Image and Vision Computing 30 (2012) 217–226

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Human skeleton tracking from depth data using geodesic distances and optical flow ☆

Loren Arthur Schwarz *, Artashes Mkhitaryan, Diana Mateus, Nassir Navab

Computer Aided Medical Procedures (CAMP), Department of Informatics Technische Universität München (TUM), Boltzmannstr. 3, 85748 Garching, Germany

ARTICLE INFO

Article history:
Received 17 July 2011
Received in revised form 21 November 2011
Accepted 4 December 2011

Keywords:
Human pose estimation
Depth imaging
Geodesic distances

ABSTRACT

In this paper, we present a method for human full-body pose estimation from depth data that can be obtained using Time of Flight (ToF) cameras or the Kinect device. Our approach consists of robustly detecting anatomical landmarks in the 3D data and fitting a skeleton body model using constrained inverse kinematics. Instead of relying on appearance-based features for interest point detection that can vary strongly with illumination and pose changes, we build upon a graph-based representation of the depth data that allows us to measure geodesic distances between body parts. As these distances do not change with body movement, we are able to localize anatomical landmarks independent of pose. For differentiation of body parts that occlude each other, we employ motion information, obtained from the optical flow between subsequent intensity images. We provide a qualitative and quantitative evaluation of our pose tracking method on ToF and Kinect sequences containing movements of varying complexity.

1. Introduction

Human gestures are a natural means of communication and allow complex information to be conveyed. Using gestures for interaction with computer-assisted systems can be of great benefit, particularly in scenarios where traditional input devices are impractical, such as the medical operating room [1]. In order to track human full-body pose in real-time, camera-based motion capture systems can be used that typically require a person to wear cumbersome markers or suits. Lately, research has focussed on markerless human pose estimation [2,3]. However, even if multiple cameras are used, this task is challenging due to the complexity of human movements and their highly variable visual appearance in images that is sensitive to illumination and occlusions [4,5].

Recent technological advances have lead to the development of novel depth cameras that allow acquiring dense, three-dimensional scans of a scene in real-time, without the need for multi-camera systems (Fig. 1). Such depth images are almost independent of lighting conditions and variations in visual appearance, e.g. due to clothing. In every image pixel, these cameras provide a measurement of the distance from the camera sensor to the closest object surface. Typical depth cameras are based on the Time of Flight (ToF) principle that requires complex hardware and is thus expensive. Recently, the Microsoft Kinect device appeared in the consumer electronics market and made depth imaging available to a broad audience of researchers, resulting in a multitude of novel computer vision applications [6,7].

ToF cameras measure the distance from a sensor array into the scene by emitting modulated infrared light rays and measuring the phase shift of incoming rays after reflection from objects in the scene [8]. While ToF cameras have a relatively low resolution, they simultaneously provide co-registered depth and intensity images which can be used as regular grayscale images. Cameras based on the structured light principle project a known infrared light pattern into the scene and infer depth at any image location by evaluating the distortion of the projected pattern. The Kinect combines a structured light camera with a regular RGB camera that can be calibrated to the same reference frame. Despite their favorable properties, depth cameras provide data that suffers from noise and estimating human full-body pose remains a challenging problem [9].

Typical noise in ToF cameras is due to systematic errors and other sources impacting the measurements [8]. For example, each ToF camera has a particular distance error that varies throughout the measurement range. Moreover, surfaces with low infrared reflectance properties (often black surfaces) result in inconsistent depth estimates. So-called "flying pixels" appear at object boundaries due to the fact that individual ToF camera pixels can receive multiple depth measurements. Finally, when observing a full human with a ToF camera, the currently low resolution leads to thin body parts being represented by just a few pixels. The Kinect provides higher resolution depth images, but different surface properties can also lead to depth artifacts. In addition, the structured light principle used by the Kinect causes shadow-like artifacts to appear around objects (or body parts) that are closer to the camera.

In this paper, we propose a method that allows tracking the full-body movements of a person from depth images, suitable for gesture-based interaction. While learning approaches for human pose estimation (e.g. [2,3,10]) rely on training data and are thus restricted

☆ This paper has been recommended for acceptance by special issue Guest Editors Rainer Stiefelhagen, Marian Stewart Bartlett and Kevin Bowyer.

* Corresponding author. Tel.: +49 89 289 19412.
E-mail address: schwarz@in.tum.de (L.A. Schwarz).

0262-8856/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2011.12.001

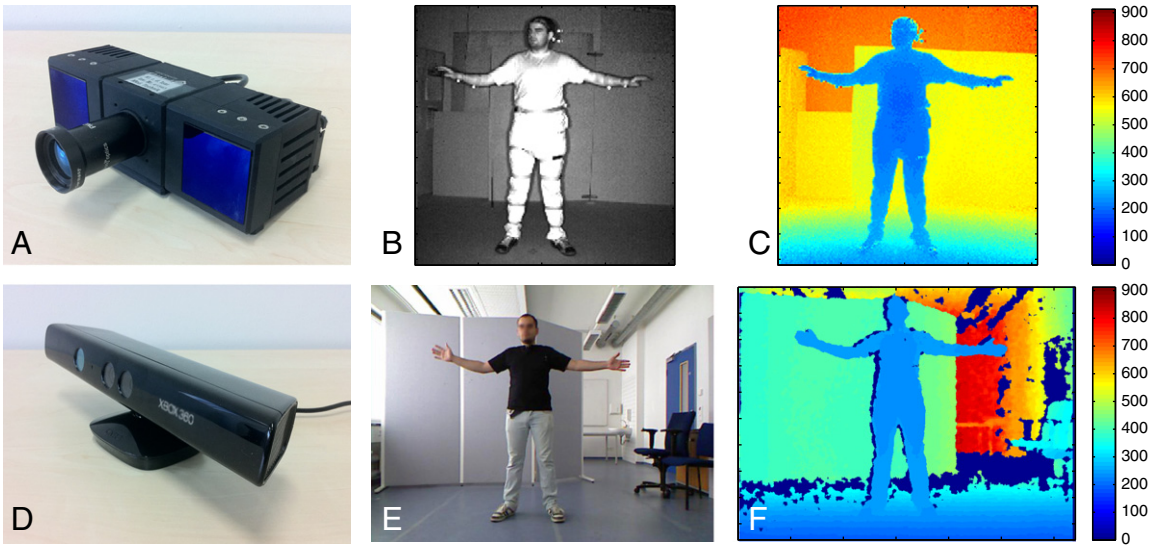

Fig. 1. Time of Flight (ToF) camera and Kinect examples. A: ToF camera with two infrared flashes and optics at the center. B: ToF amplitude image. C: ToF depth image with color-coded depth in centimeters. D: Kinect device with an infrared projector, an infrared camera and an RGB camera. E: Kinect RGB image. F: Color-coded Kinect depth image.

to a particular set of movements, our method can track general, previously unseen motions. The method is based on robustly identifying anatomical landmarks in the depth data that then serve as targets for fitting a skeleton using inverse kinematics. We propose to represent the background-subtracted depth data by means of a graph that facilitates detection of body parts. In addition, we use the optical flow between subsequent intensity or RGB images for depth disambiguation when body parts occlude each other.

