Human Model Fitting from Monocular Posture Images

Daniel Grest, Dennis Herzog and Reinhard Koch

Institute of Computer Science
Christian-Albrechts-University Kiel, Germany
Email: {grest,dherzog,rk}@mip.informatik.uni-kiel.de

Abstract

Marker-less motion capture systems usually rely on a 3D skin and skeleton model of the observed person. We present a system that is able to fit a template MPEG4 body model to a person from multiple views taken with a single camera. The person is observed in 6 different postures. Based on contour differences between model and person, a global nonlinear optimization method estimates the scale values of each body segment. Qualitative and quantitative results for different persons show, that a good fitting can be achieved in spite of the simple setup.

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 marker-less motion capture applicable. There are very different approaches using very different kind of information, e.g. tracking features, contour information, color tracking, or depth data. An overview of marker-less motion capture systems and algorithms is given in [9].

Most motion capture systems and algorithms rely on 3D skin and skeleton models of the person being captured. Especially marker-less systems are in the need for an at least rough model of the observed person.

We present a system, which is able to construct a 3D skin and skeleton model of a person by analyzing images from a single camera. Assuming a known static background, the person is captured in 6 roughly predefined poses. A template model is fitted to the person's contour in the 6 images, by scaling each body segment differently in size. The Levenberg-Marquardt (LM) method is used to find the scale values that minimize the difference between the model's projected contour and the person's segmented contour. The fitting for one pose is visible in figure (1). On the left the segmented contour is shown in grey, the middle image shows the template model and on the right the fitting result. Additional 3D pose parameters like joint angles and the transformation between model and camera are optimized simultaneously, which allow a good fitting even if the person did not strike the pose adequately.

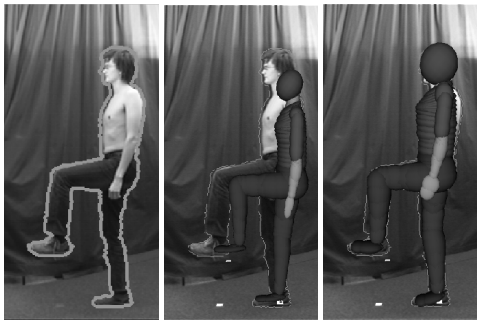

Figure 1: The grey contour in the left image is used to fit the template (middle image). Fitted model on the right.

The main advantage of our method is the simple setup: Only a single calibrated low-cost camera is needed and standard room lighting is sufficient, because the segmentation is invariant to casted shadows and the fitting algorithm is very robust to small errors in the segmented contours.

2. Related Work

There are two main approaches to model fitting of human bodies. The first are shape-constructing methods like 3D-shape-by-silhouette, laser range scanning (as used for obtaining the digital 3D hull of handmade models) or shape from dense disparity maps etc. These methods obtain first a surface

skin model of the person and estimate or fit the underlying movement capabilities (the skeleton) in a second step to the estimated skin model. These approaches rely on perfect segmentation of the person and background and are sensitive to noisy measurements, but do not need a template model and are therefore well suited to fit anomalies like unusual clothes etc. An example for a shape-by-silhouette approach is [13], where an individual is captured by 16 calibrated cameras in a box with homogenous background. In [8] a real-time silhouette approach using 4 cameras is able to track roughly the motion of a person and recognizes different postures.

The second approach is to start from a template model, usually a combination of skin and skeleton, and to adapt the template to the currently observed person by minimizing the difference of some calculated features between model and images. Contour or silhouette information is usually among them. A good example is the work of Pascal Fua, e.g. [10], where a body model consisting of scalable spheres is fitted to silhouette and depth information. Another recent approach, which gives very good fitting results, but relies on manual interaction is given in [11]. In that work a template skin and skeleton model is fitted to 6 synchronized camera views, where only the internal camera parameters are known.

Our method only needs a single camera and doesn't need a homogenous background, because the method is robust against small errors in the segmented contour. We achieve this by our global optimization approach, that simultaneously optimizes the pose and the scale values in all images.

