Monocular Body Pose Estimation by Color Histograms and Point Tracking

Daniel Grest, Dennis Herzog and Reinhard Koch

Christian-Albrechts-University Kiel, Germany
Multimedia Information Processing
Email: {grest,dherzog,rk}@mip.informatik.uni-kiel.de

Abstract. Accurate markerless motion capture systems rely on images that allow segmentation of the person in the foreground. While the accuracy of such approaches is comparable to marker based systems, the segmentation step makes strong restrictions to the capture environment, e.g. homogenous clothing or background, constant lighting etc. In our approach a template model is fitted to images by an Analysis-by-Synthesis method, which doesn't need explicit segmentation or homogenous clothing and gives reliable results even with non-static cluttered background.
Keywords: optical motion capture, articulated objects, pose estimation

1 Introduction

Motion capture and body pose estimation are very important tasks in many applications. Motion capture products used in the film industry or for computer games are usually marker based to achieve high quality and fast processing. A lot of research is devoted to make markerless motion capture applicable. Accurate markerless systems rely on images that allow segmentation of the person in the foreground. While the accuracy of such approaches is comparable to marker based systems [13, 5], the segmentation step makes strong restrictions to the capture environment, e.g. homogenous clothing or background, constant lighting, camera setups that cover a complete circular view on the person etc. Most systems create first a visual hull from the segmented images and fit a template model afterwards by minimizing an objective function.

Our approach also fits a template model by minimizing correspondences, however it doesn't need explicit segmentation or homogenous clothing and gives reliable results even with non-static cluttered background. Additionally, less views are sufficient, as the underlying motion and body model is directly incorporated in the image processing step. While motion capture from stereo depth images already allows such complex environments [8], we present here results from a single camera view, that show the efficiency of our approach even with complex movements.

Capturing human motion by pose estimation of an articulated object is done in many approaches and is motivated from inverse kinematic problems in robotics[10]. Solving the estimation problem by optimization of an objective function is also very common [13, 9, 11]. Silhouette information is usually part of this function, that tries to minimize the difference between the model silhouette and the silhouette of the real person either by background segmentation [13, 11] or image gradient [12, 9].

Matching feature points from one image to the next in a sequence is also a useful cue to estimate the body pose as done in [3]. However this cue alone will introduce drift, because an error in the estimation accumulates over time.

The above mentioned approaches to markerless motion capture all have in common, that the underlying movement capabilities of a human (body parts are connected by joints) are formulated directly in the optimization, while the degrees of freedom and the projection model differ. In [3] a scaled orthographic projection approximates the full perspective camera model and in [13] the minimization of 2D image point distances is approximated by 3D-line 3D-point distances.

While some kind of template body model is common in most approaches, adaption of body part sizes of these template during the motion estimation is also possible like in [12]. Others assume the body model is known and fitted offline beforehand. This reduces the degrees of freedom (DOF) for the optimization significantly and allows fast and accurate estimation. In most applications it is possible to measure the size of the person before the capturing, like in sport motion analysis or in capturing motion for movies or video games.

Our approach incorporates silhouette information and point tracking using the full perspective camera model. Different cues result in different types of optimization equations. Our method minimizes errors, where they are observed and makes no approximations to the movement or projection model. Additionally it allows analytical derivations of the optimization function, which speeds up the calculation by more accuracy and less function evaluations than numerical derivatives. Therefore the approach is fast enough for real-time applications in the near future as we process images already in less than a second.

2 Body and Movement Model

Depending on the kind of work different body models are used for the estimation process. The models range from simple stick figures [3] over models consisting of scalable spheres (meta-balls) [12] to linear blend skinned models [2].

We use models with movement capabilities as defined in the MPEG4 standard. However not all 180 DOF are estimated, but a subset of up to 30 parameters. The MPEG4 description allows a simple change of body models and reanimation of other models with the captured motion data. An example of one model used in this work is shown in (1). The model for a specific person is obtained by silhouette fitting of a template model as described in [7].

