Nonlinear Body Pose Estimation from Depth Images

Daniel Grest, Jan Woetzel, and Reinhard Koch

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

Abstract. This paper focuses on real-time markerless motion capture. The body pose of a person is estimated from depth images using an Iterative Closest Point algorithm. We present a very efficient approach, that estimates up to 28 degrees of freedom from 1000 data points with 4Hz. This is achieved by nonlinear optimization techniques using an analytically derived Jacobian and highly optimized correspondence search.

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. 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 markerless motion capture systems and algorithms is given in [7].

In some applications highly accurate motion data is not needed every frame. In Human-Computer-Interaction applications the focus is usually more on real-time evaluation and robustness with respect to lighting, background etc. Therefore this paper focuses on markerless motion capture that estimates body pose in at least near to real-time.

We estimate human motion from depth data calculated from stereo images. The proposed approach uses an estimation method similar to Rosenhahn [9] and Malik & Bregler [2]. Our approach however uses standard non-linear optimization methods that involves an analytically derived Jacobian of the optimization function. Therefore it can be applied to methods like gradient descent, Quasi-Newton, *Levenberg-Marquardt (LM)* etc. Authors of recent work, which is very similar to our, seem not to be aware of this possibility. Their work might benefit from the derivations and results presented here.

Recent work most similar to this one are, as far as we know, that of D. Demirdjian [4] and M. Bray [1]. Both estimate the body pose from depth data and have an Iterative Closest Point (ICP) approach to find necessary correspondences. Other similar work on body pose estimation using depth data can also be found in [8,6], which is however not fast enough for real-time purposes.

The approach of Demirdjian is fast enough to fulfill real-time requirements as poses are estimated with 10Hz. An ICP estimation is done for each body segment

W. Kropatsch, R. Sablatnig, and A. Hanbury (Eds.): DAGM 2005, LNCS 3663, pp. 285–292, 2005.

separately. Later, the underlying movement constraints (segments are connected by joints) are applied by projecting the evaluated transformations onto the set of possible articulated transformations. As the authors state themselves in [4] '*constraint projection methods may give an sub-optimal solution, but are easier to implement than direct approaches*'. The direct approach was implemented in our work and showed to be fast enough for real-time requirements in spite of its complexity.

In [1] the pose of a human hand is estimated from depth data obtained by a structured light sensor. Movement of points is described as a concatenation of rigid body motions. However the derivative is calculated by numerical methods, which takes a significant amount of time. Different nonlinear optimization methods are applied to the pose estimation problem and a extended gradient descent method, the stochastic meta descent, is proposed to give the best results. Estimation takes 4 seconds per frame from 45 data points.

Our approach uses an analytically derived Jacobian for the optimization, which decreases the computation time significantly. Further optimizations for efficient correspondence calculations enable us to estimate body pose from 1000 data points in 250ms on a Pentium4 3Ghz.

2 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) [8] to linear blend skinned models [1]. We use models consisting of rigid, fixed meshes in 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 (1).

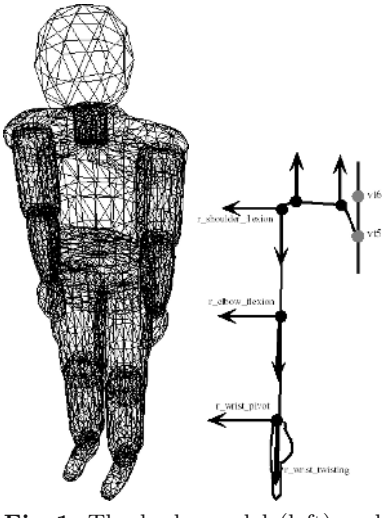

Fig. 1. The body model (left) and the joints of the arm (right)

The body model is fitted to the current observed person offline by scaling each segment separately in size. To fit the template model a contour based monocular algorithm is used. The person is captured in specific predefined poses in front of a known static background and the scale values for each body segment are evaluated by hierarchical nonlinear optimization.