The paper is organized as follows. We begin with a description of the template model. The next section describes our segmentation approach, which leads to contour information. The segmented contour is then used to minimize the distance between model and person contour by a global optimization method. The estimated parameters and the error function are explained in the following sections. Qualitative and quantitative fitting results are given in the results section. At last the fitted body model is used to capture the motion of a person and results are given.

3. Body Model

Depending on the kind of work different body models are used for the estimation process. The models range from simple stick figures [2] over models consisting of scalable spheres (meta-balls) [10] to linear blend skinned models [1]. We use models consisting of rigid, scalable meshes for each body segment, that try to make a balance between fast computation, which requires low resolution models with few points and accurate modeling of the person. The movement capabilities are the same as defined in the MPEG4 standard. An example for the movement capabilities of the arm are shown in figure (2). Depending on the complexity of the model some degrees of freedom are missing. One of the models used in the experiments of this work has for example a rigid upper body, as visible in figure (2). Another model fitted here, which has the full movement capabilities of the MPEG4-body (figure 4), has in contrast at each vertebra of the spine up to 3 degrees of freedom, with limited or unlimited rotation around each axis. The use of the MPEG4 definition of movement gives the opportunity to exchange the body model easily and to have the fitted model in a common format.

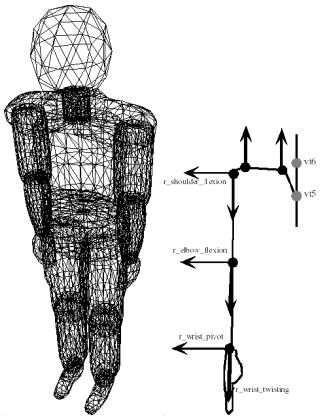

Figure 2: The body model (left) and the joints of the arm (right)

4. Segmentation

The fitting process relies on the contour difference between observed person and projected model. To obtain the contour of the observed person a segmentation of the image fore- and background is necessary. We assume here that the background is static and known, which is achieved by taking images from the background without the person.

The segmentation should have the following properties:

- The image noise should be taken into account
- Casted shadows of the person should be segmented as background
- The background may be any static scene

Figure 3: Segmentation result

The segmentation itself is basically done by thresholding a difference image. For each pixel the variance and mean is calculated on a sequence of the static scene without the person to reflect the image noise. This modeling of person and background regions is very similar to that of [7].

To treat shadows, the original image RGB values are converted into the HSL (hue, saturation, lightness) space, which de-couples color and brightness and represents the color as hue and saturation. Ideally shadows don't change the color of the covered scene part, but only their brightness, therefore a representation in the HSL space is well suited to make the segmentation invariant to shadows. It was seen from experiments that these invariance is at least achieved to some degree. Problematic image regions are pixels of the person, that have similar color with the background pixels at the same image position, because segmentation of those pixels has to be done by their brightness difference. In case that the person can be well distinguished from the background by color, because there are only highly saturated colors present, the segmentation is indeed invariant to shadows. Correct segmentation of shadows could be improved by a similarity measure like in [4], but was not necessary in the setup used here.

Features like color and brightness aren't sufficient for a background of arbitrary appearance. However the use of color and brightness allows for more varying backgrounds than a simple grey value thresholding as visible in figure (6).

To decide whether a pixel belongs to the background or to the person, the Mahalanobis distance between HSL-pixel value and HSL-mean is thresholded. Therefore it is necessary to convert the polar coordinates of the HS channels into Cartesian coordinates (written here as hsL).

Let Σ be the covariance matrix at an image position and $\boldsymbol{\mu} = (\bar{h}, \bar{s}, \bar{L})$ the corresponding mean value. Then a pixel with values (h, s, L) belongs to the person if

$$\sqrt{\begin{pmatrix} h-\bar{h} \\ s-\bar{s} \\ \gamma(L-\bar{L}) \end{pmatrix}^{\top} \Sigma^{-1} \begin{pmatrix} h-\bar{h} \\ s-\bar{s} \\ \gamma(L-\bar{L}) \end{pmatrix}} \geq t.$$

The manually given threshold t and the parameter γ are the same for all pixels. The scale value γ gives the possibility to vary the importance of the brightness difference. In case of fully saturated colors present in the background, this value may be set to zero, and is usually in $[0, 1]$. To handle cases, where no variance in color is present at a certain pixel position over the image sequence of the background, e.g. if a pixel value is dense or zero in all images, we add the identity to the covariance matrix.

