Upper-Body Pose Estimation Using Geodesic Distances and Skin-Color

Sebastian Handrich and Ayoub Al-Hamadi

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

Abstract. We propose a real-time capable method for human pose estimation from depth and color images that does not need any pre-trained pose classifiers. The pose estimation focuses on the upper body, as it is the relevant part for a subsequent gesture and posture recognition and therefore the basis for a real human-machine-interaction. Using a graph-based representation of the 3D point cloud, we compute geodesic distances between body parts. The geodesic distances are independent of pose and allow the robust determination of anatomical landmarks which serve as input to a skeleton fitting process using inverse kinematics. In case of degenerated graphs, landmarks are tracked locally with a mean-shift algorithm based on skin color probability.

1 Introduction

Gesture recognition plays an important role in real human computer interaction (HCI) environments since it is very intuitive and close to natural human-human interaction. The analysis of gestures in HCI systems requires a robust and real-time capable estimation of the human pose. In the literature pose estimation techniques can be categorized by several criteria: (1) Whether the approach is a learning based method or not, (2) Whether the pose estimation is based on single frames or frame sequences, (3) dimensionality of the input data, i.e. the approach is image based or 3D, (4) use of markers or marker-less. Learning based approaches [1] [2] try to match several observed features with a set of previously trained poses. For this, typical machine learning methods like neural networks or support vector machines are used. An advantage of these methods is that they require a less accurate feature extraction compared to learning free approaches but are restricted to previously trained poses. Methods without any prior knowledge, e.g. [3], require an exact feature extraction but can estimate general poses. Much research has been done on image-based pose estimation techniques which are usually based on features like skin color [4], contours [5] and silhouettes [6] but often lack the ability to resolve ambiguities, e.g. self-occlusions. One possibility to resolve the ambiguities is the use of markers [7]. Typical applications for such an approach are the generation of ground truth data or motion-capture systems. In a real HCI environment, however, the need of wearing markers, is too awkward and not suitable. Another possibility to

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-319-02895-8_64

J. Blanc-Talon et al. (Eds.): ACIVS 2013, LNCS 8192, pp. 150–161, 2013.

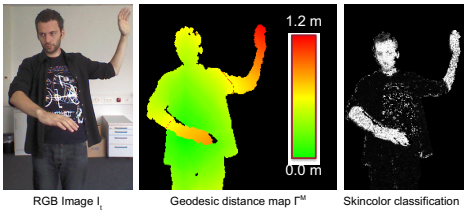

Fig. 1. *Left*: RGB color image of the scene with a user performing a pose. *Middle and Right*: Main features used for pose estimation are: Geodesic distances along the surface of the users body (*middle*) and skin color probability (*right*).

overcome the limits of the image-based pose estimation is the use of 3D data. The recent development in the field of 3D sensors – primarily time-of-flight (ToF) and structured infrared light (IR) based sensors – allows the generation and processing of dense depth maps in real time. Several authors have used 3D sensors for pose estimation [8] [9] [2].

In this work, we propose a method that tracks the upper body pose from depth data. Using a graph-based representation of the 3D information, we compute geodesic distances, i.e. distances along the surface of the human body, and extract anatomical landmarks which are used as input to a preliminary pose estimation. In the case of a degenerated graph, landmarks are determined locally by skin color tracking. A similar method was provided in [10], where the authors used geodesic distances and optical flow.

1. We provide a framework that robustly estimates and tracks human upper body poses in real time.
2. The method does not require any offline training or learning and estimates arbitrary poses, which is important for different HCI scenarios.
3. Due to the robust measurements of the anatomical landmarks based on geodesic distances, our method quickly recovers from tracking failures.
4. Typical parameters such as, the length of the forearm and upper arm, are not a priori required, but determined online.