3 Body Pose Estimation

We will show in this section how nonlinear optimization methods can be applied to the problem of body pose estimation. This approach is used within a variety

of applications, e.g. [1,8], but usually relies on numerical derivatives, which are inaccurate and time consuming to compute.

3.1 Rigid Body Motion as Rotation Around Arbitrary Axis

A rigid body motion (RBM) in $\mathbb{R}^3$ may be expressed as a rotation around an arbitrary axis plus a translation along that axis. There are different formulations for rigid body motions, e.g. twists [2], which use a exponential term e^{ψ} or rotors [9], which may be seen as an extension of quaternions. Important here is not a complete RBM, but a rotation around an arbirtrary axis, which is given in standard vector notation of the Euclidean space as follows.

Consider the normal vector $\boldsymbol{\omega}$, which describes the direction of the axis, and the point $\boldsymbol{q}$ on the axis, which has the shortest distance to the origin, i.e. $\boldsymbol{q}$ lies on the axis and $\boldsymbol{q}^T\boldsymbol{\omega} = 0$, refer to figure (2). The rotation of a point $\boldsymbol{x}$ around that axis may then be written as

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{x} + \sin\theta(\boldsymbol{\omega} \times (\boldsymbol{x} - \boldsymbol{q})) + (1 - \cos\theta)(\boldsymbol{q} - \boldsymbol{x}^p) \\ &= \boldsymbol{x} + \sin\theta(\boldsymbol{\omega} \times (\boldsymbol{x} - \boldsymbol{q})) + (\boldsymbol{q} - \boldsymbol{x}^p) - \cos\theta(\boldsymbol{q} - \boldsymbol{x}^p) \\ &\equiv R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta) \circ \boldsymbol{x} \end{aligned} \tag{1}$$

where $\boldsymbol{x}^p = \boldsymbol{x} - (\boldsymbol{x}^T\boldsymbol{\omega})\boldsymbol{\omega}$ is the projection of $\boldsymbol{x}$ onto the plane through the origin with normal $\boldsymbol{\omega}$. Note that $\boldsymbol{q}$ is also on that plane. This expression is very useful as the derivative $R'_{\boldsymbol{\omega},\boldsymbol{q}}(\theta) = \frac{\partial R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta)}{\partial\theta}$ is easy to calculate:

$$R'_{\boldsymbol{\omega},\boldsymbol{q}}(\theta) = \cos\theta(\boldsymbol{\omega} \times (\boldsymbol{x} - \boldsymbol{q})) + \sin\theta(\boldsymbol{q} - \boldsymbol{x}^p) \tag{2}$$

3.2 Concatenation of Rotations

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. As the rotation axis are known, e.g. the flexion of the elbow, the rotation has only one 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 $p - 6$ joints the transformation may be written as:

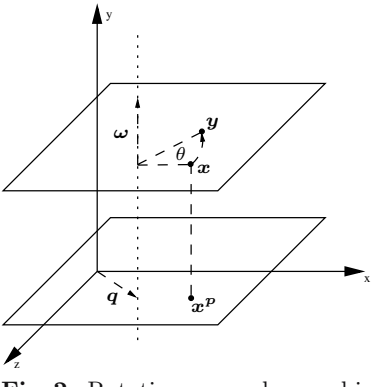

Fig. 2. Rotation around an arbitrary axis in space

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = (\theta_1, \theta_2, \theta_3)^T + R_x(\theta_4) \circ R_y(\theta_5) \circ R_z(\theta_6) \circ R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta_7) \circ \cdots \circ R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta_p) \circ \boldsymbol{x} \tag{3}$$

where $(\theta_1, \theta_2, \theta_3)^T$ is the global translation, R_x is the rotation around the global x-axis etc. and $R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta_i), i \in 7..p$ denotes the rotation around the known axis $(\boldsymbol{\omega_i}, \boldsymbol{q_i})$ with angle θ_i.