A typically segmentation result is visible in figure (3) left. To eliminate noise we apply an opening operator and take the largest connected region afterwards. Small holes are filled by a closening operator, which results in the final segmentation as in figure (3) right.

5. Distance Transform

The contour of the segmented person is found by a contour following algorithm. The extracted image contour is then distance transformed.

The distance transform is an operator usually applied to binary images. The result of the transform is an image that has at each image position the distance to the contour as the pixel value. There

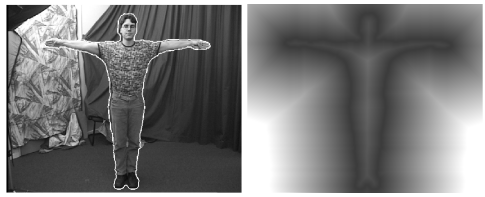

Figure 5: Distance transformed image of the person contour.

are several different sorts of distance transforms depending upon which distance metric is being used to determine the distance between pixels. We use here the *Borgefors* metric, which assigns a horizontal displacement of 1 pixel the value 3 and a diagonal the value 4. This way integer operations can be performed. We implemented the two-pass algorithm as described in [12], which is faster than a convolution approach.

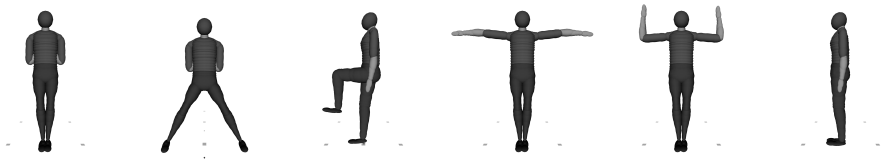

Figure 4: The 6 poses used for optimization

An example of a distance transformed image of the segmented person contour is shown in figure (5). The contour of the segmented person is shown in white in the left image.

6. Optimization Function

The resulting model parameters after the fitting are the scale values of each body segment, one for each direction of the local coordinate system. Let $\boldsymbol{\sigma}_i = (\sigma_i^x, \sigma_i^y, \sigma_i^z)$ be the scale values for the i-th body segment. The optimization problem can then be defined as follows: Find the scale values $\boldsymbol{\sigma}_1, .., \boldsymbol{\sigma}_N$ of the N body segments, such that the model contour looks most similar to the segmented contour of the person.

6.1. Error Function

The error function, which models these similarity, consists of three different parts:

$$f(\boldsymbol{p}) = \boldsymbol{e} = (\boldsymbol{e}^P, \boldsymbol{e}^M, \boldsymbol{e}^B)$$

The three error vectors are now described in more detail.

6.1.1. Person to Model Distance

The contour C^M of the model is calculated with respect to the current scale parameters $\boldsymbol{\sigma}_1, .., \boldsymbol{\sigma}_N$. Afterwards for each pixel on the person's contour C^P, which does not change during the optimization, the corresponding value in the distance transformed image D^M of the model contour is taken as the error for that pixel. Afterwards each error value is normalized by the length of the person's contour. This error $\boldsymbol{e}^P \in \mathbb{R}^{|C^P|}$ has for each pixel on the person's segmented contour a value, which is the distance to the nearest pixel on the model's contour. Each entry of $\boldsymbol{e}^P$ is defined as

$$e_i^P = \frac{1}{\sqrt{|C^P|}} D^M(\boldsymbol{x_i}),$$

where $\boldsymbol{x_i}, i = 1..|C^P|$ is the i-th contour point.

