Automatic Human Model Generation

Bodo Rosenhahn[*], Lei He, and Reinhard Klette

University of Auckland (CITR), Computer Science Department,
Private Bag 92019 Auckland, New Zealand
rosenhahn@mpi-sb.mpg.de

Abstract. The contribution presents an integrated system for automatic acquisition of a human torso model, using different input images. The output model consists of two free-form surface patches (with texture maps) for the torso and the arms. Also, the positions for the neck joint on the torso, and six joint positions on the arms (for the wrist, elbow and shoulder) are determined automatically. We present reconstruction results, and, as application, a simple tracking system for arm movements.

1 Introduction

Human motion modeling plays an increasingly important role in medical applications, surveillance systems or avatar animation for movies and computer games. This work is part of a human motion analysis project as presented in [16]. For a detailed study of human motions the research project requires an automatic model generation system, so that the pose recovery can be evaluated for different persons (e.g. male, female, small, tall and so forth). Our goal is to present an integrated framework for the automatic generation of human models that consist of free-form surface patches and body segments connected by joints. The input of the algorithm is a set of 4 images and the output is a VRML-model of the torso. The basic structure is given in Figure 1.

In the literature 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 being a low-cost technique. Kakadiaris et al. propose in [7] 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. [4] propose an approach for modeling 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 find elbow or wrist joints. Plänkers et al. [13] model an articulated body by using layers for a skeleton, ellipsoidal meta-balls (to simulate muscles) and a polygonal surface representation (to model the skin). But as discussed in [15] we prefer a non-layered representation, where free-form surface patches are directly assigned to joint indexes. This leads to a

[*] From Nov. 2005: Max Planck Center Saarbrücken.

A. Gagalowicz and W. Philips (Eds.): CAIP 2005, LNCS 3691, pp. 41–48, 2005.

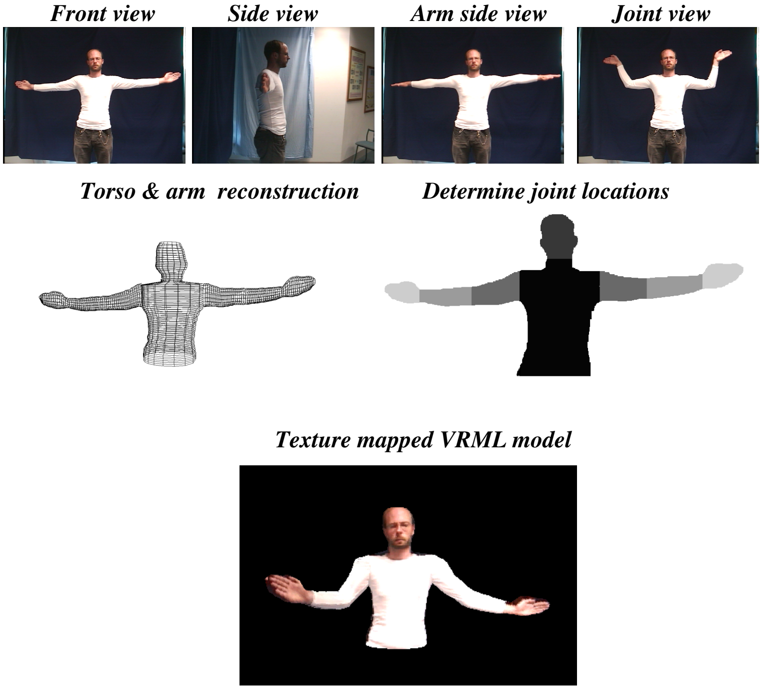
Front view
Side view
Arm side view
Joint view
Torso & arm reconstruction
Determine joint locations
Texture mapped VRML model

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

more compact representation and allows further freedom for modeling mesh deformations or joint shifts during tracking. Lee et al. [6] build a seamless human model. Their approach obtains robust and efficient results, but it cannot detect joint positions which have to be arranged manually.

The next section presents the implemented modules needed for model reconstruction. Section three presents some reconstruction and tracking results. Section four concludes with a brief discussion.

2 Implemented Modules

This section describes implemented modules and those modifications of existing algorithms which have been necessary to adapt them to our specific tasks.

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 static background. Here we use a modified version of [5], which proves to be fast and stable: To decide 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 decomposed in two components, brightness and chromaticity. Thresholds are used to segment the images as shown in Figure 1. Afterwards the images are smoothed using morphological operators [9].