Note that '$\circ$' does here not denote multiplication but concatenation of rotations around arbitrary axis' in space, which must evaluated by recursive insertion, as we use the description of equations (1) and (2). Equation (3) gives the position of a point $\boldsymbol{x}$ on a specific segment of the body (e.g. the foot), with respect to joint angles $\boldsymbol{\theta}$ and an initial body pose.

The first derivatives of $f(\boldsymbol{\theta}, \boldsymbol{x})$ with respect to θ give the Jacobian matrix, which is $J_{ki} = \frac{\partial f_k}{\partial \theta_i}$. Of special interest is here the derivative at some specific position $\boldsymbol{\theta^t}$, which reads $J^t_{ki} = \left.\frac{\partial f_k}{\partial \theta_i}\right|_{\theta^t}$.

The Jacobi matrix for the movement of the point $\boldsymbol{x}$ in a kinematic chain is

$$J^t = \begin{bmatrix} 1 & 0 & 0 & & & & & \\ 0 & 1 & 0 & \frac{\partial f}{\partial \theta_4} & \frac{\partial f}{\partial \theta_5} & \frac{\partial f}{\partial \theta_6} & \frac{\partial f}{\partial \theta_7} & \cdots & \frac{\partial f}{\partial \theta_p} \\ 0 & 0 & 1 & & & & & \end{bmatrix} \tag{4}$$

with

$$\begin{aligned} \frac{\partial f}{\partial \theta_6} &= R_x(\theta^t_4) \circ R_y(\theta^t_5) \circ R'_z(\theta^t_6) \circ (R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta^t_7) \circ \cdot\cdot \circ R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta^t_p)\boldsymbol{x}) \\ \frac{\partial f}{\partial \theta_i} &= R_x(\theta^t_4) \circ \cdot\cdot \circ (R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta^t_{i-1}) \circ R'_{\boldsymbol{\omega_i},\boldsymbol{q}}(\theta^t_i) \circ R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta^t_{i+1}) \circ \cdot\cdot \circ R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta^t_p)\boldsymbol{x} \end{aligned} \tag{5}$$

where $\boldsymbol{x}$ corresponds to the initial pose with $\boldsymbol{\theta} = 0$ and $\frac{\partial f}{\partial \theta_{4,5}}$ are similar to $\frac{\partial f}{\partial \theta_6}$.

The partial derivative $\frac{\partial f_k}{\partial \theta_i}$ gives the direction in which the point $\boldsymbol{x}$ will move, if θ_i is changed, which is the tangent vector on the circle, on which $\boldsymbol{x}$ moves around $(\boldsymbol{\omega}_i, \boldsymbol{q}_i)$.

3.3 Simplifying the Jacobian

There is a special case to be considered, if $\boldsymbol{\theta^t}$ is zero. The partial derivative $\frac{\partial f}{\partial \theta_i}$ then simplifies to:

$$\left.\frac{\partial f}{\partial \theta_i}\right|_0 = \left.\frac{\partial R_{\boldsymbol{\omega},\boldsymbol{q}}(\theta_i)}{\partial \theta_i}\right|_0 = \boldsymbol{\omega}_i \times (\boldsymbol{x} - \boldsymbol{q}_i), \tag{6}$$

With this simplification the partial derivatives $\frac{\partial f}{\partial \theta_{4,5,6}}$ for the global rotation are exactly the same as a linearized rotation matrix. When used within Newton Iteration the above equation leads also to a similar, if not equal, linear equation system as for 3D-3D correspondences in Rosenhahn [9].

4 Nonlinear Optimization by Newton Iteration

Assume the intially known model points $X^0 = (\boldsymbol{x^0}_1, \boldsymbol{x^0}_2, ..., \boldsymbol{x^0}_n)$ are observed at $Y = (\boldsymbol{y}'_1, \boldsymbol{y}'_2, ..., \boldsymbol{y}'_n)$. The task of pose estimation is to find the parameters $\boldsymbol{\theta}$ that map X to Y, where each point is transformed by $f^\theta \equiv f(\boldsymbol{\theta}, \boldsymbol{x})$. If the observed points Y are disturbed by noise, a best fit has to be found.