2 Upper Body Pose Estimation

An overview of our proposed method is shown in figure 2. At each time instant the depth image D_t and color image I_t is captured from the Microsoft Kinect sensor. This capturing is performed in a separate thread, that triple buffers the sensor data. Thus, the reading thread, in which we perform the pose estimation, does not have to wait for the writing process to be completed, which results in a higher processing rate. The segmentation of the observed person is beyond the scope of this paper. We assume that D contains only depth image pixels that belong to an already segmented person.

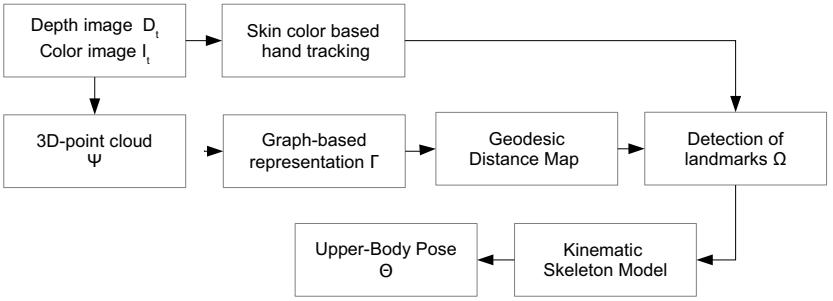

Fig. 2. Overview of the suggested method for upper-body pose estimation

Using the intrinsic camera parameters (principal point and focus length) and D_t, we then compute the 3D point cloud data Ψ. It is an organized point cloud, i.e. each 3D point (vertex) has only one corresponding depth image pixel in D_t. This is in terms of required computation time of great advantage in the next two steps, the computation of the graph based representation Γ of Ψ and the measurement of the geodesic distances (section 2.1).

Especially, when the user touches itself, the graph can, however, contain cycles and thus landmark positions for the elbows and hands are not determinable. To overcome this problem, we additionally use a local mean-shift tracker based on skin color probability to track the hand position in subsequent frames (Section 2.3).

Our goal is to detect 3D feature points (landmarks) Ω for the head, both shoulder, elbows and hands (section 2.2). Given these landmarks Ω, we then use methods of inverse kinematics to find an estimate of the upper-body pose Θ, i.e. to compute the joint rotations θ of a kinematic skeleton model (section 2.4).

2.1 Graph-Based Representation

Given the point cloud data Ψ, we compute a graph-based representation of it. The graph $\Gamma = (n, e)$ consists of nodes n and edges e. The graph creation and measurement of geodesic distances is performed in one single step. Each node n_i is described by three parameters

$$n_i = (\psi, d_g, n_p)_i, \tag{1}$$

where $\psi \in \Psi$ is the corresponding 3D point of the point cloud data, d_g is the total geodesic distance to the root node n_0 of the graph, and node n_p is its parent, i.e. predecessor, node. For the computation of Γ we make use of the fact that each node has a corresponding 2D projection $n'_i = (x, y)_i$ in the depth image. Instead of comparing each 3D point with each other, the graph creation can therefore be done very efficiently in the image domain.

A node n_i is connected to another node n_j by edge e if they fulfill one of two edge criteria, c^1_Γ (eq. 3) or c^2_Γ (eq. 4) . The set of edges is thus defined as:

$$e = \{(n_i, n_j) \in n \times n \mid c^1_\Gamma(i,j) \vee c^2_\Gamma(i,j)\}, \tag{2}$$

with the edge criteria:

$$c^1_\Gamma(i,j) = ||n_i(\psi) - n_j(\psi)||_2 \leq \epsilon_\Gamma \wedge d(n'_i, n'_j) \leq 1 \tag{3}$$

$$\begin{aligned} c^2_\Gamma(i,j) = &||D(n'_i) - D(n'_j)||_2 \leq \epsilon^D_\Gamma \wedge \\ &D(n'_i) > \bar{D}_{ij} + \epsilon_\Gamma \wedge d(n'_i, n'_j) > 1, \end{aligned} \tag{4}$$

and $d(n'_i, n'_j) = ||(x,y)_i - (x,y)_j||_2$ is the 2D distance between the projections of node n_i and n_j to the depth image and $D(n'_i)$ the depth value at location n'_i.

