005.
- Human motion - understanding, modeling, capture and animation, Schloss Dagstuhl, Germany 2006.
(and more ...)

[23] Hiltion A., Beresford D., Gentils T. Smith R. and Sun W. Vitual people: capturing human models to populate virtual worlds. In *Proc. Computer Animation*, pp. 174-185, 1999.

[24] Horprasert T., Harwood D. and Davis L.S. A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection *In: International Conference on Computer Vision*, FRAME-RATE Workshop, Kerkyra, Greece, 1999. Available at
`www.vast.uccs.edu/~tboult/FRAME`
`/Horprasert/HorprasertFRAME99.pdf` (Last accessed February 2005).

[25] Kakadiaris I. and Metaxas D. Three-dimensional human body model acquisition from multiple views. *International Journal on Computer Vision*, Vol. 30 No. 3, pp. 191-218, 1998.

[26] Klette G. A Comparative Discussion of Distance Transformations and Simple Deformations in Digital Image Processing. *Machine Graphics and Vision* Vol. 12, No. 2, pp. 235-356, 2003.

[27] Klette R. and Rosenfeld A. Digital Geometry–Geometric Methods for Digital Picture Analysis *Morgan Kaufmann*, San Francisco, 2004.

[28] Klette R., Schlüns K. and Koschan A. Computer Vision. Three-Dimensional Data from Images. *Springer*, Singapore, 1998.

[29] Lee W. Gu J. and Magnenat-Thalmann N. Generating animatable 3D virtual humans from photographs. *Computer Graphics Forum*, Vol. 19, No. 3, pp. 1-10, 2000.

[30] Mikic I., Trivedi M, Hunter E, and Cosman P. Human body model acquisition and tracking using voxel data *International Journal of Computer Vision (IJCV)*, Vol. 53, Nr. 3, pp. 199–223, 2003.

[31] Moeslund T. B. and Granum E. A survey of computer vision based human motion capture. *Computer Vision and Image Understanding*, 2002.

[32] Mori G. and Malik J. Estimating human body configurations using shape context matching. In *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part III*, Heyden A., Sparr G., Nielsen M. and Johansen P. (Eds.), LNCS 2352, Springer-Verlag Heidelberg, pp. 666-680, 2002.

[33] Motion Analysis Corporation `www.motionanalysis.com` last accessed February 2005.

[34] Murray R.M., Li Z. and Sastry S.S. A Mathematical Introduction to Robotic Manipulation. *CRC Press*, 1994.

[35] Magnenat-Thalmann N., Seo H. and Cordier F. Automatic Modeling of Virtual Humans and Body Clothing. In *Journal of Computer Science and Technology*, Chinese Academy of Sciences, Beijing, China (publisher), Vol. 19 No. 5, pp.575-584, September 2004.

[36] Orourke J. *Computational Geometry in C*. Cambridge University Press, Cambridge, UK, 1998.

[37] Osher S. and Sethian J. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, Vol.79, pp. 12-49, 1988.

[38] Parenti-Castelli V., Leardini A, Gregorio R. and O'Connor J. On the Modeling of Passive Motion of the Human Knee Joint by Means of Equivalent Planar and Spatial Parallel Mechanisms. *Autononmous Robots*, Kluwer, Vol. 16, pp. 219-232, 2004.

[39] Park J. and Park S. and Aggarwal J.K. Human Motion Tracking by Combining View-Based and Model Based Methods for Monocular Video Sequences *Korean Journal of Information Processing*, Vol. 10, No. 6, pp. 657-664, October 2003.

[40] Richards J. The measurement of human motion: A comparison of commercially available systems *Human Movement Science*, Vol. 18, pp. 589-602, 1999.

[41] Rosenhahn B. Pose estimation revisited. Technical Report TR-0308, Institute of Computer Science, University of Kiel, Germany, Oct. 2003.

[42] Rosenhahn B. and Klette R. Geometric algebra for pose estimation and surface morphing in human motion estimation *Tenth International Workshop on Combinatorial Image Analysis (IWCIA)*, R. Klette and J. Zunic (Eds.), LNCS 3322, pp. 583-596, 2004, Springer-Verlag Berlin Heidelberg. Auckland, New Zealand,

[43] Rosenhahn B., Perwass Ch. and Sommer G. Pose Estimation of Free-form Contours *International Journal of Computer Vision (IJCV)*, Vol. 62, No 3, pp. 267-289, 2005.

[44] Rosenhahn B. and Sommer G. Pose Estimation of Free-form Objects. *Proceedings of the European Conference on Computer Vision, ECCV '04*, Part I, T. Pajdla and J. Matas (Eds.), Springer-Verlag, Berlin Heidelberg, LNCS 3021, pp. 414-427, Prague, 2004.

[45] Rosenhahn B., Kersting U., Smith A., Gurney J., Brox T. and Klette R. A system for marker-less human motion estimation *Pattern Recognition 2005, DAGM*, W. Kropatsch, R. Sablatnig and A. Hanbury (Eds.), Springer-Verlag, Berlin Heidelberg, LNCS 3663, pp. 230-237, Wien, 2005.

[46] Sommer G., editor. Geometric Computing with Clifford Algebra. *Springer Verlag*, Berlin, 2001.

[47] Sommer G., Rosenhahn B. and Perwass Ch. Twists - An Operational Representation of Shape. *IWMM 2004 - Computer Algebra and Geometric Algebra with Applications*, H. Li and P.J. Olver and G. Sommer (Eds.), Springer-Verlag Berlin Heidelberg, LNCS 3519, pp. 278-297, 2005.

[48] Shi Y. and Sun H. *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards.* CRC Press, Boca Raton, FL, USA, 1999.

[49] Simi Reality Motion Systems. `www.simi.com/en/` last accessed June 2005.

[50] Theobalt C., Carranza J. and Magnor M.A. Enhancing silhouette-based human motion capture with 3D motion fields. *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, Canmore, Canada, 185-193, 2003.

[51] Tsai R. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. 1986. Available at `http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html`. Last accessed at 6.4.2004.

[52] Motion Capture Systems from Vicon Peak. `www.vicon.com` last accessed June 2005.

[53] You L. and Zhang J. J. Fast Generation of 3D Deformable Moving Surfaces. *IEEE Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol.33, No. 4, pp. 616-625, 2003.

[54] Zang Z. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, Vol. 13, No. 2, pp. 119-152, 1999.

[55] Zerroug, M. and Nevatia, R. Pose estimation of multi-part curved objects. *Image Understanding Workshop (IUW)*, pp. 831-835, 1996