Full-Body Human Pose Estimation by Combining Geodesic Distances and 3D-Point Cloud Registration

Sebastian Handrich(✉) and Ayoub Al-Hamadi

Institute of Information Technology and Communications,
Otto-von-Guericke-University Magdeburg, Magdeburg, Germany
{sebastian.handrich,ayoub.al-hamadi}@ovgu.de

Abstract. In this work, we address the problem of recovering the 3D full-body human pose from depth images. A graph-based representation of the 3D point cloud data is determined which allows for the measurement of pose-independent geodesic distances on the surface of the human body. We extend previous approaches based on geodesic distances by extracting geodesic paths to multiple surface points which are obtained by adapting a 3D torso model to the point cloud data. This enables us to distinguish between the different body parts - without having to make prior assumptions about their locations. Subsequently, a kinematic skeleton model is adapted. Our method does not need any pre-trained pose classifiers and can therefore estimate arbitrary poses.

1 Introduction

The robust estimation of human poses has a wide range of applications like human-computer interaction, gaming, and action recognition, but is still a challenging task, since the human body is capable of an enormous range of poses. Existing techniques can be classified into several categories. Learning based approaches [1], [2], [3] and [4] are often restricted to previously trained poses and require a large set of training data. The authors of [3], for example, have used almost one million training images. Further, the localization of the various body parts is often not very accurate. In contrast, methods without any prior knowledge [5] can estimate arbitrary human poses but rely on an exact feature point extraction. Image based methods use feature like silhouettes [6], skin color [7] or contours [8], but often lack the ability to resolve ambiguities, e.g. self-occlusion. The recent development of 3d-sensors like Kinect offered the opportunity to overcome the limits of the image-based approaches. In this work, we propose an approach that estimates the full-body human pose from depth images based on geodesic distances. Geodesic distances are independent of the human pose and therefore suitable for a robust body part segmentation. Geodesic distances for pose estimation were used before [9]. In this work, we extent prior works by extracting geodesic paths to multiple surface points which are obtained by registering a rigid torso model to the depth data. This allows for a labeling of geodesic paths and is used to robustly segment the body parts.

© Springer International Publishing Switzerland 2015
S. Battiato et al. (Eds.): ACIVS 2015, LNCS 9386, pp. 287–298, 2015.
DOI: 10.1007/978-3-319-25903-1_25

2 Pose Recognition Method

An overview of the suggested method is provided in Fig. 1. At each time stamp t we are given a depth image $\boldsymbol{D_t}$ that contains the depth data of a user in an articulated pose. Depending on the experimental setup, $\boldsymbol{D_t}$ is either captured by a Kinect or contains simulated data which is generated by rendering an animated 3D-character. It is assumed that the user is within the foreground region. User segmentation is therefore limited to applying a depth threshold to $\boldsymbol{D_t}$. Using a pinhole camera model, we also compute in each frame the corresponding 3D point cloud $\boldsymbol{W_t}$. Our goal is to estimate the full-body pose $\boldsymbol{q_t} \in \mathbb{H}^K$, with $\boldsymbol{q_t} =$

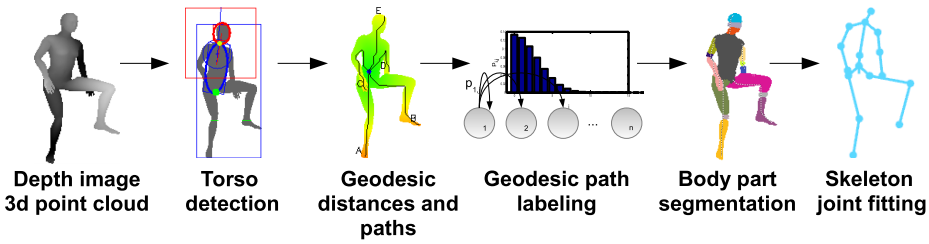

Fig. 1. Suggested method for full-body pose estimation.

$\{q^{to}, q^{ne}, \dots\}$ representing the joint rotations of a hierarchical skeleton model. In each frame, the following steps are performed: a) Fit a 3D-torso model to the torso region (Section 2.1); b) Compute for each 3D-point its geodesic distance to the torso center and extract geodesic paths with maximum length. Determine for each path which limb it represents using a Hidden-Markov-Model (HMM) based approach (2.2); c) Segment individual body parts and fit a kinematic skeleton (2.3).