The first criterion (eq. 3) connects two nodes, whose Euclidean distance is below a threshold ϵ_Γ and whose 2D projections are adjacent points. This threshold depends on the resolution and density of the depth image. We used $\epsilon_\Gamma = 0.02m$. The criterion alone, however, is not sufficient. If two nodes, which should be connected by an edge, are separated by another occluding body part, in particular a limb, then the creation of the graph would be incorrect or, at worst, only performed for a subset of all nodes.

Thus, edge criterion c^2_Γ (eq. 4) connects nodes with non adjacent projections $d(n'_i, n'_j) > 1$, if they have similar depth values $||D(n_i) - D(n_j)||_2 \leq \epsilon^D_\Gamma$ and are separated by 3D points that have a less mean depth value $\bar{D}_{ij}$. This is an extension to the method described in [10] as we do not require the projections of two nodes (n_i, n_j) to be adjacent points and can therefore create complete graphs also in the case of partial occlusions.

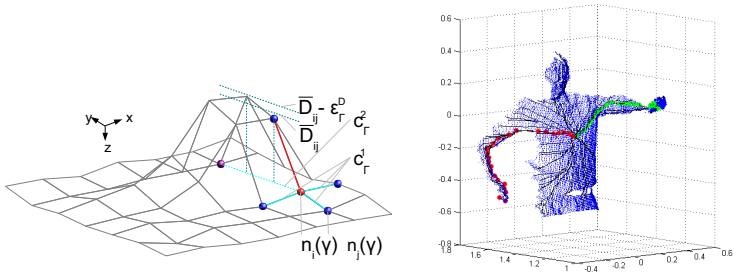

Fig. 3. *Left:* Schematic representation of the graph creation: 3D point cloud data is depicted as a mesh (gray lines). The current node (red sphere) is connected to the first node in each 2D direction that fulfills either edge criterion c^1_Γ or c^2_Γ. *Right:* Results of the graph creation. Blue points show the 3D-point cloud data of the segmented user. A selection of the created graphs is depicted as black lines. The two graphs with maximum geodesic distance are shown as red and green lines.

We denote the Euclidean distance between two nodes connected by an edge as the weight of the edge $w(e_{i,j}) = ||n_i(\psi) - n_j(\psi)||_2$. The geodesic distance d_g of each node is then the cumulated weights of the sequence of edges that belong to the shortest path P back to the root node n_0: $d_g = \sum_{e \in P} w(e)$.

The shortest path P is found using the Dijkstra algorithm. In each iteration step, we search for the node with the smallest total geodesic distance d_g and set it as the current node. We implemented the list of all nodes as a priority queue, because it speeds up the search for current node significantly. Starting at the projected 2D point of the current node we determine for all four 2D-directions (up, down, left, right) the first valid 3D-point that fulfills an edge criterion. This is shown in figure 2.1. When graph creation is completed, we store the total geodesic distance d_g of each node in a 2D map Γ^M.

The choice of an appropriate root node n_0 is important. A good initialization is simply the centroid $\bar{\Psi}$ of the point cloud Ψ. The projection of this point may, however, be occluded by a limb in front of the torso. The graph creation would then begin in the limb and result in incorrect geodesic distances. To overcome this, we define a search window R_Γ centered around the projection of $\bar{\Psi}$ and search for the point with maximum depth D_0 . All nodes in R_Γ that have a similar depth value are then marked as candidates for the root node: $L_0 = \{n_i | \; ||D(n_i') - D_0||_2 \leq \epsilon_{D_0}\}_{n_i' \in R_\Gamma}$. As a root node, we then take the node, which is closest to $\bar{\Psi}$ and element of L_0:

$$n_0 = \underset{n_i \in L_0}{\arg\min} \; ||n_i(\psi) - \bar{\Psi}||_2. \tag{5}$$

2.2 Landmark Detection

In total we use eight landmark positions $\Omega_t = \{\omega_c, \omega_h, \omega_{sl}, \omega_{sr}, \omega_{el}, \omega_{er}, \omega_{wl}, \omega_{wr}\}$ that specify the 3D-position of the body center ω_c, head ω_h, left and right shoulder $\omega_s = (\omega_{sl}, \omega_{sr})$, elbow $\omega_e = (\omega_{el}, \omega_{er})$ and both hands $\omega_w = (\omega_{wl}, \omega_{wr})$.