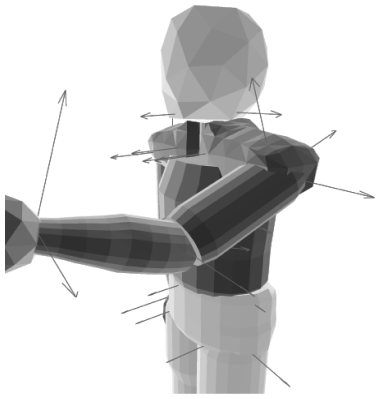

Fig. 1. The body model with rotation axes shown as arrows

The MPEG4 body model is a mixture of articulated objects. The movement of a point, e.g. on the hand, may therefore be expressed as a concatenation of rotations [8]. As the rotation axes are known, e.g. the flexion of the elbow, the rotation has only one degree of freedom (DOF), the angle around

that axis. In addition to the joint angles there are 6 DOF for the position and orientation of the object within the global world coordinate frame. For an articulated object with p joints the transformation may be written according to [8] as:

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = (\theta_x, \theta_y, \theta_z)^T + \left(R_x(\theta_\alpha) \circ R_y(\theta_\beta) \circ R_z(\theta_\gamma) \circ R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta_1) \circ \cdots \circ R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta_p)\right)(\boldsymbol{x}) \tag{1}$$

where $(\theta_x, \theta_y, \theta_z)^T$ is the global translation, R_x, R_y, R_z are the rotations around the global x, y, z-axes with Euler angles α, β, γ and $R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta_i), i \in \{1..p\}$ denotes the rotation around the known axis with angle θ_i. The axis is described by the normal vector $\boldsymbol{\omega_i}$ and the point $\boldsymbol{q_i}$ on the axis with closest distance to the origin.

Equation (1) gives the position of a point $\boldsymbol{x}$ on a specific segment of the body (e.g. the hand) with respect to joint angles $\boldsymbol{\theta}$ and an initial body pose.

The first derivatives of $f(\boldsymbol{\theta}, \boldsymbol{x})$ with respect to $\boldsymbol{\theta}$ give the Jacobian matrix $J_{ki} = \frac{\partial f_k}{\partial \theta_i}$. The Jacobian for the movement of the point $\boldsymbol{x}$ on an articulated object is

$$J = \begin{bmatrix} 1 & 0 & 0 & & & & & & \\ 0 & 1 & 0 & \frac{\partial f}{\partial \theta_\alpha} & \frac{\partial f}{\partial \theta_\beta} & \frac{\partial f}{\partial \theta_\gamma} & \frac{\partial f}{\partial \theta_1} & \cdots & \frac{\partial f}{\partial \theta_p} \\ 0 & 0 & 1 & & & & & & \end{bmatrix} \tag{2}$$

with the simplified derivative at zero:

$$\left.\frac{\partial f}{\partial \theta_i}\right|_0 = \left.\frac{\partial R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta_i)}{\partial \theta_i}\right|_0 = \boldsymbol{\omega}_i \times (\boldsymbol{x} - \boldsymbol{q}_i) = \boldsymbol{\omega}_i \times \boldsymbol{x} - \boldsymbol{\omega}_i \times \boldsymbol{p}_i \tag{3}$$

where $\boldsymbol{p}_i$ is an arbitrary point on the rotation axis. The term $\boldsymbol{\omega}_i \times \boldsymbol{p}_i$ is also called the momentum. The simplified derivative at zero is valid, if relative transforms in each iteration step of the *Nonlinear Least Squares* are calculated and if all axes and corresponding point pairs are given in world coordinates.