Representing the 3D points on the surface of a person as a graph allows us to measure geodesic distances between different points on the body. While the Euclidean distance between two body points is measured through 3D space and thus can vary significantly with body movement, the geodesic distance is defined along adjacent graph nodes, i.e. along the surface of the body. Consequently, the geodesic distance between two points on the body, e.g. the centroid and an extremity, can be assumed constant, independent of body posture [11,12] (Fig. 2). We can therefore extract anatomical landmarks by searching for points at mutual geodesic distances that correspond to the actual measurements of a person. Using only depth and intensity data also decreases our dependence on visual appearance. Thus, we avoid typical problems that arise when using intensity-based feature descriptors for interest point detection, e.g. lack of texture and illumination or perspective changes.

When body parts occlude each other, the graph constructed from the 3D points can degenerate. In this case, separating the occluding body part from the part behind it becomes difficult, leading to undesired graph edges. These edges between points on different body parts result in erroneous geodesic distances, and consequently, to undetected anatomical landmarks. We address this issue by taking into account the motion occurring between subsequent frames. In particular, we identify and remove the undesired graph edges based on optical flow fields that are computed from the intensity or RGB images.

Our method takes full advantage of the available information by simultaneously using depth images (for segmentation and generation of 3D points), intensity or RGB images (for optical flow) and the graph-based representation (for geodesic distances). Compared to other depth-camera-based human body tracking approaches, ours is able to quickly recover from tracking failures due to the robust

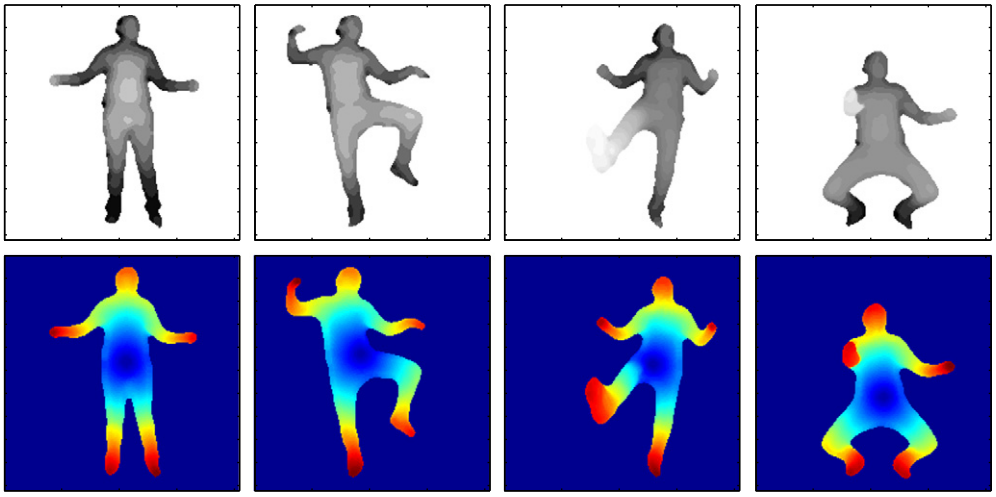

Fig. 2. Illustration of the robustness of geodesic distances against pose changes. Top: Background-subtracted depth images for various poses. Bottom: Geodesic distances from the body center to all other surface points. Colors range from blue (zero distance) to red (maximal distance). Note that the distance to hands and feet remains almost constant across all poses.

anatomical landmark detection approach. The experiments show that we are able to efficiently track the full-body pose in several ToF and Kinect sequences containing various human movements.

2. Related work

Techniques for human pose estimation from visual observations can be broadly categorized into learning-based approaches that facilitate the problem by means of training data (e.g. [2,3,13,10]) and approaches estimating human pose parameters from observed features without prior knowledge. A disadvantage of the former is that pose estimation is typically restricted to a set of activities known to the algorithm beforehand. For instance, Jaeggli et al. [3] use monocular images and extract human silhouettes as an input to a generative pose estimation method trained on walking and running. The generative model presented in [10] is also limited to a set of trained motions, however multiple activities can be considered simultaneously. Discriminative pose estimation approaches have also been proposed, e.g. in [13]. Here, body poses for a pre-determined activity are predicted from voxel data, obtained from a 3D reconstruction system, using non-linear regression. Similarly, Fossati et al. [14] train a regressor to predict human body poses from observations in video images.

Methods that do not use prior knowledge for pose estimation (e.g. [4,15,5,16–18]) are typically more dependent on reliable feature extraction, as the appearance of the human body is heavily affected by illumination and pose changes, and by noise in the observation data. Moreover, efficient state inference techniques are required to deal with the high dimensionality of full-body pose space. Kehl and van Gool [4] cope with these issues by using a multi-camera setup and generating 3D volumetric reconstructions for human pose estimation. Bandouch et al. [5] employ a particle filter for state estimation using a multi-camera reconstruction system with the disadvantage of non-interactive frame rates. In [16], body poses are estimated by assisting a multi-camera system with inertial sensors attached to the human body.

To overcome the limitations of standard camera-based observations, several authors have recently used ToF cameras for analysis of human motions (e.g. [19–22,10]). In [20], a system is described that recognizes simple hand gestures for navigation in medical imaging applications. Hand movements are also tracked in the work presented in [23]. The method of Jensen et al. [21] allows tracking the movement of legs in side-views for medical gait analysis. Holte et al. [24] propose a method that integrates ToF range and intensity images for human gesture recognition. Their approach is not used for pose tracking, but is able to classify upper-body gestures, such as raising an arm. The authors avoid identifying anatomical landmarks by using a global pose descriptor. Zhu et al. [15] present a full-body pose estimation system that relies on fitting templates for each body part to the ToF data. In [22], the authors combine a template fitting technique based on dense point correspondences with an interest point detector for increased robustness. While the approach can track full-body motion, it relies on an independent, heuristic treatment for each body part. In [10], a ToF-based method is described that simultaneously estimates full-body pose and classifies the performed activity. As opposed to our method, the system can only process movements known a priori.