6.1.2. Model to Person Distance

Ideally the opposite distances should be also taken into account, which is the distance from each pixel on the model contour to the nearest one on the person's contour. Because the model contour changes during the optimization, only the average error of all contour pixels is taken, such that

$$e^M \in \mathbb{R} = \frac{1}{\sqrt{|C^M|}} \sqrt{\sum_{\boldsymbol{x} \in C^M} (D^P(\boldsymbol{x}))^2}.$$

6.1.3. Constraints

To ensure that the estimated parameters are in a valid range, barrier functions are used. For each parameter an additional error is calculated that represents the distance to the desired interval. E.g. each scale value has to be in $[s_{\min}, s_{\max}]$. In the experiments we set $s_{\min} = 0.4$ and $s_{\max} = 1.9$. The actual choice of the barrier function is not important, but is usually like that in figure (7). This part of the error vector is $\boldsymbol{e}^B \in \mathbb{R}^{3N}$.

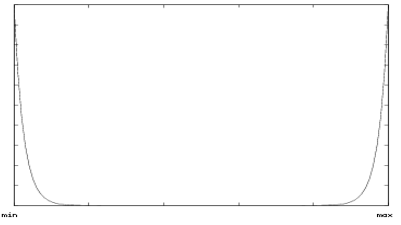

Figure 7: The barrier function

6.2. Justification

We now justify briefly, why the above composition of $\boldsymbol{e}$ is appropriate for our minimization problem. In the optimization process $|\boldsymbol{e}|$ gets minimized or

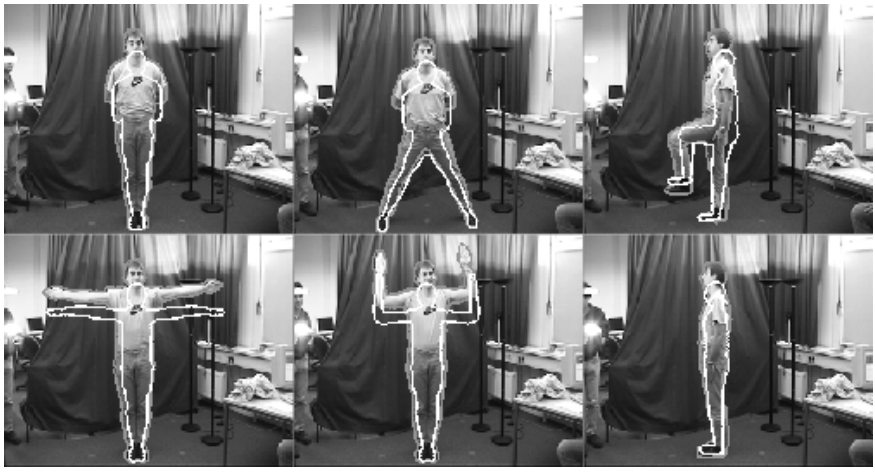

Figure 6: Optimization starting configuration. Initial model contour in white.

rather

$$|\boldsymbol{e}|^2 = |\boldsymbol{e}^P|^2 + |e^M|^2 + |\boldsymbol{e}^B|^2 \quad (1)$$

$$= \frac{\sum_{\boldsymbol{x}\in C^P} D^M(\boldsymbol{x})^2}{|C^P|} + \frac{\sum_{\boldsymbol{x}\in C^M} D^P(\boldsymbol{x})^2}{|C^M|} + |\boldsymbol{e}^B|^2 \quad (2)$$

Equation (2) shows that in $|\boldsymbol{e}|^2$ the person to model contour distance and the model to person contour distance have the same influence on the optimization in spite of the higher dimensionality of $\boldsymbol{e}^P$.

The LM method is only appropriate for overdetermined problems. A good description of the LM method is given in the Appendix of [6]. The error function $\boldsymbol{e} = F(\boldsymbol{p})$ is of much higher dimensionality than the parameter vector $\boldsymbol{p}$. However this does not guarantee an overdetermination. Only if the change of each parameter does at least change one error value it is possible to reach the desired solution. Or to be more exact, only if the Jacobian of F has full rank. If only a single view is taken to estimate all scale values, there are some scale values that have no effect on the value of the error function.

7. Multiple Views

So far we explained how the optimization can take place for a single image of the person using the person's segmented contour and the rendered model with its contour. The optimization minimizes the differences of appearance for both contours. However it is not possible to estimate all scale values for each body segment from a single view. Therefore we extend the optimization to multiple views.