Body Separation
Firstly it is necessary to separate the arms from the torso of the model. Since we only reconstruct 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 walking 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 positions of 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 skeletonization procedure. Skeletonization is a process of reducing object pixels in a binary image to a skeletal remnant that largely preserves the extend and connectivity of the original region while eliminating most of the original object pixels. Two skeletonization approaches are common, those based on thinning and those based on distance transforms. We implemented the thinning approach presented in [8] called iterative thinning algorithm. The left image of Figure 2 shows that the algorithm leads to a connected skeleton, but unfortunately it is not centered. Furthermore, we are interested in detecting corners of the skeleton, but the resulting curve is very smooth which makes it hard to detect, for example, the position of the elbow joint. The middle image of Figure 2 shows the result using the Chamfer distance transform. Here the skeleton is centered, but unfortunately it is not connected. We decided to work with the skeletons based on the Chamfer distance transform and close the skeleton by connecting nearest non-neighboring points. This leads to centered skeletons as shown on the arms in right of Figure 2. We further use the method presented in [2] to detect corners on the skeleton to identify joint positions of the arms.

Furthermore, we would like to 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 such that it corresponds to the human anatomy, see

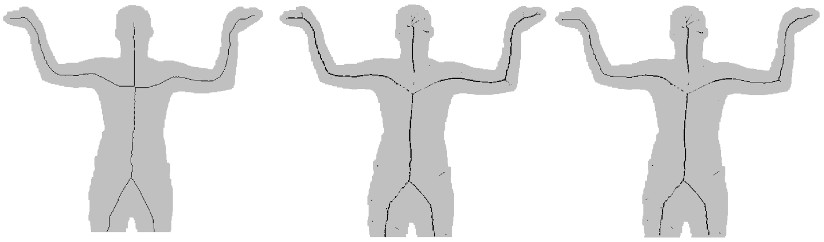

Fig. 2. Skeletonization using iterative thinning (left), Chamfer distance transform (middle), and our modified version (right)

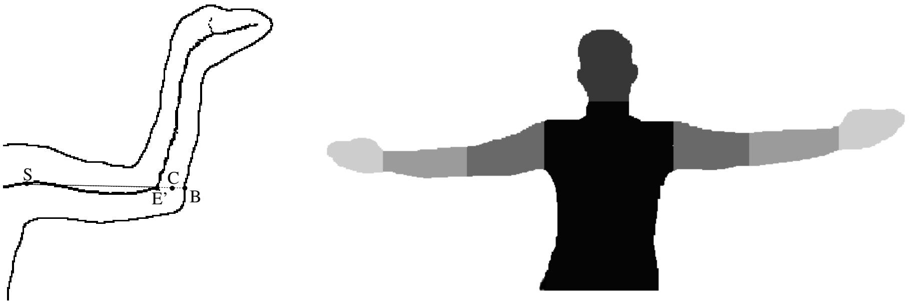

Fig. 3. Left: Adapting elbow joints. Right: Extracted joint segments.

in the left image of 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 result of joint locations is shown at the right image of Figure 3.

Surface Mesh Reconstruction

For surface mesh reconstruction we assume calibrated cameras in nearly orthogonal views. Then a shape-from-silhouettes approach [10] is applied. We attempt to find control points for each slice, and then to interpolate them as a B-spline curve using the DeBoor algorithm. For this we start with one slice of the first image and use its edge points as the first two reference points. Then they are multiplied with the fundamental matrix of the first to the second camera and the resulting epipolar lines are intersected with the second silhouette resulting in two more reference points. The reference points are intersected leading to four control points in 3D space.

For arm reconstruction 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 such so that they are horizontally and have the same fingertip starting point. This is shown in Figure 4. 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.

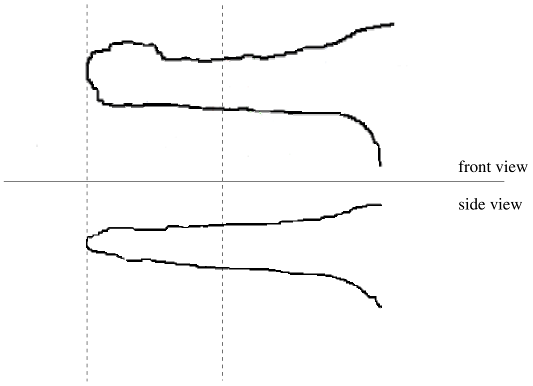

Fig. 4. Alignment of the arm

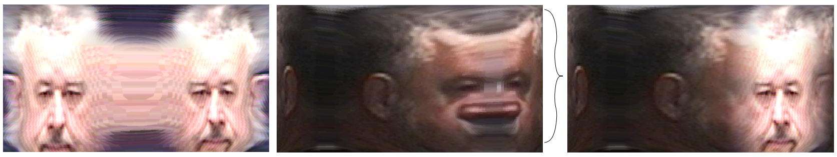

Fig. 5. Texture fusion: The images from the stereo setup are merged to get a texture map for the head: the left texture gives a good face, whereas the right texture gives a good ear and side view of the face. The fusion of both textures leads to a new texture used for the 3D model.