After its recent appearance, the Microsoft Kinect has caused a plethora of applications to appear, including approaches for human body tracking. For instance, Hu et al. [6] extract leg movement information for medical gait analysis using a Kinect that is mounted on a moving walker. While appropriate for their intended purpose, this method cannot track full-body poses. Shotton et al. [7] present a method for body joint detection in Kinect depth images. Their method is based on a learning approach that classifies individual depth image pixels to one of multiple body parts, such as upper arm, elbow or thigh. A random forest classifier is trained with a large database of labeled depth images. The training database is generated by applying motion capture data to a variety of synthetic body models, thus obtaining training data for various body proportions. While this extensive training approach decreases the sensitivity of the method to person-specific appearance, it still depends on the movements contained in the training dataset.

Similar to our approach, Plagemann et al. [11] use a graph representation of the 3D data for detection of anatomical landmarks. Their technique extracts interest points with maximal geodesic distance from the body centroid and classifies them as hands, head and feet using a classifier trained on depth image patches. The method does not explicitly address the problem of self-occlusions between body parts and reportedly struggles in such situations. Without modifying the interest point detection technique, the authors add in [9] a pose estimation method embedded in a Bayesian tracking framework. Our proposed method uses optical flow measured in ToF intensity images to cope with body self-occlusions. Optical flow has been used in [25] for motion estimation and segmentation of a person in a monocular pedestrian tracking application. Okada et al. [26] describe a person tracking method that combines disparity computation in a stereo setup with optical flow. Similar to our approach for disambiguation in the case of self-occlusions, an interest region map is propagated through the tracking sequence using the computed flow vectors. While this method allows tracking the bounding boxes of the head and upper body, our technique estimates the joint angles of a full skeleton body model in every frame. To the best of our knowledge, using optical flow for segmentation of occluding body parts in depth-image based human body tracking is a novel approach, enabling us to track arbitrary full-body movements. Initial results of this work have been presented in [27].

3. Human full-body tracking method

We are given a sequence $\{\mathcal{T}_t\}_{t=1}^{N}$ of N depth measurements, where each $\mathcal{T}_t = (\mathbf{D}_t, \mathbf{I}_t)$ consists of a depth image $\mathbf{D}_t$ and an intensity image $\mathbf{I}_t$, both of size $n_x \times n_y$. When using a ToF camera, $\mathbf{I}_t$ denotes the gray-scale amplitude image. For the Kinect device, $\mathbf{I}_t$ refers to the RGB image. While ToF cameras acquire both images with the same sensor, the Kinect uses two separate cameras for the depth and the RGB images. Thus, the RGB image first needs to be aligned to the reference frame of the depth image, e.g. using stereo calibration techniques. The Kinect programming interface by PrimeSense [28] also provides methods allowing to access color pixels corresponding to given depth image coordinates. In the following, we assume that there is a correspondence between any depth pixel $\mathbf{D}_t(i,j)$ and an intensity image pixel $\mathbf{I}_t(i,j)$.

3.1. Method outline

Given a sequence of depth and intensity images of a person, our goal is to estimate the full-body pose $\mathbf{q}_t \in \mathbb{R}^d$ of the person at each frame t, parameterized by the d joint angles of a skeleton model. In an initialization step, the person faces the camera for a few seconds and takes on a T-pose with the arms extended sideways. After initialization, the method consists of the following steps, performed in each frame (Fig. 3):

1. Depth image preprocessing. The depth image is first median filtered to reduce the amount of noise. In addition, a 3D point cloud representation is computed from the depth data (Section 3.2).
2. Graph construction from depth data. We create a graph with the 3D points as vertices and introduce edges whenever two 3D points fulfill a specific spatial neighborhood property (Section 3.3).
3. Creation of geodesic distance map. The graph is used to generate a map of geodesic distances from the body center to all other body points. The distances represented in this map can be assumed invariant to articulation changes (Section 3.4).

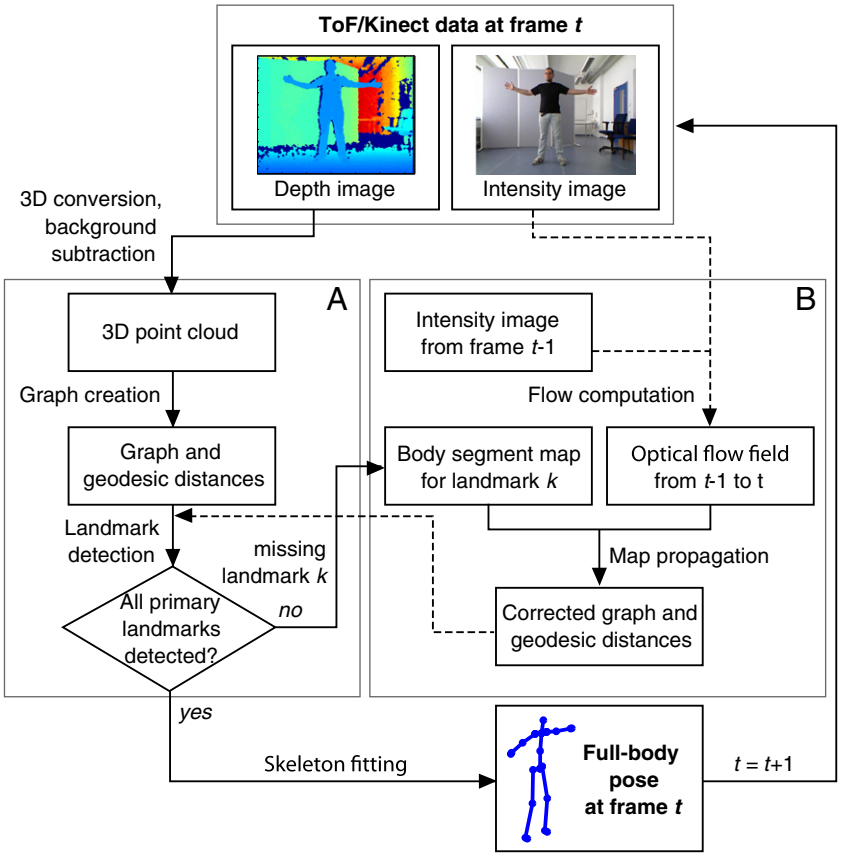

Fig. 3. Schematic of the depth-image human body tracking method. In each frame t, the algorithm constructs a geodesic distance graph based on the 3D-converted depth image and extracts anatomical landmarks (A). For each undetected landmark k, the disambiguation process using optical flow on the intensity image is executed (B). The corrected geodesic distance graph for a body part k allows detecting the missing anatomical landmarks.