2.1 Projection

If the point $\boldsymbol{x} = (x_x, x_y, x_z)^T$ is observed by a pin-hole camera and the camera coordinate system is in alignment with the world coordinate system, the camera projection may be written as:

$$p(\boldsymbol{x}) = \begin{pmatrix} s_x \frac{x_x}{x_z} + c_x \\ s_y \frac{x_y}{x_z} + c_y \end{pmatrix} \tag{4}$$

where s_x, s_y are the pixel scale (focal length) of the camera in x- and y-direction, and $(c_x, c_y)^T$ is the center of projection in camera coordinates.
We now combine $f(\boldsymbol{\theta}, \boldsymbol{x})$ and $p(\boldsymbol{x})$ by writing $g(s_x, s_y, c_x, c_y, \boldsymbol{\theta}, \boldsymbol{x}) = p(f(\boldsymbol{\theta}, \boldsymbol{x}))$. The partial derivatives of g can now be easily computed using the chain rule. The resulting Jacobian reads as follows:

$$\begin{aligned} J &= \begin{bmatrix} \frac{\partial g}{\partial s_x} & \frac{\partial g}{\partial s_y} & \frac{\partial g}{\partial c_x} & \frac{\partial g}{\partial c_y} & \frac{\partial g}{\partial \theta_x} & \frac{\partial g}{\partial \theta_y} & \frac{\partial g}{\partial \theta_z} & \frac{\partial g}{\partial \theta_\alpha} & \cdots & \frac{\partial g}{\partial \theta_p} \end{bmatrix} \\ &= \begin{bmatrix} \frac{f(\boldsymbol{\theta})_x}{f(\boldsymbol{\theta})_z} & 0 & 1 & 0 & \frac{s_x}{f(\boldsymbol{\theta})_z} & 0 & s_x \frac{-f(\boldsymbol{\theta})_x}{(f(\boldsymbol{\theta})_z)^2} & \frac{\partial g_x}{\partial \theta_\alpha} & \cdots & \frac{\partial g_x}{\partial \theta_p} \\ 0 & \frac{f(\boldsymbol{\theta})_y}{f(\boldsymbol{\theta})_z} & 0 & 1 & 0 & \frac{s_y}{f(\boldsymbol{\theta})_z} & s_y \frac{-f(\boldsymbol{\theta})_y}{(f(\boldsymbol{\theta})_z)^2} & \frac{\partial g_y}{\partial \theta_\alpha} & \cdots & \frac{\partial g_y}{\partial \theta_p} \end{bmatrix} \end{aligned} \tag{5}$$

and

$$\frac{\partial g}{\partial \theta_i} = \begin{pmatrix} \frac{\partial \left(s_x \frac{f_x}{f_z} \right)}{\partial \theta_i} \\ \\ \frac{\partial \left(s_y \frac{f_y}{f_z} \right)}{\partial \theta_i} \end{pmatrix} = \begin{pmatrix} \frac{s_x \left(\frac{\partial f_x}{\partial \theta_i} f(\boldsymbol{\theta})_z - f(\boldsymbol{\theta})_x \frac{\partial f_z}{\partial \theta_i} \right)}{(f(\boldsymbol{\theta})_z)^2} \\ \\ \frac{s_y \left(\frac{\partial f_y}{\partial \theta_i} f(\boldsymbol{\theta})_z - f(\boldsymbol{\theta})_y \frac{\partial f_z}{\partial \theta_i} \right)}{(f(\boldsymbol{\theta})_z)^2} \end{pmatrix} \tag{6}$$

The partial derivatives $\frac{\partial f}{\partial \theta_i}, i \in \{\alpha, \beta, \gamma, 1, .., p\}$ are given in equation (2) and $f(\boldsymbol{\theta}) = (f_x, f_y, f_z)^T$ is short for $f(\boldsymbol{\theta}, \boldsymbol{x})$. Note that $f(\boldsymbol{\theta})$ simplifies to $\boldsymbol{x}$, if $\boldsymbol{\theta}$ is zero.