For texture mapping, we generate a texture file as a combination of the different views: Here we apply the multi-resolution method proposed by Burt et al. [1] 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.

3 Experiments

We tested the algorithm on four different models. Figure 6 shows in the lower two rows reconstruction results from two persons. One useful application is to animate the models using motion capture data: The top row in Figure 6 shows some capture results

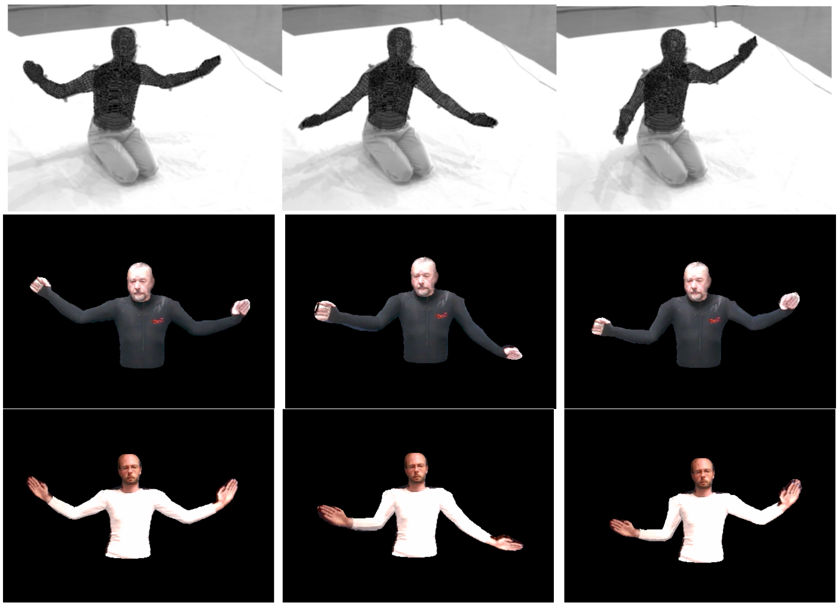

Fig. 6. Pose results of the pose recognition software. Mimicking the arm configurations with the reconstructed models.

Table 1. Example lengths for a test subject (unit: cm)

Name		real person	reconstructed model	error
Bodo	lower arm	25.0	27.0	2.0
	hand	20.0	19.0	1.0
	width	185.2	187.3	2.0

using a human motion estimation algorithm and below are the pose configurations of the reconstructed model. This allows us to let the "Reinhard" model mimic the actors (Bodo) motion.

For a quantitative error analysis we compare reconstructed body parts with (manually) measured ones. One example is shown in Table 1. A comparison with all (four reconstructed) subjects and 6 body parts (head, upper arm, lower arm, width, etc.) shows a maximum deviation of 2 cm. The average error is 0.88 cm.

3.1 Joint Tracking

In a second experiment we apply the model on a simple joint tracking algorithm. Therefore we assume that the reconstructed model is standing in front of a camera and moving its arms. The images are separated in their arm and body components and a skeletonization is applied to detect the arm joints. This approach is similar to the one described in