2.1 3D Torso Model Fitting

In order to determine the position of the limb start point (shoulders, hips, neck), a rigid 3D-torso model is translated and rotated to match with the corresponding points of the point cloud $\boldsymbol{W_t}$. We detect corresponding points by determining for each model point the closest point in $\boldsymbol{W_t}$ and vice versa. This requires an appropriate initialization of the torso model position and orientation. The torso center is detected based on a distance transformation (Fig. 2a). A point is labeled as being located within the torso region $\mathcal{T}$, if its distance is below the half of the maximum distance: $\mathcal{T} = \{[x, y]^T |\ I_{DT}(x, y) < 0.5 \max(I_{DT})\}$. The initial position is set to $\bar{\mathcal{T}}$ and the orientation is given by the largest eigenvector U_0 of the covariance matrix of $\mathcal{T}$ (Fig. 2b). Subsequently, the corresponding 3D-point sets $\boldsymbol{X}, \boldsymbol{Y}$ between the model and $\boldsymbol{W_t}$ are determined. The model is then translated by $\Delta \boldsymbol{p}_{to} = \bar{\boldsymbol{X}} - \bar{\boldsymbol{Y}}$. For the rotation, the matrix

$$\boldsymbol{N} = \begin{bmatrix} S_{xx}+S_{yy}+S_{zz} & S_{yz}-S_{zy} & S_{zx}-S_{xz} \\ S_{yz}-S_{zy} & S_{xx}-S_{yy}-S_{zz} & S_{xy}+S_{yx} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & S_{yy}-S_{xx}-S_{zz} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} \end{bmatrix}$$

is created with $S = Cov(\boldsymbol{X}, \boldsymbol{Y})$ the covariance matrix. The rotation $\dot{q}$ is given by the largest eigenvector of $\boldsymbol{N}$ and the torso is rotated by $q^{to} = \dot{q} \cdot q^{to}$. The fitting is repeated until the residual error $r = \sum_i ||\boldsymbol{X(i)} - \boldsymbol{Y(i)}||_2$ converges. In each frame the transformation of the previous frame is used. If r is too large, the torso fitting is re-initialized using the aforementioned distance transform based method. For a more detailed description of this approach, see [10].

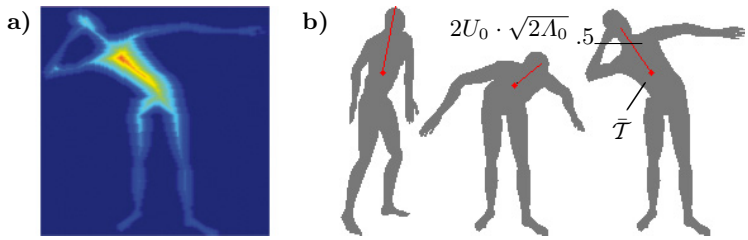

Fig. 2. Torso detection: a) Distance transformation I_{DT}. b) Torso center $\bar{\mathcal{T}}$ and initial orientation (red lines).

2.2 Extraction and Labeling of Geodesic Pathes

We proceed by computing the geodesic distances between all 3D-points $\boldsymbol{W_t}$ and the torso center. For this, a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is constructed. The nodes of the graph are the points of the point cloud $\mathcal{N} = \boldsymbol{W_t}$ and are connected by an edge $\mathcal{E}_{i,j}$, if either an edge criterion $\mathcal{C}_1$ or $\mathcal{C}_2$ is fulfilled. The former connects two nodes if their euclidean distance is below a threshold ϵ_{c1} and if they correspond to adjacent depth pixels (Eq. 1). The second one (Eq. 2) connects two nodes, if all depth image pixels between them have a lower depth value, i.e. are closer to the camera (Fig. 3b).