For each view the person has to position itself in a special configuration, called a pose in the latter. We use 6 different poses as shown in figure (4). The different poses allow estimation of specific scale parameters. For example the overall height is covered in the first pose. The length of the legs is specified in the second pose and the third pose gives the relation between lower and upper leg etc. Important is, that the optimization is not done for single views separately, but all parameters are estimated for all poses simultaneously.

Multiple views increase the number of error values by $\boldsymbol{e}^P$ for each view. Additionally there is one error value e^M calculated on the model's contour.

Someone can imagine that it is rather difficult for a person to perfectly strike all the poses. For example it was seen to be very difficult to stand on one foot and bend the knee in exactly 90 degrees.

If the captured pose of the person is not exactly as desired a simple scaling of all body segments, can never match the contour perfectly, like the bended knee in image 3 (upper right) of figure (6). Therefore we estimate for each pose additional parameters, namely the joint values, which were seen to vary significantly. These were for example the angle of the knee in image 3, or the angle of the elbow flexion and the shoulder abduct as visible in the middle image of the lower row of figure (6).

In each pose the person has to stand at the same marked position on the floor. Usually people tend to

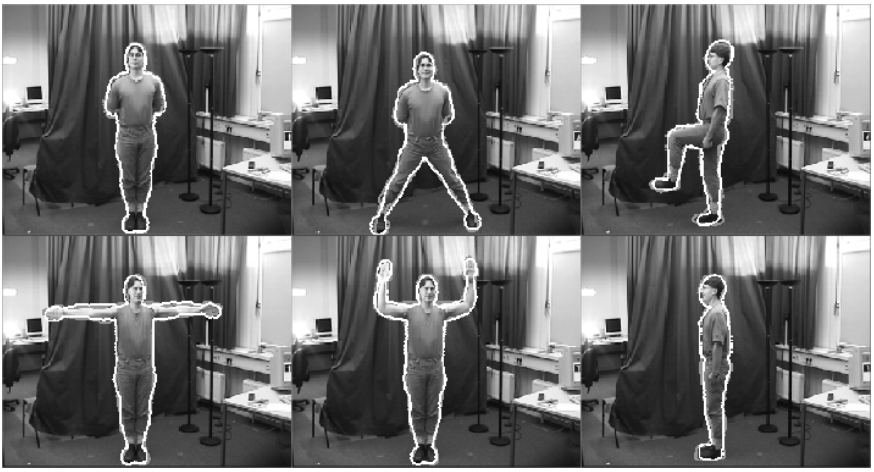

Figure 8: Fitting result. Model contour in white. Color version in the Appendix.

move a little aside from that position, therefore the translation on the ground is also optimized within a constrained interval. Because it is not possible to stand quite straight without moving for a untrained person, the global rotation is also estimated for each pose separately. The overall optimization parameters are

$$\boldsymbol{p} = (\boldsymbol{\sigma_1}, .., \boldsymbol{\sigma}_N, \boldsymbol{v_1}, ..\boldsymbol{v_L})$$

where $\boldsymbol{v_j}$ is the vector with parameters for the j-th pose and $\boldsymbol{\sigma_i}$ are the desired scaling parameters. For each pose $\boldsymbol{v_j}$ consists of parameters for some important joints and values for the global translation and rotation.

8. Fitting Algorithm

The captured images of the person striking the 6 poses are used offline to fit the template body model. The difference between model and person contour is minimized. As the correspondences between both are given only implicitly due to the nearest neighbor approach, the fitting does only reach the global optimum, if the starting point is near enough to it. Problematic are especially the outstretched arms of the 4th pose, see lower left of figure (6). If the person is significantly smaller or larger than the template model, the nearest neighbor approach will fit both the upper and lower contour part of the person's arm to only one contour part of the model. Therefore it is necessary to do the optimization in two steps. In the first step only the length of the legs, and the upper body are estimated. In the second step all parameters for all poses are estimated simultaneously. In both steps we use the LM algorithm from the MINPACK package [3], which calculates the Jacobian numerically.