The center position is identical with the root node of the graph, $\omega_c = n_0(\psi)$. The head appears in D as an elliptical region. We find such regions by a 2D template matching (sum of squared differences) between D and an template image T_H, which contains an ellipse, whose rotation and size depends on the last known head position. The head landmark ω_h is then the centroid of all 3D points that correspond to the resulting template location.

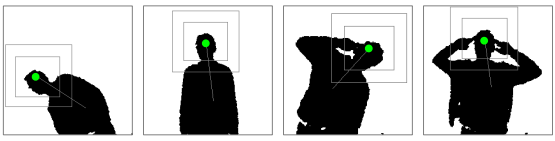

Fig. 4. Results of the head detection based on SSD template matching for various poses. The green points depicts the position of head. The inner rectangle defines the size of the template and the outer rectangle depicts the search region.

To find the hand landmark positions, we threshold the geodesic distance map $\Gamma_t^M > \tau$. We set τ to $\tau = 1m$. For each segmented region R_j, we find the node with maximum geodesic distance d_g^j and compute the mean 3D position x_j:

$$x_j = \sum_{n_i | n_i' \in R_j} \gamma(n_i) \mid d_g(n_i) > d_g^j - \Delta_d^w \tag{6}$$

with Δ_d^w the mean estimated geodesic extent of a hand ($\Delta_d^w = 0.2m$). The set $\{x_j\}$ may also contain 3D locations of the feet or even the head. We reject these locations by considering the Euclidean distance to the detected head landmark $||x_j - \omega_h||_2$ (not further described). The next step is to decide whether the remaining locations in $\{x_j\}$ refer to the left or right hand. For this, the Euclidean distances between nodes of the corresponding paths and the shoulder landmarks are used (Fig.5 left): Tracing back the paths (P_0, P_1) we search for the points (ψ_0^S, ψ_1^S) with minimum Euclidean distance to one of the shoulders and set the hand landmarks according to:

$$(\omega_{sl}, \omega_{sr}) = \begin{cases} (x_0, x_1), & \text{if } ||\psi_0^S - \omega_{sl}||_2 \leq ||\psi_1^S - \omega_{sr}||_2 \\ (x_1, x_0), & \text{otherwise} \end{cases} \tag{7}$$

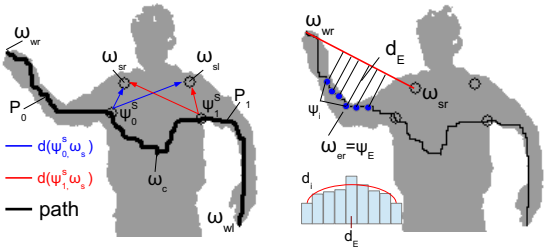

Fig. 5. *Left*: Discrimination between left and right hand based on the minimal Euclidean distances between shoulder landmarks and geodesic paths. *Right*: Detection of the elbow landmarks based on curvature analysis of the corresponding geodesic path.

For the elbow landmarks, we first detect, whether the arm of the person is bent. Here for, the Euclidean distances between the nodes of the hand paths and the 3D line $\omega_w \bar{\omega}_s$ are determined (Fig.5 right). If the maximum distance exceeds a threshold the arm is considered to be bent and we set the elbow landmark ω_e to the corresponding graph node. Otherwise, we set the elbow landmark to the first node of the graph whose geodesic distance to the respective hand landmark is above a threshold. The threshold is initially set to $0.3m$ and updated each time the arm is bent.