We minimize the distance between the projected 3D model point with its corresponding 2D image point, while in [13] the 3D-difference of the viewing ray and its corresponding 3D point is minimized. The minimization in 3D space is not optimal, if the observed image positions are disturbed by noise, as shown in [15], because for 3D points, which are farther away from the camera, the error in the optimization will be larger as for points nearer to the camera, which leads to a biased pose estimate due to the least squares solution. In [15] a scaling value was introduced, which down weights correspondences according to their distance to the camera, which is in fact very close to the equation (5).

Another relation exists to the work of [1], where the first 10 partial derivatives of Equation (5) are used for estimating the internal and external camera parameters by nonlinear optimization. This allows full camera calibration from (at best) five 2D-3D correspondences or pose from 3 correspondences. An implementation of it with an extension to the *Levenberg-Marquardt* algorithm[4], which ensures an error decrease in each iteration, is available for public in our open-source C++ library [6].

3 Estimating Body Pose

Assume a person, whose body model is known, is observed by a pinhole camera with known internal parameters at some time t resulting in an image I_t. Let $X = \{\boldsymbol{x_0}, \boldsymbol{x_1}, .., \boldsymbol{x_N}\}$ be the set of model points and $X' = \{\boldsymbol{x'_0}, \boldsymbol{x'_1}, .., \boldsymbol{x'_N}\}$ the set of their projected image points. Additionally assume that the pose of the person is known at that time, such that the projected body model aligns with the observed image as in the second image of figure 6. If the person now moves a little and an image I_{t+1} is taken, it is possible to capture the movement by estimating the relative joint angles of the body between the frames I_t and I_{t+1}. If the image points $\hat{X}'$ in I_{t+1} that correspond to X' are found, e.g. by some matching algorithm, the pose estimation problem is to find the parameters $\hat{\boldsymbol{\theta}}$ that best fit the transformed and projected model points to $\hat{X}'$, which can be formulated as follows:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \left| g(\boldsymbol{\theta}, \boldsymbol{x_i}) - \hat{\boldsymbol{x}}'_{\boldsymbol{i}} \right|^2 \tag{7}$$

This problem is known as *Nonlinear Least Squares* and can be solved by *Newton's Method* [4]. We use the *Gauss-Newton Method* [4], which doesn't require the the second derivatives of $g(\boldsymbol{\theta}, \boldsymbol{x_i})$.

The solution is found by iteratively solving the following equation:

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} - (J^T J)^{-1} J^T \left(G(\boldsymbol{\theta_t}, \boldsymbol{X}) - \hat{\boldsymbol{X}}' \right) \tag{8}$$

Here the Jacobian matrix J consists of all partial derivatives for all N points, where the Jacobian for a single point is given in equation (5). In case of convergence the final solution $\hat{\boldsymbol{\theta}}$ is found.

To get the initial pose, the user has to position the model manually. Because the depth is difficult to measure from a single view, markers on the floor give the user helpful information, where to position the model. Small errors in the manual positioning are not crucial, because the silhouette correspondences are correcting small errors.

3.1 Tracking of Image Features

For the estimation from above it is necessary to have correspondences between 2D image points and 3D model points. These can be calculated by tracking 2D features from one image to the next. Because we assume that the initial pose of the person is known as in figure 6, it is possible to get the relation between the image of the real person and the 3D model point by intersection of the feature's viewing ray and the 3D model surface using the known projection matrix. Then the same feature point has to be found within the next image and gives the necessary 2D image 3D model point correspondence. We use corners, which are

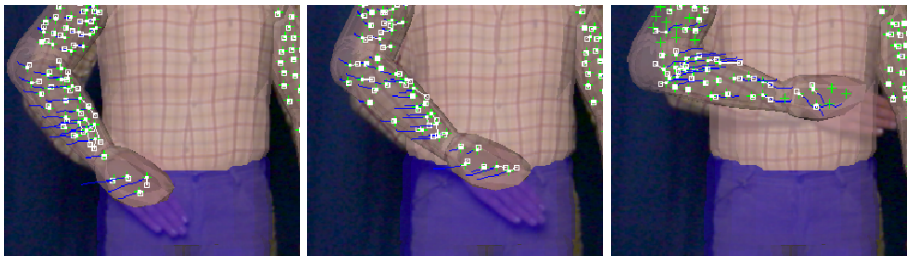

Fig. 2. Tracking of point features (Cross marks). Boxes indicate a tracked corner. The movement of a corner over the last frames is shown as a black line.