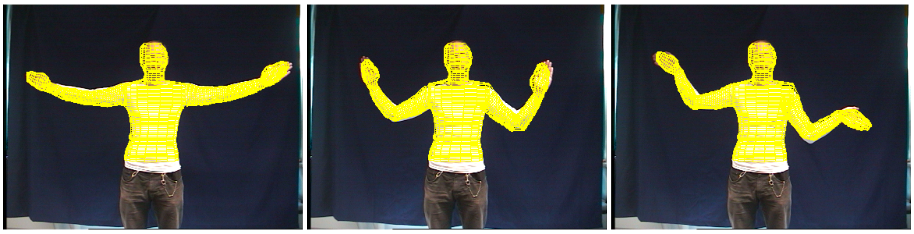

Fig. 7. Joint tracking

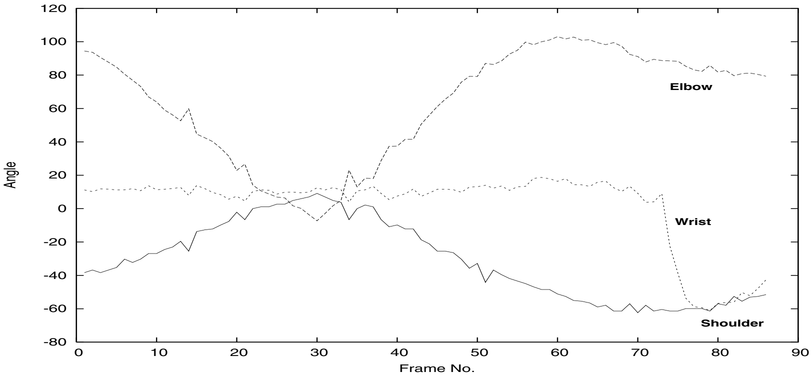

Fig. 8. Joint angles

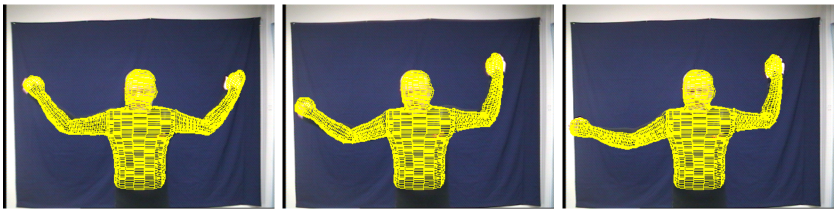

Fig. 9. Arms and head tracking of the "Reinhard"-model

Section 2. If the arm is stretched, the position of the arm is known from the length ratios of the arm components. Tracking results are shown in Figure 7. Some joint angles of the left arm are shown in Figure 8. Though we do not have any ground truth, the angles match with the real motion and the curves are relatively smooth indicating a reasonable stable algorithm.

Figure 9 shows results of the tracked Reinhard model, where also the head angle is estimated. As can be seen, the model fits satisfactory to the image data.

4 Discussion

We present an automatic human model generation system. Several sub tasks are solved and integrated into a system, including a GUI for comfortable interaction of threshold parameters. The algorithm takes a sequence of four images and separates the arms from the torso from a front and a side view. Then a shape-from-silhouettes approach is applied to reconstruct the torso from two views, and the arms from two other views. The joint locations are determined from a fourth image showing a special pose of the arms. Here a skeletonization is applied to detect the joint locations. Finally the model is reconstructed including a texture map from the image data. We apply the reconstruction to a simple joint tracking procedure and show that we are able to track a persons arms with reasonable quality.

The system is developed under a Linux Redhat 9.0 environment. `OpenGL` is used as the graphics API. The application is written in `C/C++`. Human models are specified in the format `VRML`. A scene graph system `OpenSG` helps to parse and visualize the `VRML` file. `GTK` is our GUI development toolkit.

Acknowledgments

This work has been supported by the DFG project RO 2497/1.

References

1. Burt P.J. and Andelson E.H. A multiresolution spline with aplication to image mosaics. *ACM Tarns. on Graphics*, II, No 4, pp. 217-236, 1983.
2. Chetverikov D. A simple and efficient algorithm for detectiopn of high crvature points. *In: Computer Analysis of Images and Patterns*, N. Petkov and M.A. Westenberg (Eds.) Springer-Verlag Berlin, LNCS 2756, pp. 746-753, 2003.

3. Grimson W. E. L. *Object Recognition by Computer.* The MIT Press, Cambridge, Massachusetts, 1990.
4. 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.
5. 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).
6. 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.
7. Kakadiaris I. and Metaxas D. Three-dimensional human body model acquisition from multiple views. *Internation Journal on Computer Vision*, Vol. 30 No. 3, pp. 191-218, 1998.
8. 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.
9. Klette R. and Rosenfeld A. Digital Geometry–Geometric Methods for Digital Picture Analysis *Morgan Kaufmann*,San Francisco, 2004.
10. Klette R., Schlüns K. and Koschan A. Computer Vision. Three-Dimensional Data from Images. *Springer*, Singapore, 1998.
11. Murray R.M., Li Z. and Sastry S.S. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Inc. Boca Raton, FL, USA, 1994.
12. ORourke J. *Computational Geometry in C.* Cambridge University Press, Cambridge, UK, 1998.
13. Plänkers R. and Fua P. Articulated Soft Objects for Multiview Shape and Motion Capture. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9), pp.1182-1187, 2003.
14. Rosenhahn B. Pose Estimation Revisited *Technical Report 0308, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik*, 2003. Available at `http://www.ks.informatik.uni-kiel.de`
15. 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,
16. Rosenhahn B., Klette R. and Sommer G. Silhouette based human motion estimation. *In Proc. Pattern Recognition 2004, 26th DAGM-symposium,* Tübingen, Germany, C.E. Rasmussen, H.H. Bülthoff, M.A. Giese, B. Schölkopf (Eds), Springer-Verlag Berlin, LNCS 3175, pp 294-301, 2004.
17. Shi Y. and Sun H. *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards.* CRC Press, Boca Raton, FL, USA, 1999.