2.3 Skin Color-Based Tracking

In cases where landmarks can not be detected by geodesic distances, we determine the hand landmark positions by tracking skin colored regions that are close

to the last known position of the hand. At first, a skin color probability map I_s (Fig. 6b) is computed from the intensity image I_t (Fig. 6a). For each pixel its skin color probability is taken from a pre-computed look-up table (LUT). For this purpose, we have trained a naive Bayes classifier [11]:

$$p(skin|x) = \frac{p(x|skin) \cdot p(skin)}{p(x)} \tag{8}$$

with $x = [cr, cb]^T$ the color components of the pixel in the YCrCb color space, where Y represents the luminance and C_r, C_b the chrominance values. The Bayes classifier was trained in an Histogram-based approach using a set of images containing skin and non-skin colored pixels:

$$p(x|skin) = \frac{n_s}{N_s}; \qquad p(skin) = \frac{N_s}{N_s + N_{\tilde{s}}} \qquad \text{and} \quad p(x) = \frac{n_s + n_{\tilde{s}}}{N_s + N_{\tilde{s}}}$$

where n_s, $n_{\tilde{s}}$ are the skin- and non skin histogram counts for each color $[cr, cb]_i$ and N_s, $N_{\tilde{s}}$ are the total sample sizes of skin and non skin colored pixels, respectively.

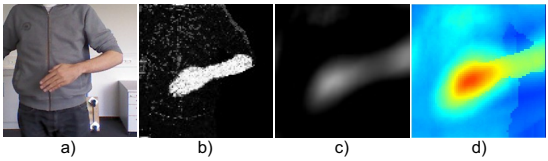

Fig. 6. Skin-color based hand tracking: (a) RGB intensity image. (b) Skin color probability map. (c) Detection of skin colored elliptical regions by means of normalized cross correlation between (b) and a template image. (d) Final probability map for hand position.

The hand appears as an elliptical skin colored region. In a template based step we find such regions in I_s. The template T_h contains a 2D-ellipse, whose size and rotation depend on the camera distance of the last known hand position and the relative position to the last known elbow location. The template match I_m is determined by means of normalized cross-correlation function (Fig. 6c). We assume that the hand position in the current frame is close to that in the prior frame and compute for each valid 3D-point $\in \bar{\Psi}$:

$$I_p = e^{-0.5(\bar{\Psi} - x_w)^2 / \sigma_w} \tag{9}$$

with x_w the last known corresponding hand position. The two probability maps I_p and I_m are combined to a probability map I_h using equation 10 (Lukasiewicz t-norm):

$$I_h = max(I_p + I_m - 1, 0) \tag{10}$$

Thus, I_h has a maximum in elliptical skin-colored regions, that match the size and rotation of the hand and are close to the last known hand position

(Fig. 6d). Its maximum position is found in a mean-shift step. To obtain the hand landmark position ω_w, we then average all 3D points that lie in a search window centered at the maximum location.

2.4 Kinematic Skeleton Model

The skeleton model (Figure 7) is defined by $\Theta = \{\mathbf{x}, q_{r0}, q_{r1}, q_{l0}, q_{l1}\}$. Here $\mathbf{x} = [x_c, x_h, x_{sl}, x_{sr}, x_{el}, x_{er}, x_{wl}, x_{wr}]^T \in \mathbb{R}^3$ denotes the position of the body center, the head, both shoulders, elbows and hands in world coordinates and $q \in \mathbb{H}$ the relative limb rotations of the left and right fore- and upper arm, respectively. The indices for left and right version of the skeleton joints are omitted.