$$\mathcal{C}_1(i,j) = ||\boldsymbol{w_i} - \boldsymbol{w_j}||_2 \leq \epsilon_{c1} \wedge d(\dot{\boldsymbol{w}}_{\boldsymbol{i}}, \dot{\boldsymbol{w}}_{\boldsymbol{j}}) \leq 1, \tag{1}$$

$$\begin{aligned} \mathcal{C}_2(i,j) = & D_t(\dot{\boldsymbol{w}}_{\boldsymbol{k}}) < \min(D_t(\dot{\boldsymbol{w}}_{\boldsymbol{i}}), D_t(\dot{\boldsymbol{w}}_{\boldsymbol{j}})) \\ & \forall \dot{\boldsymbol{w}}_{\boldsymbol{k}} \in \overline{\dot{\boldsymbol{w}}_{\boldsymbol{i}} \dot{\boldsymbol{w}}_{\boldsymbol{j}}} \wedge d(\dot{\boldsymbol{w}}_{\boldsymbol{i}}, \dot{\boldsymbol{w}}_{\boldsymbol{j}}) > 1, \end{aligned} \tag{2}$$

where $d(\cdot)$ is the spatial 2D distance. Each edge has a weight $w(\mathcal{E}_{i,j})$ which is the Euclidean distance between the connected nodes. The shortest connection (path) between two arbitrary graph nodes $\mathbf{a}, \mathbf{b} \in \mathcal{N}$ is determined using the Dijkstra's algorithm [11]. The cumulative weight of all edges that form a geodesic path is denoted as the geodesic distance $g(\mathbf{a}, \mathbf{b}) = \sum_{\mathcal{E} \in \mathcal{P}(\mathbf{a},\mathbf{b})} w(\mathcal{E})$.

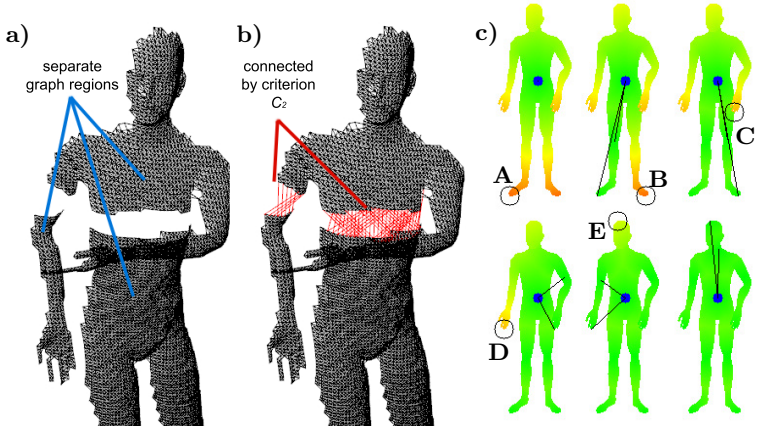

Fig. 3. Geodesic distances: a) Graph with only $\mathcal{C}_1$ applied. b) Graph, if also $\mathcal{C}_2$ is applied, former separated graph region are connected (red lines). c) Detected geodesic maxima, labeled as *A*-*E*. Black lines represent the zero-weight edges.

Limb end points have high geodesic distances from the torso center (red and orange regions in Fig. 3c) and are extracted as follows: In each pass, the node with the highest distance is detected. In order to prevent a re-detection, we connect the node and the node with half the geodesic distance to the torso center with a zero weight edge $w(\mathcal{E}) = 0$ and re-compute the geodesic distances. This is repeated until all maxima are detected (Fig. 3c).

Once the geodesic maxima $\mathbf{M} = \{\boldsymbol{m_i}\}$ and their paths to the torso center $\mathcal{P}(\boldsymbol{m_i}, \mathbf{p_t})$ are detected, we need to label them as the respective limbs $\{\mathcal{L}_j\}_5$: the left and right arm, the head, and both legs; with their start points $\{\boldsymbol{\ell_j}\}_5 = \{\mathbf{p}_{ls}, \mathbf{p}_{rs}, \mathbf{p}_{lh}, \mathbf{p}_{rh}, \mathbf{p}_{ne}\}$, the positions of the shoulders, hips, and the neck, which are obtained from the torso fitting. Labeling is based on the fact that the paths from the torso center to the ends of the limbs pass the limb start points. For example, the path to the left hand passes the left shoulder.