Additional symmetry constraints guarantee that the left and right arms and legs are of the same size. We will not give here the exact set of estimated parameters, because these are up to 100 depending on the used model. For high complex models, which have the full MPEG4-movement capability, some body segments are coupled, e.g. for the spine segments of the upper body only three scale values are estimated.

9. Fitting Results

The fitting was tested for 7 persons, which had to strike all the 6 poses. To make the task easier for the people, the current camera image augmented with the desired model pose was displayed in front of them, such that they could directly see how good they strike the pose. The model had the full MPEG4 movement capabilities as shown in figure (4). A qualitative result for the fitting is visible in figure (8), where the template was fitted to a person, which was rather different to the template model. The overall fitting process takes about 5 min for acquiring the images as the person has to strike 6 different poses and about 10 minutes for the optimization depending on the hardware.

To measure the accuracy of the model, the overall height, the arm length and the length of the lower leg of the models were compared to manual measurements of the real persons. The results are shown

	Height			Arm length			Length lower leg			mean
Pers.	Pers.	Mod.	Err.	Pers.	Mod.	Err.	Pers.	Mod.	Err.	Err.
0	1.95	1.98	1.5%	1.90	1.86	2.1%	0.62	0.66	5.7%	3.1%
1	1.78	1.84	3.4%	1.75	1.68	3.7%	0.58	0.62	7.3%	4.8%
2	1.68	1.69	0.8%	1.64	1.63	0.9%	0.51	0.54	7.7%	3.1%
3	1.81	1.86	2.8%	1.83	1.84	0.6%	0.59	0.65	9.5%	4.3%
4	1.92	1.97	2.7%	1.96	1.95	0.6%	0.62	0.69	11.3%	4.8%
5	1.88	1.92	1.9%	1.81	1.83	0.9%	0.61	0.66	8.5%	3.8%
6	1.89	1.86	1.6%	1.95	1.91	2.2%	0.63	0.65	3.6%	2.5%
mean.			2.1%			1.6%			7.6%	3.8%

Table 1: Quantitative fitting results for the 7 persons

in table (1).

The achieved average error of 2.1% for the height and 1.6% for the arm length is within the expected range and accurate enough for further use of the model. The error of 7.6% for the lower leg is higher, because on the one hand measuring the ground truth was more difficult and inaccurate. On the other hand the image information for the estimation of the lower leg scale values is much less than for the height and arm length. Therefore small segmentation errors have a larger effect on the estimation. To decrease the error further the camera calibration could be more accurate and higher resolution images could be taken. The image used here for the experiments were PAL resolution.

10. Motion Capture Results

The same images of the person in the 6 poses can be applied to different body models. To capture the motion of a person we took a simpler model as visible in figure (2) and (9), which consists of 18 rigid body parts and up to 40 degrees of freedom (DOF). We present here results as calculated by the method described in [5], where depth images from a moving person are used to estimate joint angles. The person moved its arms at first in a waving manner and later crossing the arms in front of the chest. In figure (9) three images from the sequence are shown. The top row shows the depth images with lighter values indicating closer points. The middle row shows the original images overlayed with the estimated model pose in grey. The bottom row shows the same model pose as seen from another position. The model's position is estimated below the real person throughout the sequence, because the model was fitted offline beforehand to the person showing a bare upper body. For this sequence 14 DOF were estimated: The global transform, the three shoulder angles and the elbow flexion. The processing of one frame took on the average 200ms on a 3GHz Pentium 4.

11. Conclusions

We presented a system that is able to fit MPEG4 body models to images from 6 different poses of a person. In spite of the simple setup with a single camera and standard room lighting, the fitted models are accurate enough to use them for capturing human motion as presented in the results. The accuracy of the models may be increased by using higher resolution cameras with better calibration and additional images of each pose or additional poses.

References

[1] 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.

[2] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceeding IEEE CVPR*, pages 8–15, 1998.

[3] Wayne R. Cowell. MINPACK: Numerical Library for Function Minimization and Least-Squares Solutions. www.netlib.org/minpack. Chapter 5 of Wayne R. Cowell:Sources and Development of Mathematical Software,1980, Prentice-Hall Series in Computational Mathematics, Cleve Moler, Advisor.