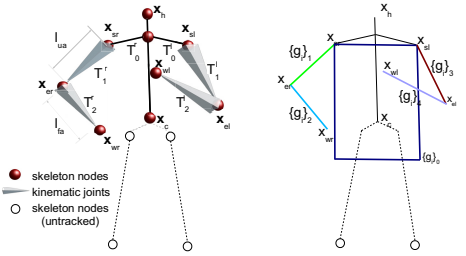

Fig. 7. *Left*: Kinematic skeleton model used in the CCD based fitting step. *Right*: Assignment of 3D points to individual body parts is based on the minimum Euclidean distance to each line in {g}.

The positions of the center, head, and shoulders are set to the corresponding landmark positions $[x_c, x_h, x_s]^T = [\omega_c, \omega_h, \omega_s]^T$. Thus, we only model the arms as kinematic chains: Let $T(q,t)$ denote a transformation with translation t and q. The joint positions are then given by:

$$x_e = T_0 T_1 [0\ 0\ 0\ 1]^T \qquad x_w = T_0 T_1 T_2 [0\ 0\ 0\ 1]^T$$

with transformation matrices:

$$T_0 = T(0, \omega_s) \qquad T_1 = T(q_0, [\pm l_u\ 0\ 0]^T) \qquad T_2 = T(q_1, [\pm l_f\ 0\ 0]^T)$$

and $l_u = ||\omega_e - \omega_s||_2$ and $l_f = ||\omega_w - \omega_e||_2$ the length of the upper- and forearm. The joint rotations (q_0, q_1) are computed by minimizing either

$$e_1 = [\omega_w - x_w, \omega_e - x_e]^T \text{ or} \tag{11}$$

$$e_2 = [\omega_w - x_w]^T \tag{12}$$

depending on whether the hand landmark was found using geodesic distances and we therefore detected an elbow landmark as target position (eq.11) or via skin color tracking (eq.12). We used the Cyclic Coordinate Descent method (CCD) because it is numerically stable, computationally inexpensive, and provides reasonable results for kinematic chains with only a few elements [12].

3 Experimental Results

We have created a set of test sequences to evaluate our proposed method. The sequences were recorded with a Microsoft Kinect for Windows sensor. The resolution was 640 by 480 and 320 by 240 for the color and depth image, respectively. Sampling frequency was 25 frames per second. We assume that no object is between the user and the sensor and that the user is facing the camera. This is a reasonable assumption in a gesture recognition environment. The test database contains both simple and complex poses. Here, simple means that body parts do not overlap. In particular, it means that the created graph is not circular and landmark positions of the hands and elbows can be detected by thresholding geodesic distances. In the complex test sequences, occlusions of body parts and self-contacts occurs. Overall, we have recorded eight test sequences with a length of 70 seconds each. The Microsoft SDK used to control the Kinect Sensor also provides a skeleton. This gives us the possibility to compare our pose estimation method to that of the Kinect. The proposed method is implemented in C++. On a standard dual core computer (2.66 GHz) we can process the sensor data and estimate complete upper-body poses with 22 frames per second. The processing speed differs from sampling frequency because in our implementation capturing and processing are implemented as two separated threads.

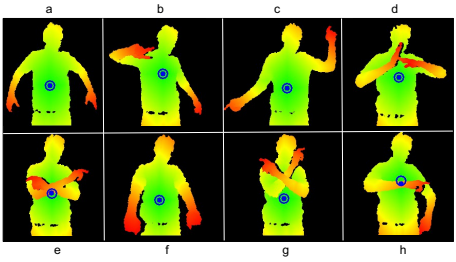

Fig. 8. Robust determination of geodesic distances for various poses. The color denotes the geodesic distance from the root node (blue circle).