One could assume that is sufficient for a path labeling to determine for each geodesic path the closest limb start point. This, however, does not work for two reasons. Firstly, there are poses in which the path of a limb is very close or even occludes the start point of another limb. For example, when the right arm occludes the left shoulder. In this case, an incorrect closest limb start point would be detected and the path labeling would fail. Another reason is that, when only single points are used for the path labeling, there is no possibility to reject incorrect geodesic paths.

We therefore propose another approach which is based on the comparisons of complete paths. For each geodesic maximum, five hypotheses are generated (Fig. 4) that it represents limb $\mathcal{L}_j$ (Eq. 3)

$$H_{i,j}: \ \mathcal{P}(\mathbf{p_t}, \boldsymbol{m_i}) \mathrel{\widehat{=}} \mathcal{P}(\mathbf{p_t}, \boldsymbol{\ell_j}, \boldsymbol{m_i}), \tag{3}$$

where $\mathcal{P}(\mathbf{p_t}, \boldsymbol{\ell_j}, \boldsymbol{m_i})$ is the concatenation of two paths starting at the torso center $\mathbf{p}_t$ to the geodesic maximum position $\boldsymbol{m_i}$ via the limb start point $\boldsymbol{\ell_j}$.

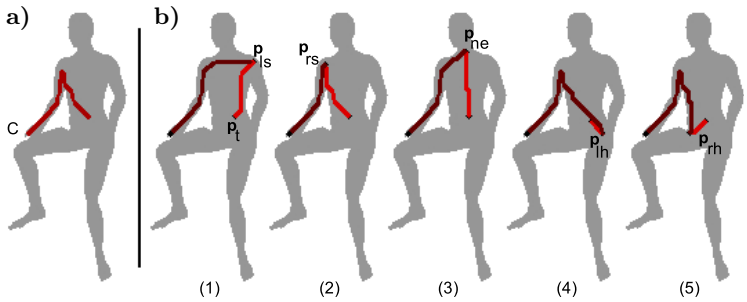

Fig. 4. Detected geodesic path (a) and the hypotheses paths (b).

In order to evaluate the hypotheses, a distance measure is introduced measuring the similarity between two geodesic paths $\mathcal{P}_a$ and $\mathcal{P}_b$. Let $\mathbf{A} = \{a_i\}_N$ and $\mathbf{B} = \{b_k\}_K$ denote the 2D-positions of the elements of $\mathcal{P}_a$ and $\mathcal{P}_b$, respectively. The distance measure is given by Eq. 4.

$$d(\mathcal{P}_a, \mathcal{P}_b) = \frac{1}{N} \sum_{i=0}^{N} e^{-\frac{1}{\sigma}|a_i - b_j|^2}, \; j = h(\mathbf{A}, \mathbf{B}, i). \tag{4}$$

The function $h()$ determines for each element $\mathbf{a}_i$ the index of the corresponding path element in $\mathcal{P}_b$. In a first approach, a nearest-neighbor search was used, but this neglects the sequential order of the path points (Fig. 6a, left). In a second approach (Fig. 6a,right) the sequence of corresponding points is determined using the viterbi algorithm [12]. We define a HMM $\lambda_b = (\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\pi})$, with $\boldsymbol{\phi} = \{\phi_k\}$ the states, $\boldsymbol{\theta} = \{\theta_{ki}\}$ the transition probability matrix, $\boldsymbol{\pi}$ the initial state probability matrix and $\boldsymbol{\eta} = \{\eta_k\}$, a set of emission probability functions. The HMM represents path $\mathcal{P}_b$, so there is one state for each path element $\mathbf{B}$. The elements in $\mathbf{A}$ are considered as an observation sequence. The indices of corresponding path points are then given by the most likely sequence of hidden states $\boldsymbol{q_\phi}$ (viterbi), so that

$$h(\mathbf{A}, \mathbf{B}, i) = \mathbf{q}_\phi(i), \text{ with } \boldsymbol{q_\phi} = \text{viterbi } (\lambda_b, \mathbf{A}) \tag{5}$$

The HMM is in a left-right form, where each state of λ_b is connected to itself and to the next N_λ states. The transition probability (Eq. 6) decreases for more distant states. This topology considers the sequential order of the path points:

$$\theta_{ki} = \begin{cases} \frac{\exp(-(k-i)^2/N_\lambda)}{\sigma_k}, & i \geq k \wedge i < k + N_\lambda \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

where σ_k is a normalization term that ensures that the outgoing transition probabilities add up to one.