As this minimization problem for pose estimation involves concatenations of *sin* and *cos*, it is analytically hard to calculate. However an assumption can be made that simplifies the problem and makes it possible to evaluate $\boldsymbol{\theta}$ by an iterative method. The assumption made for Newton Iteration is that the function f^θ is locally linear at some point $\boldsymbol{\theta}^t$.

The minimization problem now reads $min_{\boldsymbol{\theta}} \ |\boldsymbol{Y} - F(\boldsymbol{\theta})|_2$ with $F : \mathbb{R}^p \rightarrow \mathbb{R}^{3n}$ and $F(\boldsymbol{\theta}) = \left(f^{\theta}(\boldsymbol{x_1^0}), f^{\theta}(\boldsymbol{x_2^0}), .., f^{\theta}(\boldsymbol{x_n^0})\right)^T$.

The locally linear assumption leads to:

$$F(\boldsymbol{\theta^t} + \boldsymbol{\Delta\theta}) \approx F(\boldsymbol{\theta^t}) + \left.\frac{\partial F}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta^t}} \cdot \boldsymbol{\Delta\theta} \tag{7}$$

The derivative $\left.\frac{\partial F}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta^t}}$ of F at position $\boldsymbol{\theta^t}$ is the Jacobi matrix from above, now for all points from the set X.

Let $\boldsymbol{\epsilon^t} = Y - F(\boldsymbol{\theta^t})$ be the error at iteration step t. In each iteration $\boldsymbol{\Delta\theta}$ is estimated by solving a linear minimization problem:

$$min_{\boldsymbol{\Delta\theta}} \ |\boldsymbol{\epsilon}^t - J\boldsymbol{\Delta\theta}|_2 \tag{8}$$

The solution is given by $\boldsymbol{\Delta\theta} = J^+\boldsymbol{\epsilon^t}$, where $J^+ = (J^T\ J)^{-1}J^T$ is the Pseudo-inverse of J. This may also be solved efficiently by Gauss-Elimination using the linear equation system $(J^T\ J)\boldsymbol{\Delta\theta} = J^T\boldsymbol{\epsilon^t}$, which may be faster than building the inverse of $(J^T\ J)$.

As stated above the derivatives becomes much simpler if $\boldsymbol{\theta^t}$ is zero. This can be achieved by estimating only the relative transforms in each iteration step of the Newton Iteration, which requires recalculation of the known parameters of $F(\boldsymbol{\theta})$. These are the known axes $\boldsymbol{\omega_1}, .., \boldsymbol{\omega_p}$ and their corresponding points $\boldsymbol{q_1}, .., \boldsymbol{q_p}$ in world coordinates. Additionally the new point set X^{t+1} must be evaluated from X^t with respect to the estimated values $\boldsymbol{\Delta\theta}$.

5 ICP Approach

The optimization above assumes that correspondences between the observed points and the model points are known. As the observed points are calculated from depth maps, these correspondences are not known. Equal to [9,1,2,4] we take an Iterative Closest Point (ICP) approach. For each observed point the nearest point on the model is assumed to be the corresponding one. With these correspondences the body pose of the model is calculated. Those two steps are then repeated i times or until the change in the pose parameters is below a certain threshold. Similar to other works, we assume that the model of the observed person is given and that the initial position and initial pose in the first frame are approximately known. Also we assume that there is an upper bound on the displacement of each point on the known model from frame to frame. To fit the body model to the observed point set, a segmentation of the person from the background is necessary. In [1] this is done by using skin color. We assume here only that there are no scene objects (or only negligible parts) within a certain distance to the person by using reweighted least-squares for equation (8).

The calculation of correspondences involves several steps, which are optimized in the following ways.

1. The depth image is randomly subsampled and the 3D-points are calculated from the known focal length and principal point of the camera. To do this efficiently the camera coordinate system is assumed to be equal with the world coordinate system and the body model is positioned initially, such that it is close to the observed point set.