We first investigated the measurement of geodesic distances. In Figure 8 the results of the geodesic distance measurement for various poses of the test sequences are shown. The color of each pixel of the depth map represents its geodesic distance to the root node (center of the blue unfilled circle). Green represents a distance of 0 meters and red pixels a distance of 1.2 meters. One can see that we can robustly measure the geodesic distances. In each of the shown examples the depth image pixels that represent the hands have the highest geodesic distances to the root node. We can therefore robustly detect the hands by thresholding the geodesic distances. This also true for complex poses, where both hands are connected (Fig.8d) or the user crosses the arms (Fig. 8e and 8g). As already mentioned the choice of an appropriate root node is important for a correct geodesic distance measurement. The authors in [10] have used the centroid of the point cloud (blue filled circle in Fig. 8). However, this is not sufficient if a

limb is in front of the torso, because the projection of this point to the depth image could be located in an area that belongs to the limbs (e.g. Fig.8h). The geodesic distance measurement would then start in the limb instead of the torso. In fact, in [10] only poses are shown in which the torso is not occluded by a limb. Due to our described correction of of the root node (unfilled blue circles) we are able to compute geodesic distance even if limbs are in front of the torso as long as they are not too close to it.

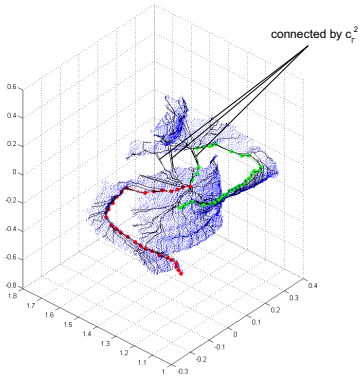

Fig. 9. Graph creation. We do not only connect nodes that are adjacent points in the depth image but also nodes that are separated by foreground objects (edge criterion c_Γ^2). The two paths with the largest geodesic distances (red and green) are also shown.

A further extension to the graph creation described in [10] is that we do not only connect nodes that are adjacent points in the depth image but also nodes that are separated by fore ground nodes (edge criterion c_Γ^2). If there were only graph edges between adjacent points in the depth image (edge criterion c_Γ^1), all 3D points in Fig.8h that are below the right fore arm in the depth image would not have been connected to the graph as they do not fulfill an edge criterion. This is also illustrated in Figure 9 which depicts a perspective view of the 3D point cloud and a subset of the detected geodesic pathes (black lines).

Figure 10 shows results of the landmark detection and skeleton fitting for a subset of poses of the test sequences. In the first row the raw point cloud data (blue dots) and the detected landmark positions are depicted. The second row shows the kinematic skeleton model that was fitted to the landmark positions. For a comparison to the Kinect skeleton, we projected the skeleton of our method (yellow) and that of the Kinect Sensor (magenta) back to depth data (third row). As it can be seen, in all cases the proposed method can determine the pose very well and the projected joint positions match the depth data. Our skeleton matches also very well that of the Kinect Sensor, but does not need any prior training. Tracking problems mainly occurs, if the hands are fully occluded. In this case the hand can neither be detected by thresholding the geodesic distance nor by skin color. However, the tracker can recover from tracking failures as soon as landmarks are determinable by geodesic distances again.

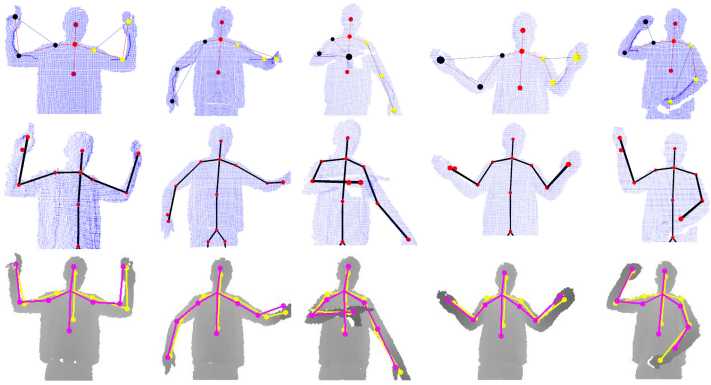

Fig. 10. Evaluation of the proposed method. Each image (a-f) shows the depth data and the projected skeleton of our approach (yellow) and the skeleton of the Microsoft Kinect SDK (magenta). Right next to each depth image a 3D-view of the skeleton is depicted. The thin orange lines show the Microsoft skeleton.