The emission probability function $\boldsymbol{\eta}$ determines the probability to observe the path element a_i in state ϕ_k, i.e. at path element b_k and is defined by the Euclidian distance between two path points:

$$\eta_k(a_i) = \exp(-\frac{1}{\sigma}|a_i - b_k|^2). \tag{7}$$

We want to emphasize that the HMM is not trained but constructed out of path $\mathcal{P}_b$, where the transition and emission probabilities are set according to Eq. 6 and Eq. 7, respectively.

Using the distance function (Eq. 4), the hypotheses (Eq. 3) that a path $\mathcal{P}(\boldsymbol{m_i}, \mathbf{p_t})$ represents the limb $\mathcal{L}_j$ are evaluated. This results in a distance matrix Ω^* (Eq. 8). We further assume that limb paths only slightly change between two consecutive frames and compute the path distances ω_{ij}^{-1} to previously assigned limb paths $\mathcal{P}_{\mathcal{L}_j}^{t-1}$ (Eq. 9).

$$\Omega^*(i,j) = \omega_{ij}^* = d(\mathcal{P}(\mathbf{p_t}, \boldsymbol{m_i}), \mathcal{P}(\mathbf{p_t}, \boldsymbol{\ell_j}, \boldsymbol{m_i})) \tag{8}$$

$$\Omega^{-1}(i,j) = \omega_{ij}^{-1} = d(\mathcal{P}(\boldsymbol{\ell_j}, \boldsymbol{m_i}), \mathcal{P}_{\mathcal{L}_j}^{t-1}) \tag{9}$$

Both distances are combined (Eq. 10) as follows: If ω_{ij}^{-1} is above a gating threshold $\omega_\mathrm{G} = 1.0$, the path $\mathcal{P}(\boldsymbol{\ell_j}, \boldsymbol{m_i})$ is too far away from a previously assigned limb path and ω_{ij} is set to $\omega_\mathrm{max} = 2.0$, which prevents in a subsequent step the path from being labeled as limb $\mathcal{L}_j$. If ω_{ij}^{-1} is below the gating threshold, ω_{ij}^* is weighted with ω_{ij}^{-1}. This ensures that if multiple paths are within the gate interval of the previous assigned limb path $\mathcal{P}_{\mathcal{L}_j}^{t-1}$, it becomes more likely for a path that is closer to the previous one to be labeled as $\mathcal{L}_j$.

$$\Omega(i,j) = \omega_{ij} = \begin{cases} \omega_{ij}^*, & \mathcal{P}_{\mathcal{L}_j}^{t-1} \textit{ not seen} \\ \omega_{ij}^* \omega_{ij}^{-1} / \omega_G, & \omega_{ij}^{-1} < \omega_\mathrm{G} \\ \omega_{max}, & \omega_{ij}^{-1} \geq \omega_\mathrm{G} \end{cases} \tag{10}$$

From all possible assignments α_k between the detected geodesic paths and the limbs $\mathcal{L}_j$, the one $\hat{\alpha}$ that minimizes the total assignment distance is selected (Eq. 11). In Fig. 5, the path distances and assignments for a given set of geodesic paths and their hypotheses are shown.

$$\hat{\alpha} = \underset{\alpha_k \in \{\alpha_0 \ldots \alpha_K\}}{\operatorname{argmin}} \sum_{j=0}^{4} \Omega(\alpha_k(j), j) \tag{11}$$

2.3 Body Part Segmentation and Skeleton Fitting

The subsequent steps are behind the scope of this work but briefly mentioned. Having labeled the geodesic paths, it would we possible to adapt the bones of