4. Localization of anatomical landmarks. We locate L anatomical landmarks $\mathcal{P}_t = \{\mathbf{p}_i^t\}_{i=1}^L$ using the geodesic distance map, where $\mathbf{p}_i^t \in \mathbb{R}^3$, and determine the discrete landmark labels $\alpha(\mathbf{p}_i^t)$, e.g. *head*, *left knee*, *right hand*. (Section 3.5).
5. Disambiguation using optical flow. The optical flow between the previous and the current frame, measured using the intensity images $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$, is used to track body parts that occlude each other (Section 3.6).
6. Skeleton fitting to anatomical landmarks. We finally employ a model-based skeleton fitting approach to estimate the full-body pose that best fits the extracted anatomical landmarks (Section 3.7).

3.2. Depth image preprocessing

After median-filtering the incoming depth image $\mathbf{D}_t$, we subtract a previously recorded background image to segment the person in the foreground. The underlying assumption is that indoor scenes with a static background are the typical setting for our method. We then transform the segmented depth image into a 3D point cloud based on the known intrinsic parameters of the depth camera. Let $\mathcal{X}_t = \{\mathbf{x}_{ij}\}$ denote the resulting set of $n_x n_y$ 3D points. We assume that, after creating the 3D point cloud, we are still able to identify the 2D depth image coordinates belonging to a given 3D point. Our notation indicates that the point $\mathbf{x}_{ij}$ corresponds to the depth image pixel with coordinates (i,j).

3.3. Graph construction from depth data

We construct a graph $G_t = (V_t, E_t)$, where $V_t = \mathcal{X}_t$ are the vertices and $E_t \subset V_t \times V_t$ are the edges. Whether two vertices, i.e. 3D points, are connected with an edge or not is based on their spatial distance in 3D and on their vicinity in the 2D depth image. We thus define the set of edges as

$$E_t = \left\{ \left(\mathbf{x}_{ij}, \mathbf{x}_{kl}\right) \in V_t \times V_t \mid \|\mathbf{x}_{ij} - \mathbf{x}_{kl}\|_2 < \delta \wedge \|(i,j)^\top - (k,l)^\top\|_\infty \leq 1 \right\}, \quad (1)$$

where $\|\cdot\|_2$ is the Euclidean and $\|\cdot\|_\infty$ is the maximum norm and $(i,j)^\top$, $(k,l)^\top$ are the 2D coordinates of the two points $\mathbf{x}_{ij}, \mathbf{x}_{kl}$ in the depth image. For each edge $e = (\mathbf{x}, \mathbf{y}) \in E_t$, we store a weight $w(e) = \|\mathbf{x} - \mathbf{y}\|_2$. We thus connect points with a 3D Euclidean distance of less than δ that project to neighboring pixels in 2D. Incorporating the 2D neighborhood allows us to efficiently construct the graph in linear time by traversing the depth image once. The classical way of constructing a neighborhood graph from a set of 3D points would first require the expensive computation of all pairwise point distances in 3D.

3.4. Creation of geodesic distance map

Using G_t, we are able to measure geodesic distances between different body locations, as opposed to measuring Euclidean distances between the corresponding 3D points. The advantage of geodesic distances is that the local topology of the body is taken into account. The geodesic distance $d_G(\mathbf{x}, \mathbf{y})$ between two points $\mathbf{x}, \mathbf{y} \in V_t$ is given by

$$d_G(\mathbf{x}, \mathbf{y}) = \sum_{e \in SP(\mathbf{x},\mathbf{y})} w(e), \quad (2)$$

where $SP(\mathbf{x}, \mathbf{y})$ contains all edges along the shortest path between $\mathbf{x}$ and $\mathbf{y}$. Intuitively, the geodesic distance between two locations on the body is thus the length of the shortest path over the body surface.

Our central assumption is that all anatomical landmarks remain at a nearly constant geodesic distance from the body center of mass, independent of body pose [11]. Let $\mathbf{c}^t$ denote the 3D point that is closest to the centroid of the segmented point cloud $\mathcal{X}_t$. We then define the geodesic distance map $\mathbf{M}_t$, of the same size as the depth image $\mathbf{D}_t$, as

$$\mathbf{M}_t(i,j) = d_G\left(\mathbf{c}^t, \mathbf{x}_{ij}\right). \tag{3}$$

Examples of geodesic distance maps are given in Fig. 2. To determine the geodesic distance from the body center for the 3D point belonging to the depth image pixel $\mathbf{D}_t(i,j)$, we thus simply evaluate the geodesic distance map at $\mathbf{M}_t(i,j)$. Given a single source point, the shortest paths to all other points in the graph can be computed using Dijkstra's single source shortest paths algorithm. In an efficient implementation, this algorithm has a runtime complexity of $O(|V_t| \cdot \log|V_t|)$.

3.5. Detection of anatomical landmarks

Having constructed the graph G_t and the geodesic distance map $\mathbf{M}_t$ in frame t, we proceed by locating $L=11$ anatomical landmarks $\mathcal{P}_t = \{\mathbf{p}_i^t\}_{i=1}^L$ and determining their labels $\alpha(\mathbf{p}_i^t)$. We distinguish between *primary* landmarks $\mathcal{P}_t'$ (body center, head, hands, feet) and *secondary* landmarks P_t'' (chest, knees, elbows).

The first *primary* anatomical landmark is given by the body center of mass $\mathbf{c}^t$. To extract the extremities, we use the geodesic distance map to select all points $\mathbf{x}$ with $d_G(\mathbf{x},\mathbf{c}^t) > \tau$. Here, τ is a person-specific threshold that approximates the distance from the body center to the shoulders. We therefore obtain spatially isolated sets of points that we treat as belonging to different limbs. For each of these isolated sets, we store the point with largest geodesic distance from the body center, yielding the set of primary anatomical landmarks $\mathcal{P}_t'$.