2. The visible points of the model are calculated by rendering each triangle of the body model in a different unique color. The RGB-color of each triangle is then used as an index to get the 3D-points of the triangle by a simple array subscript similar to [3]. The segment to which the model point belongs is also found by an array subscript. For efficient nearest neighbor search, the visible points are ordered into an associative array that uses a binary search tree, where the main order value is that coordinate of the model that has the largest extent, usually the height of the person.
3. Nearest neighbor search. For each observed point the associative array is searched for the point with the next height-value. As the height distance is a lower bound for the Euclidean distance, the search can be stopped, if the height distance is larger than the Euclidean distance to the next point.

6 Depth Estimation

Our motion estimation is based on dense depth information which could be estimated directly from correspondences between images. Traditionally, pair-wise rectified stereo images were analyzed exploiting geometrical constraints along the epipolar lines. More recently, generalized approaches were introduced that can handle multiple images and higher order constraints. See [10] for an overview. Achieving realtime performance on standard hardware has become reality with the availability of free programmable graphics hardware (GPU) and the additional benefit of keeping the CPU free for other tasks like our pose estimation [11]. The results presented here are calculated from depth images generated by a dynamic programming based disparity estimator with a pyramidal scheme for dense correspondence matching along the epipolar lines [5].

7 Results

We tested our implementation with depth images from a real sequence, where a person moved his arms at first in a waving manner, and later crossing his arms in front of his chest. The arms close to the body is a very difficult pose. All contour based approaches will fail here. In figure (3) 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. This shows the robustness of the estimation with respect to inaccurate model geometry. For this sequence 14 DOF were estimated: The global transform, the three shoulder angles and the elbow flexion. Important for the accuracy and speed are the number of iterations and the number of data points (the subsampling rate). The fastest result were achieved by taking approx. 800 data points and ca. 10 iterations per frame. Taking less iterations or data points can lead to invalid tracking for large motions between frames. The processing of one frame took 200ms on a