tracked with the KLT feature tracker [14]. Tracking point features allows us to capture motion, which wouldn't be possible from the silhouette alone, e.g. an arm moving in front of the body like in figure 2. However as also visible, the motion estimation is not very accurate, because the assumption, that the correspondence of model and image point is given by projection of the model point, leads to an error accumulation over time. As visible the elbow position drifts away to the left. To stabilize the estimation we combine the corner tracking with silhouette information as described in the next section.

Because the movement of the arms and legs is usually larger than of the torso, we distribute feature points equally on the body of the person, such that there are enough correspondences for estimation of the arm joint angles. Limiting the number and distributing the position of the tracked points is also necessary for fast computation. We achieve this by projecting the 3D model into the real image using OpenGL similar to [8], which gives directly the relation of feature points and visible body segments. In this way we can distribute the feature points equally over the visible segments.

4 Correspondences by Silhouette

To compensate the drift we add silhouette information to our estimation. This is achieved by calculating additional 2D-3D correspondences for the model silhouette and the silhouette of the real person. In contrast to [13] we don't utilize explicit segmentation of the images in fore- and background, but use the predicted model silhouette to search for corresponding points on the real silhouette. Previous work like [9] already took this approach by searching for a maximum grey value gradient in the image in the vicinity of the model silhouette. However we experienced that the gray value gradient alone gives often erroneous correspondences, especially if the background is heavily cluttered and the person wears textured clothes.

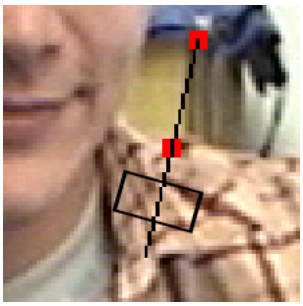

Fig. 3. Correspondence search along the normal.

Therefore we also take color information into account. As the initial pose is known, it is possible to calculate a color histogram for each body segment. We use the HSL color space to get more brightness invariance. This reference histogram is then compared with a histogram calculated over a small window on the searched normal. In figure 3 the normal is shown and the rectangular window, which is used for histogram and gradient calculation. The expectation is, that the histogram difference changes most rapidly on the point on the normal of the correct correspondence, where the border between person and background is. The type of combination function was chosen by analyzing the developing of gradient and histogram values over 15 normals in different images. The actual values of the combination were then evaluated experimentally trying different values and counting the number of correct correspondences manually for about 100 silhouette points in 4 different images.

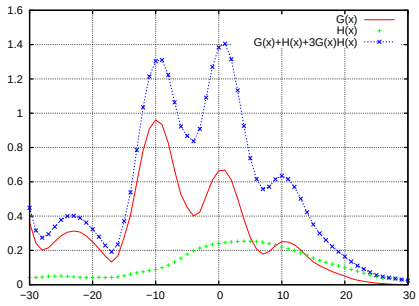

Fig. 4. The gradient (G(x)) and histogram (H(x)) values along the normal. Correct correspondence at 0.

A rather difficult case is shown in figure 4, which shows a plot of the maximum search along the normal of figure 3. The grey value gradient $G(x)$ is shown as a solid line, the gradient of the histogram differences $H(x)$ as points and the combination with lines and points. As visible, the grey value gradient alone would give a wrong correspondence, while the combination yields the correct maximum at zero.