Given the locations of the primary anatomical landmarks, we need to determine their labels in order to detect the secondary landmarks, i.e. the chest, elbows and knees. During initialization, where the person takes on a T-pose, we create an initial labeling of the anatomical landmarks. Each landmark $\mathbf{p}_i^0$ detected in the initialization frame ($t=0$) is assigned an appropriate label $\alpha(\mathbf{p}_i^0)$ based on the assumed T-pose. In any subsequent frame t, we determine the labels for the primary landmarks by matching the detected positions $\mathbf{p}_i^t$ to the known landmarks in the previous frame. The label for the i-th landmark is thus

$$\alpha\left(\mathbf{p}_i^t\right) = \alpha\left(\bar{\mathbf{p}}^{t-1}\right), \text{ where } \bar{\mathbf{p}}^{t-1} = \underset{\mathbf{p}\in\mathcal{P}_{t-1}'}{\arg\min} \|\mathbf{p}_i^t - \mathbf{p}\|_2. \tag{4}$$

We can then extract the location of the *secondary* landmarks $\mathcal{P}_t''$, i.e. the chest, elbows and knees, by measuring geodesic distances from the localized primary anatomical landmarks. That is, we select points on the body as the chest, the elbows and the knees that are located at respective distances from the body center, the hands and the feet.

3.6. Depth disambiguation using optical flow

In cases when the extremities are clearly separated from each other, the graph-based landmark identification approach allows us to detect all primary and secondary landmarks. However, when body parts occlude each other, the graph G_t will likely contain edges that connect points on different body parts. In such a situation, two points $\mathbf{x},\mathbf{y}\in V_t$ on distinct body parts can easily satisfy the two conditions of (1). Consequently, the geodesic distances will be computed inappropriately. Fig. 4B gives an example where an arm in front of the torso is connected to the upper body and the geodesic distance from the body center to the hand is underestimated. Without correction, anatomical landmarks on the arm cannot be detected.

We therefore propose a disambiguation approach that makes use of movement occurring between frames. Assuming that distinct body parts move separately, this approach allows us to disconnect points belonging to different body parts. We introduce a binary map indicating the location of the entire *occluding* body segment in the depth image. This map is propagated and updated from frame to frame using optical flow, until the body parts become separable again.

3.6.1. Creation of body segment map

Let $\mathbf{p}_m^t \in \mathcal{P}_t'$ be the location of a primary anatomical landmark at time t and let b_m^t denote the corresponding body part, i.e. an arm or a leg. We define the body segment map $\mathbf{S}_t^m$ for b_m^t to be a binary image of the same size as the depth image $\mathbf{D}_t$, such that

$$\mathbf{S}_t^m(i,j) = \begin{cases} 1 & \text{if } d_G\left(\mathbf{p}_m^t, \mathbf{x}_{ij}\right) < \mu, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

All pixel locations (i,j) in the map are assigned a value of 1 if the geodesic distance between their corresponding 3D point $\mathbf{x}_{ij}$ and the landmark $\mathbf{p}_m^t$ does not exceed μ. This threshold is chosen based on the length of the person's limbs (determined during initialization), such that the entire body segment b_m^t is included in the segment map. Fig. 4C shows a body segment map for the person's left arm.

3.6.2. Map propagation using optical flow

For every primary landmark $\mathbf{p}_m^t$ that is not detected using the approach described in Section 3.5, we obtain the corresponding body segment map $\mathbf{S}_{t-1}^m$ from the previous frame. If the landmark was detectable in that frame, we construct the map according to (5), otherwise we assume that the map is available from previous propagation steps. Let $\mathcal{F}_t = (\mathbf{F}_{t,x}, \mathbf{F}_{t,y})$ denote the optical flow between the intensity images $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$. Note that we use the ToF grayscale amplitude

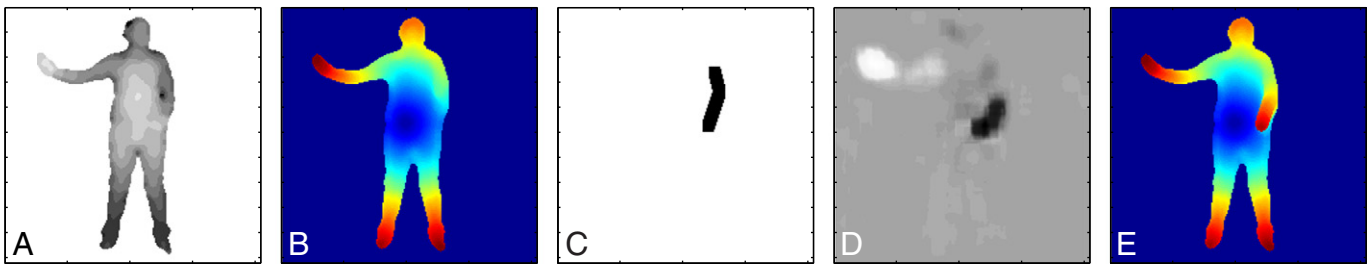

Fig. 4. Illustration of the depth disambiguation approach using optical flow. A: Background-subtracted ToF depth image with a hand in front of the torso. B: Geodesic distance map computed using the graph G_t with the origin at the body center. The occluding arm is too close to the torso for being separated. C: Body segment map for the arm obtained in the previous frame. D: Optical flow field (x-component) from previous to current frame. E: Geodesic distance map after removal of undesired edges in G_t. The arm is now separated and has the expected geodesic distance from the body center.

image or the Kinect RGB image, respectively. $\mathbf{F}_{t,x}(i,j)$ is the x-component of the estimated movement for pixel (i,j) between the two images, and similarly for y. Fig. 4D shows an exemplary flow field. We use the optical flow to update the map $\mathbf{S}^m_{t-1}$ to reflect the assumed position of body part b^t_m in frame t. The propagated map $\mathbf{S}^m_t$ is computed such that

$$\mathbf{S}^m_t\left(i+\mathbf{F}_{t,x}(i,j), j+\mathbf{F}_{t,y}(i,j)\right)=\mathbf{S}^m_{t-1}(i,j). \tag{6}$$

A set of image processing steps are applied to the propagated map, including morphological operations, to remove noise and cavities caused by artifacts in the optical flow field.