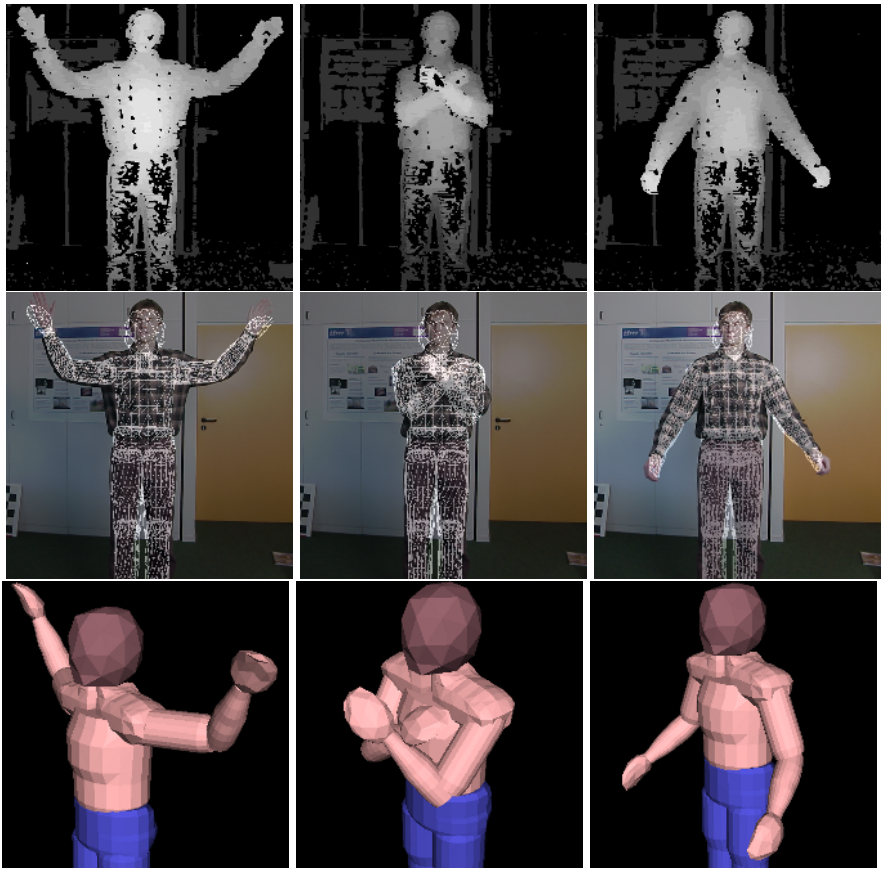

Fig. 3. Top Row:Depth images, middle row: original image overlayed with esimtated model pose, bottom row: model view from the side

Table 1. Comparison with related work

	Demirdjan	Bray	Our
DOF	ca. 18	30	28
correspondences	unknown	45	1000
model complexity	6 simple cylinders	complex hand model linear blend skinned	18 seperately fitted body parts
speed	6-10fps Pentium4 2Ghz	0.22 fps fps Sunfire 1.2Ghz	4 fps Pentium4 3 Ghz

3Ghz Pentium 4 on the average. For a bowing sequence with synthetic data, that involved movement of the arms, legs and head, 26 DOF were estimated. The estimation time increased only slightly to 250ms per frame with about 10 iterations and 1000 data points.

Table (1) shows the effectiveness of our approach. We are able to use far more correspondences for the estimation than Bray. This is probably due to

the analytically derived Jacobian. Processing more data points stabilizes the estimation significantly if noisy depth data has to be used. The approach of Demirdjan is faster than our, however the estimation method is inferior [4] to our direct approach and uses a simpler model and probably less correspondences.

8 Conclusions and Outlook

We showed how body pose estimation by nonlinear optimization methods can be improved using the correct derivatives and described an optimized ICP approach that calculates body pose from depth images in near to real-time. We are optimistic to accelerate our algorithm to more than 10fps by the use of a simpler model and a smarter way of subsampling the depth image, such that less correspondences are sufficient. The stereo algorithm [11] can provide depth images with up to 20fps and is therefore well suited for real-time body pose estimation within HCI applications. Further constraints like self collision and limiting the joint movement to realistic angles will further increase the performance. To integrate constrained motion into Newton Iteration barrier functions can be used. First experiments with barrier functions for the elbow flexion showed promising results. Additional cameras from other views can be easily integrated in the estimation, as they simply provide additional correspondences.

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. Roman Calow, Bernd Michaelis, and Ayoub Al-Hamadi. Solutions for model-based analysis of human gait. In B. Michaelis, editor, *Proc. of DAGM-Symposium*, pages 540–547, Magdeburg, 2003.
4. D. Demirdjian, T. Ko, and T. Darrell. Constraining human body tracking. In *Proceedings of ICCV*, Nice, France, October 2003.
5. L. Falkenhagen. Hierarchical block-based disparity estimation considering neighbourhood constraints, 1997.
6. David G. Lowe. Fitting parameterized three-dimensional models to images. In *Trans. on Pattern Analysis and Machine Intelligence*, pages 13(5):441–450, 1991.
7. T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding: CVIU*, 81(3):231–268, 2001.
8. Ralf Plaenkers and Pascal Fua. Model-based silhouette extraction for accurate people tracking. In *Proc. of ECCV*, pages 325–339. Springer-Verlag, 2002.
9. B. Rosenhahn and G. Sommer. Adaptive pose estimation for different corresponding entities. In *Proc. of DAGM*, pages 265–273. Springer-Verlag, 2002.
10. D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proceedings of IEEE Workshop on Stereo and Multi-Baseline Vision*, Kauai, HI, December 2001.
11. J. Woetzel and R. Koch. Real-time multi-stereo depth estimation on GPU with approximative discontinuity handling. In *CVMP 2004, London, UK*, March 2004.