For an qualitative assessment, we manually labeled the positions $\{\hat{x_i}\}$ of the hand, shoulders and elbows in 3D and computed the mean Euclidean distances at each time instant to the joint positions $x_i \in \Theta$ obtained by our method.

$$e_{geo} = \frac{1}{N} \sum_i ||x_i - \hat{x_i}||_2. \tag{13}$$

A similar error was computed between the ground truth positions and the joint positions of the Kinect skeleton. In all 8 test sequences the error of our method was between $40mm$ – $120mm$ and $60mm$ – $140mm$ for the Kinect skeleton. The reason for this difference is the following: The landmark positions of the hands and elbows are located on the paths that were found by the Dijkstra algorithm. It computed for each node of the graph the shortest path to the root node. If the user bents his arm this path, however, does not run throught the center of the limb but is shifted towards its inner side. This is also shown in Fig. 9.

4 Conclusion and Discussion

In this work, we proposed a method for estimating and tracking the human upper-body pose from sequences of depth and color images. The method is a learning-free approach and does not need any pretrained pose classifiers. We can therefore track arbitrary poses as long as the user is not turned away from the camera and there is no object between the user and the camera.

At first, we segment the user based on the depth image and determine a graph based representation of the 3D-data. Using this graph, we measure the geodesic distances along the surface of the users body. By thresholding the geodesic distances, landmarks for hand and elbow locations can be obtained. The distinction

between right and left arm are done by backtracking the corresponding geodesic pathes. In the cases where geodesic distances could not be measured (degenerated graph), hand landmarks are determined by tracking skin colored regions by means of a mean-shift algorithm. The presented experimental evaluation showed that we can robustly and exactly estimate arbitrary poses, which builds the basis for a subsequent gesture recognition process. The proposed method is real-time capable and can track rapid limb movements. Problems can occure, when multiple skin-colored regions exists, e.g. skin colored clothes. This is due to the simplicity of the used skin color tracker. In [13] we presented a multi hypotheses based approach tracker which we will integrate into the proposed approach.

References

1. Jaeggli, T., Koller-Meier, E., Gool, L.: Learning generative models for multi-activity body pose estimation. IJCV 83(2), 121–134 (2009)
2. Le Ly, D., Saxena, A., Lipson, H.: Pose estimation from a single depth image for arbitrary kinematic skeletons. CoRR, vol. abs/1106.5341 (2011)
3. 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)
4. 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)
5. Liang, Q., Miao, Z.: Markerless human pose estimation using image features and extremal contour. In: ISPACS, pp. 1–4 (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. Wang, Y., Qian, G.: Robust human pose recognition using unlabelled markers. Appl. of Comp. Vision, 1–7 (2008)
8. Soutschek, S., Penne, J., Hornegger, J., Kornhuber, J.: 3-d gesture-based scene navigation in medical imaging applications using time-of-flight cameras. In: CVPR Workshops, pp. 1–6 (2008)
9. Hu, R.Z.-L., Hartfiel, A., Tung, J., Fakih, A., Hoey, J., Poupart, P.: 3d pose tracking of walker users' lower limb with a structured-light camera on a moving platform. In: CVPRW, pp. 29–36 (2011)
10. Schwarz, L.A., Mkhitaryan, A., Mateus, D., Navab, N.: Estimating human 3d pose from time-of-flight images based on geodesic distances and optical flow. In: IEEE Automatic Face Gesture Recog. a. WS, pp. 700–706 (2011)
11. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. Int. J. Comput. Vision 46(1), 81–96 (2002)
12. Wang, L.-C.T., Chen, C.C.: A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. Robotics and Automation 7(4), 489–499 (1991)
13. Handrich, S., Al-Hamadi, A.: Multi hypotheses based object tracking in hci environments. In: ICIP, Orlando, USA, pp. 1981–1984 (2012)