3.6.3. Removal of undesired graph edges

Using the updated and corrected map, we can remove the undesired edges in the graph G_t that connect points on body segment b^t_m to the body part in the background, e.g. the torso. We update the set of edges as $E_t = E_t - F$, with

$$F=\left\{\left(\mathbf{x}_{ij},\mathbf{x}_{kl}\right)\in E_t \;\middle|\; \mathbf{S}^m_t(i,j)\neq\mathbf{S}^m_t(k,l)\right\}, \tag{7}$$

where $\mathbf{x}_{ij}$ is the 3D point corresponding to the location (i,j) in the body segment map. In other words, all edges are removed where one point lies within the body segment map and the other point does not. Fig. 4E illustrates the geodesic distances from the body center after the edges between the occluding arm and the torso have been disconnected. The corrected graph allows us to identify the primary and secondary anatomical landmarks on body segment b^t_m by re-computing the geodesic distances from the body center and selecting points with a maximal distance, as described in Section 3.5.

In the situation that multiple primary anatomical landmarks cannot be detected, we repeat the process described above for every missing landmark, each time propagating the appropriate body segment map and disconnecting undesired graph edges. Note that the map propagation step, although based upon optical flow between subsequent frames, does not fail without movement. In such cases, the optical flow field is close to zero and the body segment map simply remains unchanged.

3.7. Skeleton fitting using inverse kinematics

Once the anatomical landmarks $\mathcal{P}_t$ have been identified and labeled in frame t, we estimate the full-body pose parameters $\mathbf{q}_t\in\mathbb{R}^d$ by fitting a skeleton to the detected points. Our skeleton model consists of $K=16$ joints, denoted as $\{\mathbf{s}_i\}_{i=1}^K$, that are distributed over five kinematic chains (both arms and legs, torso). Hinge joints, such as the knees and elbows, are represented with one degree of freedom, approximate ball joints, such as the shoulders and hips, have three degrees of freedom. In total, the parameter space for the skeleton model has $d=38$ dimensions.

Starting with the torso chain that is registered in the body centroid $\mathbf{c}^t$, the full-body pose is determined, intuitively, by attracting selected joints of the kinematic chains (effectors) to the locations of the anatomical landmarks (targets). To match the $L=11$ anatomical landmarks extracted from the depth images, we select as effectors the hands, elbows, feet, knees, the head, the pelvis and one of the spine joints. Formally, our objective is to find the optimal joint angle configuration $\mathbf{q}_t$ such that the residual error

$$\varepsilon(\mathbf{q},t)=\sum_{i=1}^{L}||\mathbf{p}^t_i-f_i(\mathbf{q})||^2_2+c(\mathbf{q}) \tag{8}$$

is minimized. Here, $f_i:\mathbb{R}^d\rightarrow\mathbb{R}^3$ is a forward kinematic function that computes the 3D position of the i-th joint, given a vector $\mathbf{q}$ of joint angles. We assume that the i-th joint is the effector corresponding to the i-th anatomical landmark. The term $c(\mathbf{q})$ penalizes joint angle configurations that violate a set of constraints. This term increases polynomially when any of the joint angles approaches its pre-specified lower and upper limits.

To determine the optimal joint angle configuration $\mathbf{q}_t$ at each time step t, we employ an iterative Gauss–Newton optimization approach that, starting with an initial value $\hat{\mathbf{q}}_0$, computes updates $\Delta_{\mathbf{q}}$ such that $\hat{\mathbf{q}}_{i+1}=\hat{\mathbf{q}}_i+\Delta_{\mathbf{q}}$, until convergence. In each frame, we use the joint angles of the previous frame as an initial value, $\hat{\mathbf{q}}_0=\mathbf{q}_{t-1}$. Assuming incremental body movement between subsequent frames, this increases convergence rates and decreases the probability of hitting local minima. In addition to the joint angles $\mathbf{q}_t$, the skeleton fitting step also gives us the corresponding 3D locations $\{\mathbf{s}_i\}_{i=1}^K$ of all skeleton joints. Note that these locations will be close to the location of the extracted anatomical landmarks, but there will not necessarily be a coincidence. This is a favorable effect, as the impact of outlier landmark detections will be limited.

4. Experiments and results

4.1. Experimental setup

We evaluated our body tracking method using a ToF camera and a Microsoft Kinect device (see Fig. 1). The ToF camera is a PMDVision CamCube with a resolution of 204×204 pixels for both, intensity and depth images. The Kinect captures depth and RGB images at a resolution of 640×480 pixels. We performed the following two sets of experiments:

- ToF data: We compared the skeleton predictions produced by our method to ground truth skeleton data obtained using a marker-based motion capture system. We recorded a series of 20 testing sequences using the ToF-camera. Each of the sequences consists of around 400 frames, at a frequency of 10 Hz. The recorded movements range from simple motions, such as waving an arm, through complex full-body movements with occlusions between body parts. Fig. 7 gives an overview of the movements in our ToF-based evaluation set.
- Kinect data: The skeleton data generated by the NITE skeleton tracker from PrimeSense [28] was used as a reference for assessing the predictions of our method. The Kinect-based data we used for evaluation consists of 10 testing sequences, each approximately of 600 frames in length, recorded at 30 Hz. The movements are similar to those performed in the ToF dataset.

In our experiments, the depth and intensity images were pre-processed using a median filter to decrease the level of noise. The Kinect RGB images were downscaled to match the depth images and aligned to the reference frame of the depth camera by means of stereo calibration. We segmented the person from the background in each frame by subtracting a static depth image of the lab acquired beforehand. We used the Horn–Schunck method for computation of the optical flow fields and low-pass filtered each of the spatial flow field components. Our current Matlab implementation reaches tracking rates of 2–6 frames per second for Kinect and ToF data, respectively. Performance on the Kinect data is lower, given its higher resolution and thus higher complexity in the graph computations.