The correspondences found in this way could be integrated into the estimation the same way as the correspondences from feature tracking. However, for most silhouette parts a 2D-3D point correspondence isn't correct, because of the aperture problem. For parallel lines it isn't possible to measure the displacement

in the direction of the lines. Therefore we use a formulation that only minimizes the distance between the tangent at the model silhouette and the target silhouette point, resulting in a 3D-point 2D-line correspondence as visible in figure 5. For a single correspondence the minimization is

$$\min_{\boldsymbol{\theta}} \left[(g(\boldsymbol{\theta}, \boldsymbol{x}) - \boldsymbol{x'})^T \boldsymbol{n} - d\right]^2 \tag{9}$$

where $\boldsymbol{n}$ is the normal vector on the tangent line and d is the distance between both silhouettes, which can be computed as $d = (\hat{\boldsymbol{x}}' - \boldsymbol{x}')\boldsymbol{n}$. The point on the image silhouette $\hat{\boldsymbol{x}}'$ is the closest point to $\boldsymbol{x}'$ in direction of the normal. In this formulation a movement of the point perpendicular to the normal will not change the error. We calculate the normal vector as the projected face normal of the triangle, which belongs to the point $\boldsymbol{x}'$.

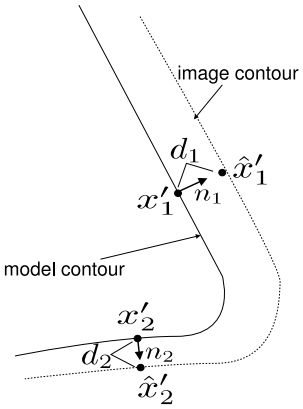

Fig. 5. Silhouette correspondences

For a set $\boldsymbol{X}$ with N points and projected image points $\boldsymbol{X'}$ the optimal solution is:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \left[(g(\boldsymbol{\theta}, \boldsymbol{x_i}) - \boldsymbol{x'_i})^T \boldsymbol{n_i} - d_i\right]^2$$

This is again a *Nonlinear Least Squares* problem and can be solved as above with the following Jacobian:

$$J_{ik} = (\frac{\partial g(\boldsymbol{\theta}, \boldsymbol{x_i})}{\partial \theta_k})^T \boldsymbol{n_i}$$

Note that each of these correspondences gives one row in the Jacobian.

Equation (9) is an implicit description of a 2D line. The same formulation is used in [3], where the normal of the line is the image gradient and the difference d is the grey value difference. The equations for the articulated object are derived using twists, but lead to the same equations and are also solved with the *Gauss-Newton* method. However in [3] the perspective projection was approximated by a scaled orthography.

5 Results

Correspondences from point tracking and from silhouette difference are combined within the optimization by joining both correspondence sets together. Because we estimate pose with different correspondences, weights are added in the Least Squares steps. That way it is possible to ensure a similar influence of 3D-2D point correspondences and 3D-point 2D-line correspondences.

In the following sequences 19 DOF were estimated. Five for the global position and rotation (rotation back and forth is not estimated), one for abduction of the whole shoulder complex, three for each shoulder, one for the elbow and two for each leg (twisting and abduction). Additionally the estimation was damped by a regularization term, such that a large change of joint angles in one iteration is unlikely, if only a few correspondences affect the joint. This way no correspondences for a segment lead to no change for that joint.

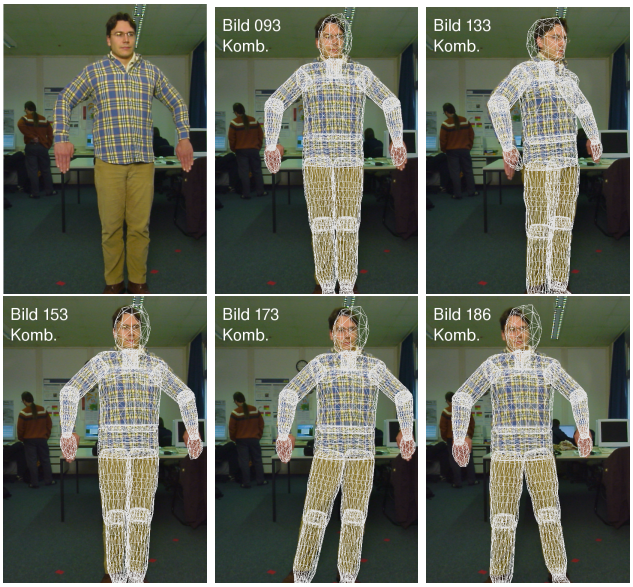

Fig. 6. Original image and estimated model pose with 19 DOF.