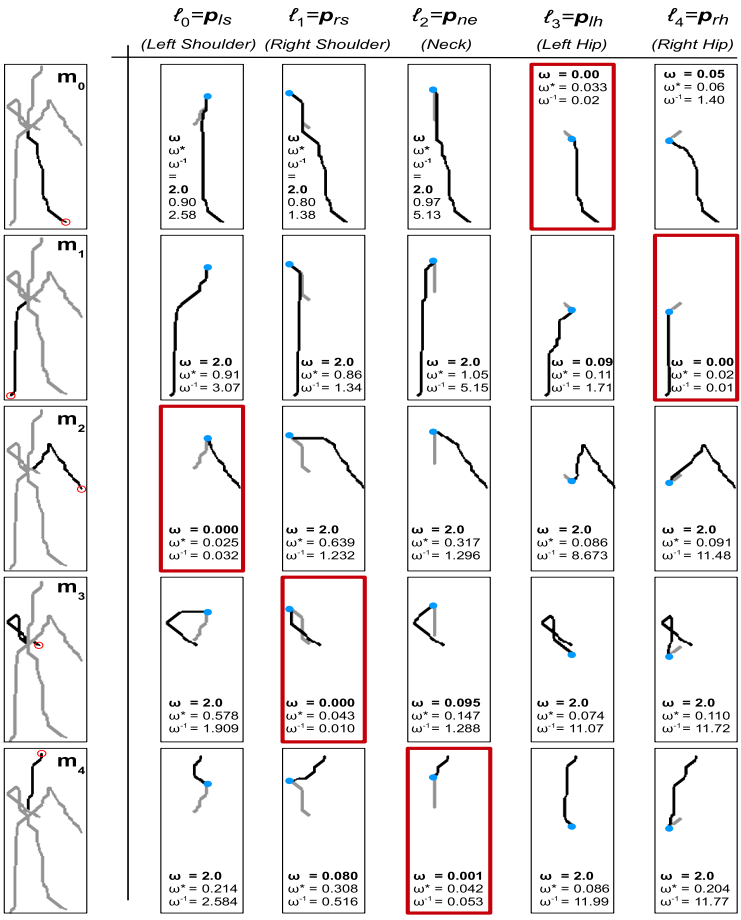

Fig. 5. Detected paths (very left column) towards geodesic maximum positions m_i (red circles) are compared (ω^*) to hypotheses-paths (right columns) defined by the torso center, the geodesic maximum and the limb start point $\boldsymbol{\ell_j}$ (blue dots). They are also compared (ω^{-1}) to previously labeled limb-paths (not shown). Both distances are combined (ω) and the paths are assigned to the limbs so that the cumulative distance is a minimum (red boxes).

a hierarchical skeleton model to the 3D-points that are located along the paths. This is, however, for two reasons inaccurate. The first reason is that paths only reflect the surface of the human body, whereas the bones should be located inside. The second reason is that the paths describe the shortest connection between the points of the body surface. When a limb is bent, the path thus does not run through the center but close to the border of the limb. This can be seen in Fig. 8. So, instead of fitting a skeleton model to the paths, we utilize them to initialize a watershed based body part segmentation (Fig. 6b). Within each body part, the 3D points are shifted inwards the body using the surface normal,

which is obtained from the graph (Fig. 6c). Using the Cyclic coordinate descent method [13], a hierarchical skeleton model is adapted to the shifted points (Fig. 6c) by minimizing the point-to-line distances between the bones and the shifted points.

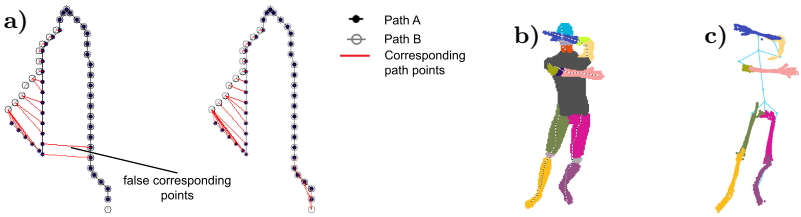

Fig. 6. a) Corresponding path points using nearest neighbor search (left) and viterbi (right) b) Limb paths (white lines) and segmented body parts. c) 3D-points shifted inwards the body.