[4] Daniel Grest, Jan-Michael Frahm, and Reinhard Koch. A Color Similarity Measure for Robust Shadow Removal in Real Time. In *Proc. of Vision, Modeling and Visualization (VMV)*, pages 253–260, Munich, Germany, Nov. 2003.

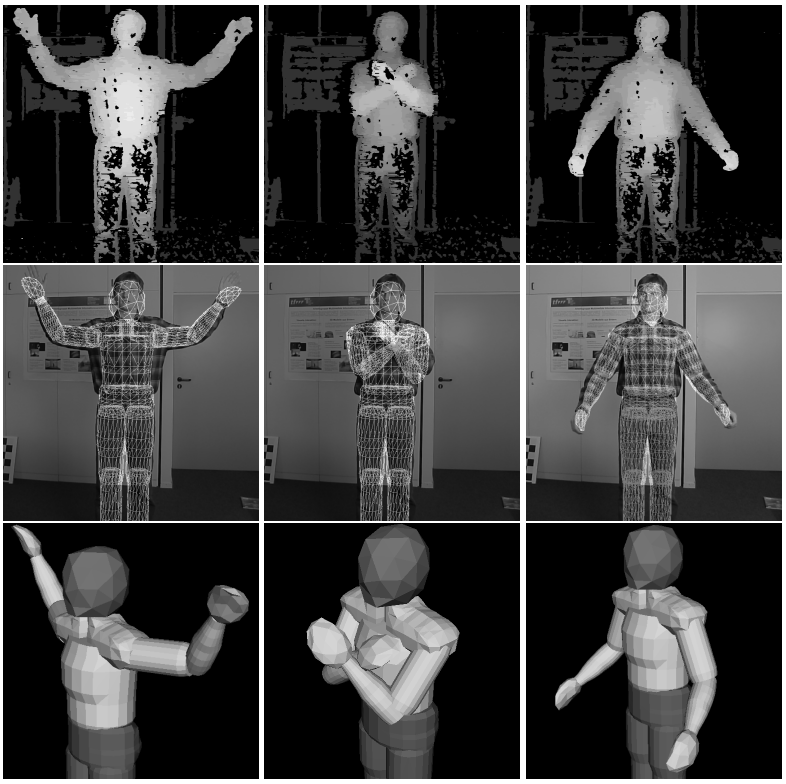

Figure 9: Top Row:Depth images, middle row: original image overlayed with estimated model pose, bottom row: model view from the side. Color version in the Appendix.

[5] Daniel Grest, Jan Wotzel, and Reinhard Koch. Nonlinear Body Pose Estimation from Depth Images (to appear). In *Proc. of DAGM*, Vienna, Sept. 2005.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, 2000.

[7] T. Horprasert, I. Haritaoglu, and D. Harwood. Real-time 3D motion capture. In *Proc. of Perceptual User Interfaces*, pages 87–90, Nov. 1998.

[8] Jason Luck, Dan Small, and Charles Q. Little. Real-Time Tracking of Articulated Human Models Using a 3D Shape-from-Silhouette Method. In *RobVis '01: Proceedings of the International Workshop on Robot Vision*, pages 19–26, London, UK, 2001. Springer-Verlag.

[9] T. B. Moeslund and E. Granum. Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding: CVIU*, 81(3):231–268, 2001.

[10] Ralf Plaenkers and Pascal Fua. Model-Based Silhouette Extraction for Accurate People Tracking. In *Proc. of ECCV*, pages 325–339. Springer-Verlag, 2002.

[11] Chee Kwang Quah, Andre Gagalowicz, Richard Roussel, and Hock Soon Sah. 3D Modeling of Humans with Skeletons from Uncalibrated Wide Baseline Views (to appear). In *Proc. of CAIP*, Paris, 2005.

[12] Pierre Soille. *Morphological Image Analysis, Principles and Applications, 2nd Edition*, pages 47–48. Springer Verlag, 2002.

[13] Jochen Wingbermühle, Claus-E. Liedtke, and Juri Solodenko. Automated Acquisition of Lifelike 3D Human Models from Multiple Posture Data. In *Proc. of CAIP*, pages 400–409, 2001.