Figure 6 shows results for a simple movement, that consists of a rotation of the upper body and stepping aside afterwards. The person is wearing a checkered shirt that exhibits lots of disturbing gray value gradients. The estimated body pose is shown in white as superimposed on the real camera image. As visible, the movement could be captured successfully from a single camera view in spite of the unknown cluttered background and the inhomogeneous clothing.

Results for a more complex movement for a different person are shown in figure 7. The person is wearing a T-shirt and the background is non-static and cluttered again. Movement between frames is quite large, because capturing was done with 7 fps, while the person was moving at normal speed. Even though the shoulder and the upper arm are completely hidden during some frames, the movement could be captured correctly.

6 Conclusions

We showed how estimation of human movement can be derived from point transformations of an articulated object. Our novel approach uses a full perspective camera model and minimizes errors where they are observed, i.e. in the image plane. That way we overcome limitations and approximations of previous work. No explicit segmentation of the images is needed. Correct correspondences are found in spite of cluttered non-static background and normal clothing. Motion with 19 DOF could be estimated that even contained partially hidden body parts. Movements parallel to the optical axis of the camera are not possible to estimate accurately from a single view, e.g. movement of the arms back and forth.

The estimation method is fast enough to fulfill real-time conditions in the near future as processing of one frame is done in less than a second. Ongoing work is to combine this approach with body pose estimation from depth images.

References

1. H. Araujo, R. Carceroni, and C. Brown. A Fully Projective Formulation to Improve the Accuracy of Lowe's Pose Estimation Algorithm. *Comp. Vis. and Image Understanding*, 70(2), 1998.
2. M. Bray, E. Koller-Meier, P. Mueller, L. Van Gool, and N. N. Schraudolph. 3D Hand Tracking by Rapid Stochastic Gradient Descent Using a Skinning Model. In *CVMP*. IEE, March 2004.
3. C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceeding IEEE CVPR*, pages 8–15, 1998.
4. Edwin K.P. Chong and Stanislaw H. Zak. *An Introduction to Optimization, Second Edition*, chapter 9. Wiley, 2001.
5. Lars Mündermann et al. Validation of a markerless motion capture system for the calculation of lower extremity kinematics. In *Proc. American Society of Biomechanics*, Cleveland, USA, 2005.
6. J.-F. Evers-Senne, J.-M. Frahm, D. Grest, K. Köser, B. Streckel, J. Woetzel, and J.-F. Woelk. Basic Image AlgorithmS (BIAS) open-source-library, C++. www.mip.informatik.uni-kiel.de/Software/software.html, 2005.
7. D. Grest, D. Herzog, and R. Koch. Human Model Fitting from Monocular Posture Images. In *Proc. of VMV*, Nov. 2005.
8. D. Grest, J. Woetzel, and R. Koch. Nonlinear Body Pose Estimation from Depth Images. In *Proc. of DAGM*, Vienna, Sept. 2005.
9. I. Kakadiaris and D. Metaxas. Model-Based Estimation of 3D Human Motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 22(12), 2000.
10. R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
11. M. Niskanen, E. Boyer, and R. Horaud. Articulated motion capture from 3-D points and normals. In *CVMP*, London, 2005.
12. Ralf Plaenkers and Pascal Fua. Model-Based Silhouette Extraction for Accurate People Tracking. In *Proc. of ECCV*, pages 325–339. Springer-Verlag, 2002.
13. B. Rosenhahn, U. Kersting, D. Smith, J. Gurney, T. Brox, and R. Klette. A System for Marker-Less Human Motion Estimation . In W. Kropatsch, editor, *DAGM*, Wien, Austria, Sept. 2005.
14. C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, Pittsburg, PA, 1991.
15. B. Wettegren, L.B. Christensen, B. Rosenhahn, O. Gran ert, and N. Krüger. Image uncertainty and pose estimation in 3d euclidian space. *Proceedings DSAGM*, pages 76–84, 2005.