3 Experimental Results and Discussion

In order to evaluate our pose estimation approach, sequences of synthetic depth images have been generated by rendering an animated 3D-character into an OpenGL depth buffer. The approach has the advantage that the exact ground truth joint positions are known and can therefore be compared to the determined ones. In total, 16 different sequences containing various poses have been created which corresponds to a total number of approx. N=25000 frames. In each sequence, the character is the only rendered object and not more than 45 degrees turned away from the camera. Our skeleton model consists of 16 joints (three for each arm and leg, two for the torso, the neck and the head). Let $z_i(t)$ denote the absolute positions of each joint at frame t. Given the ground truth joint positions $y_i(t)$, we compute in each frame the mean Euclidean distance error (Eq. 12) and the maximum error (Eq. 13).

$$e_m(t) = \frac{1}{16} \sum_{i=1}^{16} |z_i(t) - y_i(t)|_2 \tag{12}$$

$$e_h(t) = \max |z_i(t) - y_i(t)|_2 \, \forall i = 1 \ldots 16 \tag{13}$$

$$e_z(i) = \frac{1}{N} \sum_{t=1}^{N} |z_i(t) - y_i(t)|, \forall t = 1 \ldots N \tag{14}$$

In Fig. 7, the courses of both errors are shown for a single sequence. It can be seen that the mean distance is about 2cm . The maximum error rises up to 10cm. For other sequences the maximum error raised up to 28cm, which is mainly due to an inaccurate fitted elbow joint position or occurs, when the user's

arm points directly to the camera. In this case, there are too few target points for the skeleton fitting and the kinematic chain remains unchanged. For some selected time stamps in 7 the animated 3D-characters are also shown. One can see that the estimated pose (depicted by the blue skeleton) matches the ground truth pose (black skeleton) even in cases of self occlusion.

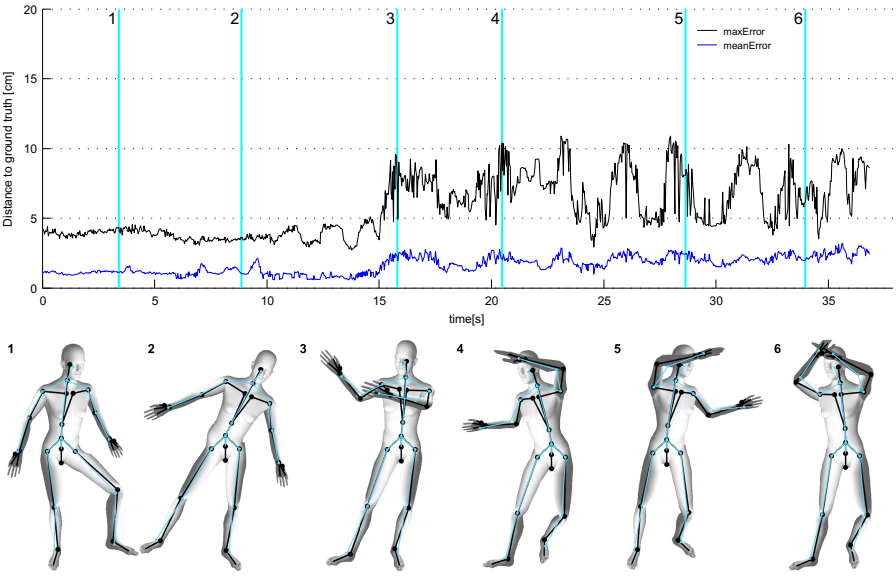

Fig. 7. Top: Mean (blue graph) and maximum (black) joint error for a single sequence. Bottom: Selected poses with ground truth skeleton (black) and the estimated one (blue).

In Fig. 8 more qualitative results of our approach are shown. For several poses, the rendered 3D character together with the estimated and ground truth pose, the geodesic distances and the 3D points of the segmented body parts to which the skeleton is fitted are shown. There are, however, some poses that cannot estimated by our approach. When the user is rotated too far away from the camera, the torso model registration fails. Another case occurs, when the limbs are too close to the torso. In this case, the geodesic paths are not detected. A possible solution would be to perform a local model based search in order to update the skeleton model.

In Fig. 9, we have further computed the per-joint errors (Eq. 14) averaged over all frames. It is between 0.6cm for the head and 3.8cm for the feet and elbows. For comparison, the authors of [9], who have also used geodesic distances, reported averaged per-joint errors between 7cm (shoulders) and 20cm (feets) and an averaged joint distance of $\bar{e}_{Kin} = 10.84$cm. However, these errors are for real depth data. Another graph-based approach was provided by the authors of [14] with reported per-joint distances between 3cm (neck) and 6.8cm (elbow).