4.2. Full-body pose estimation on ToF data

For recording ground truth full-body poses, we used an optical marker-based motion capture system based on the ART Dtrack 2 tracking system. The motion capture system was synchronized with the ToF camera and registered to its coordinate frame. Motion capture markers were placed on the back of the person to prevent interference with the depth measurements. The motion capture

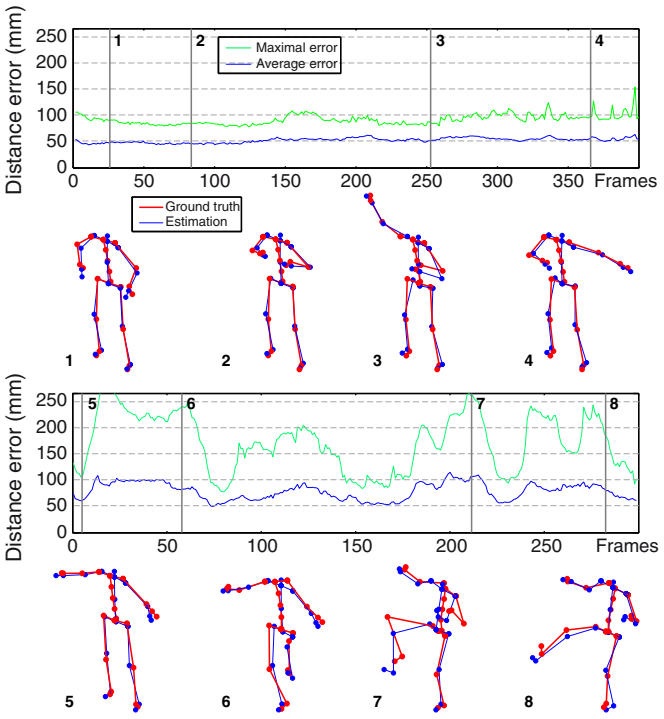

Fig. 5. Illustration of quantitative pose estimation results for ToF depth images. The two graphs show the distance error $e_{\text{tof}}(t)$ over the length of two exemplary testing sequences. The average error over all joints is plotted (blue), along with the maximum error in each frame (green). Left: Typical sequence where only hands are moved, including self-occlusions. Right: Typical sequence involving full-body movement. Results for selected frames are visualized below with overlaid estimated (blue) and ground truth poses (red).

system provides the true 3D positions $\{\hat{\mathbf{s}}_i\}_{i=1}^{K}$ of the $K=16$ body joints of the skeleton model described in Section 3.7.

As an error metric, we computed in every frame the average Euclidean distance between the estimated and true locations of these body joints. We define the distance error for our ToF experiments as

$$e_{\text{tof}}(t) = \frac{1}{K}\sum_{i=1}^{K} \|\mathbf{s}_i - \hat{\mathbf{s}}_i\|_2, \tag{9}$$

where $\hat{\mathbf{s}}_i$ is the ground truth 3D position of the i-th skeleton joint and $\mathbf{s}_i$ is the corresponding estimate. Note that, even in the case of a perfectly estimated full-body pose, $e_{\text{tof}}(t)$ will not be zero, since the markers of the motion capture system do not coincide with our detected anatomical landmarks. Moreover, the motion capture system fits the skeleton to assumed locations of joints within the body, whereas our fitting targets are on the body surface.

Averaged over all testing sequences, our method achieved a distance error of $\bar{e}_{\text{tof}} = 70.1\,\text{mm}$ with a standard deviation of 9.8 mm. Fig. 5 shows plots of the distance error over the length of two typical testing sequences. The overlaid full-body pose prediction and ground truth for selected frames allows for a better interpretability of the results. The left graph in Fig. 5 corresponds to one of the easy sequences where only the arms are moved, however, including body self-occlusions. In this case, the distance error averaged over all joints is around 50 mm. The maximum error in each frame rarely exceeds 100 mm. On the right side, results are shown for a more difficult sequence including full-body movement. Especially when legs are raised, the average error increases to around 100 mm. The effect of the maximal value of 250 mm for individual joints is visualized in example 7 (Fig. 5), where the position of the right knee deviates from the ground truth. This being an example for worst-case deviations, our method compares favorably to current state-of-art methods for ToF-based full-body pose tracking (e.g. [9]).

4.3. Full-body pose estimation on Kinect data

The open-source NITE framework, maintained by PrimeSense [28], provides a skeleton tracking algorithm that we used as a reference for evaluating the performance of our method on Kinect data. For this purpose, we recorded the NITE skeleton predictions for each frame together with the corresponding depth and intensity images. In our ToF-based experiments (Section 4.2), we used the same body model for skeleton fitting and for the motion capture ground truth, leading to a one-to-one correspondence between estimated and ground truth joints. The skeleton model employed in the NITE framework, however, is structured differently and only consists of $\bar{K} = 15$ joints. We denote the locations of these joints as $\{\bar{\mathbf{s}}_i\}_{i=1}^{\bar{K}}$. To measure the deviation of joint locations predicted by our method, we manually created an assignment of matching joints between the two body models. We define the distance error for our Kinect experiments as

$$e_{\text{kinect}}(t) = \frac{1}{\bar{K}}\sum_{i=1}^{\bar{K}} \|\mathbf{s}_i - \bar{\mathbf{s}}_{\kappa(i)}\|_2, \tag{10}$$

where $\kappa(i)$ gives the index of the NITE skeleton joint corresponding to the i-th joint of our model.

Averaged over all Kinect-based testing sequences, our method achieved a distance error $\bar{e}_{\text{kinect}} = 108.4\,\text{mm}$. This value is higher than in the ToF-based experiments as there is no exact correspondence in body joints between the two used body models. We observed that a deviation of approximately 30 mm was present even when the predicted and the NITE skeleton visually were in the

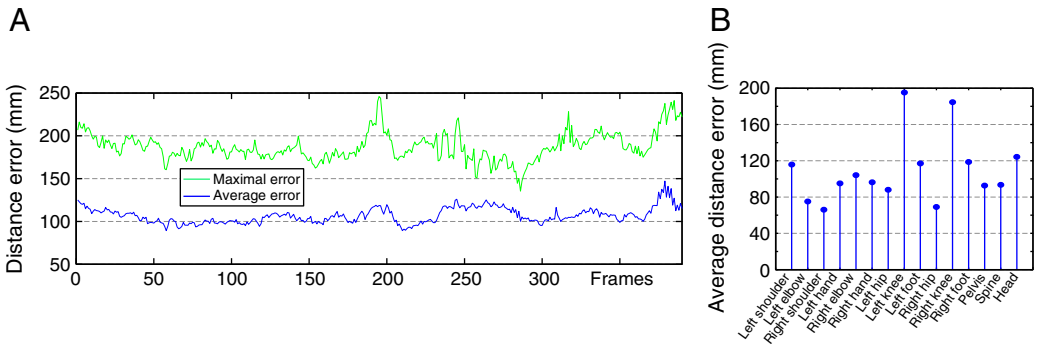

Fig. 6. Quantitative pose estimation results on Kinect depth data. A: Distance error $e_{\text{kinect}}(t)$ over the length of one exemplary testing sequence involving arm and full-body movements. The average error over all joints is plotted (blue), along with the maximum error in each frame (green). B: Distribution of distance error across the 15 joints used for comparison to the NITE skeleton model.

same pose. Fig. 6A shows a plot of the distance error over a typical Kinect-based testing sequence involving full-body movements and hand occlusions. To better understand the difference between the averaged and maximal error, Fig. 6B provides the deviations for each of the $\bar{K} = 15$ joints that were compared. Most notably, the deviations for body parts that are crucial for gesture-based interaction, such as the hands and elbows, are below 100 mm. There is a significantly higher error for both knees that is mainly due to a different location of the knee joint between both skeleton models. As can be seen in Fig. 8, the NITE skeleton tracker tends to underestimate the lower leg length, placing the knee joint lower than in our model.