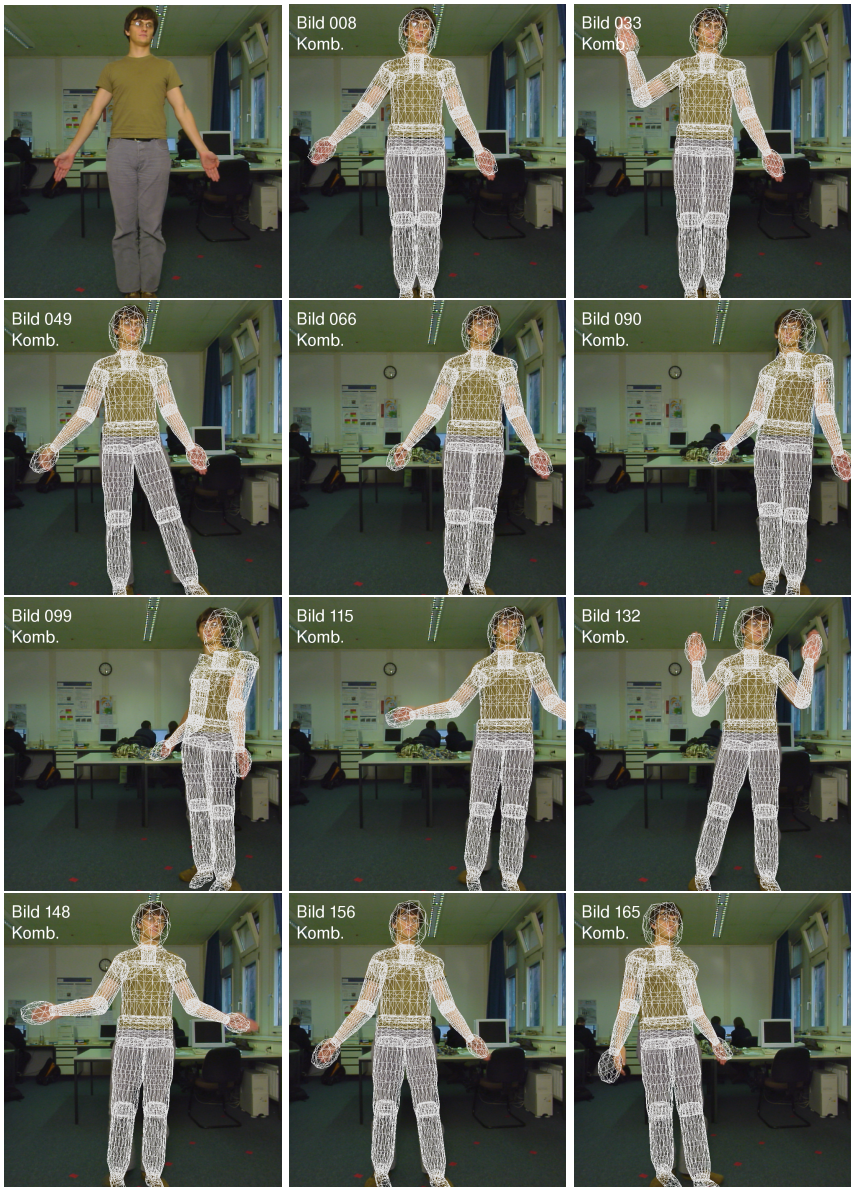

Fig. 7. Original image and estimated sequence with 19 DOF.