Fig. 8. Qualitative results of the pose estimation. The first column for each pose depicts the 3D-character, the ground truth (black) and the estimated pose (blue skeleton). The extracted geodesic paths and distances are shown in the second column. The results of the body part segmentation are shown in the third column.

Our method is suitable for an online processing (20fps). The HMM based path labeling is by far the most computationally expensive step, but can be parallelized for each detected path. Since our approach is a generative approach

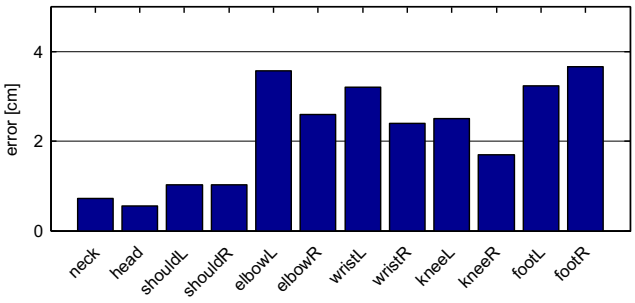

Fig. 9. Per-joint errors averaged over all frames.

that does not need any pre-trained classifiers, the pose estimation is not limited to previously trained poses. A major benefit is that it does not rely on temporal data and as a result cannot get stucked in local minima. Our approach extends prior pose estimation methods based on geodesic distances by extracting the geodesic paths not only to a single but multiple points which are obtained by a torso model registration. As a result, we do not need a specific pose (often T-pose) in which the initial positions of the limbs are determined. Currently, we have extended our method for generating synthetic depth data in order to further examine the influences of parameters like camera position, noise or different character meshes. In a future work, the geodesic distances computed to multiple points will directly be used to segment the individual body parts.

Acknowledgments. This work was supported by Transregional Collaborative Research Centre SFB/TRR 62 (Companion-Technology for Cognitive Technical Systems) funded by the German Research Foundation (DFG).

References

1. Le Ly, D., Saxena, A., Lipson, H.: Pose estimation from a single depth image for arbitrary kinematic skeletons. CoRR (2011)
2. Jaeggli, T., Koller-Meier, E., Gool, L.: Learning generative models for multi-activity body pose estimation. Int. J. Comput. Vision **2**, 121–134 (2009)
3. Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., Blake, A.: Efficient human pose estimation from single depth images. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(12), 2821–2840 (2013)
4. Chang, J.Y., Nam, S.W.: Fast Random-Forest-Based Human Pose Estimation Using a Multi-scale and Cascade Approach **35**(6) (2013)
5. Pons-Moll, G., Baak, A., Helten, T., Muller, M., Seidel, H.-P., Rosenhahn, B.: Multisensor-fusion for 3d full-body human motion capture. In: CVPR, pp. 663–670 (2010)
6. Chen, D.C.Y., Fookes, C.B.: Labelled silhouettes for human pose estimation. In: Int. C. on Inform. Science, Signal Proc. a their App. (2010)

7. Srinivasan, K., Porkumaran, K., Sainarayanan, G.: Skin colour segmentation based 2d and 3d human pose modelling using discrete wavelet transform. Pattern Recognit. Image Anal. **21**(4), 740–753 (2011)
8. Liang, Q., Miao, Z.: Markerless human pose estimation using image features and extremal contour. In: ISPACS, pp. 1–4 (2010)
9. Schwarz, L.A., Mkhitaryan, A., Mateus, D., Navab, N.: Human skeleton tracking from depth data using geodesic distances and optical flow. Image Vision Comput. **30**(3), 217–226 (2012)
10. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. J. of the Optical Society of America **4**, 629–642 (1987)
11. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1**(1), 269–271 (1959)
12. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory **13**(2), 260–269 (1967)
13. Wang, L.-C.T., Chen, C.C.: A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. IEEE Transactions on Robotics and Automation **7**(4), 489–499 (1991)
14. Rther, M., Straka, M., Hauswiesner, S., Bischof, H.: Skeletal graph based human pose estimation in real-time, pp. 69.1–69.12 (2011). doi:10.5244/C.25.69