4.4. Qualitative assessment

Fig. 7 provides example images from our ToF-based testing sequences for a qualitative assessment of our method. As can be

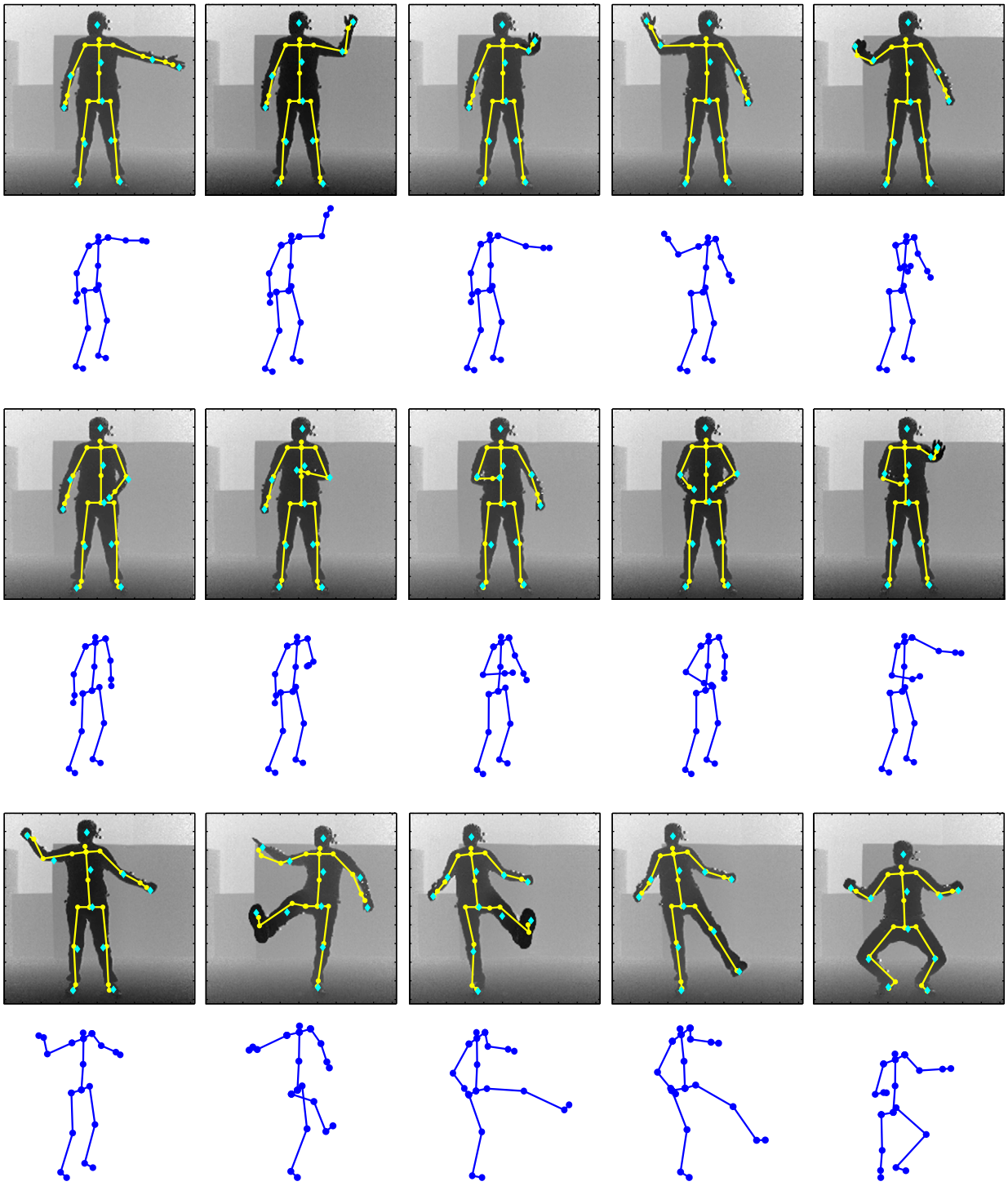

Fig. 7. Qualitative assessment of the proposed full-body pose estimation method on ToF depth images. In each of the three rows, depth images are overlaid with projections of the estimated skeleton pose (yellow). Blue markers indicate the positions of detected anatomical landmarks that play the role of targets for skeleton fitting. Below each row, perspective views of the corresponding estimated poses are displayed, emphasizing the 3D appearance of the predictions.

seen, the estimated full-body poses match with the ToF depth images. Poses are predicted faithfully, even in cases where arms or legs move toward or away from the camera. In particular, the situation where a hand is stretched forward and occludes the arm itself is handled well. The second row of images in Fig. 7 shows cases where one or both hands move in front of the torso. Here, our method relies on the optical flow-based disambiguation approach described in Section 3.6. Tracking does not fail, even when more than one limb moves in front of the body. The speed of movements is not a critical parameter to our technique, as long as the positions of primary anatomical landmarks can be matched successfully across subsequent frames.

Fig. 8 illustrates the pose estimation results on our Kinect-based testing data. Each row shows four depth images overlaid with a projection of the detected anatomical landmarks and the fitted skeleton. Displayed below are 3D views of each pose, where the output of the NITE skeleton tracker is shown for direct comparison. While the poses appear almost identical, it is worth noting that the poses predicted by our approach tend to look more anatomically plausible. The reason here is the underlying human body model with kinematic constraints and fixed, person-specific limb lengths that is fitted to anatomical landmarks. In contrast, the NITE tracker detects interest points (i.e. joints) and connects them to obtain a skeleton. In our experiments, both methods failed to correctly track the person when turning out of the frontal pose toward the image plane for more than 60°. The NITE tracker can cope with the challenging situations of limb self-occlusions that our method solves by means of optical flow. However, in such situations, the NITE skeleton becomes unstable and joint positions start jittering. Our approach faced problems when

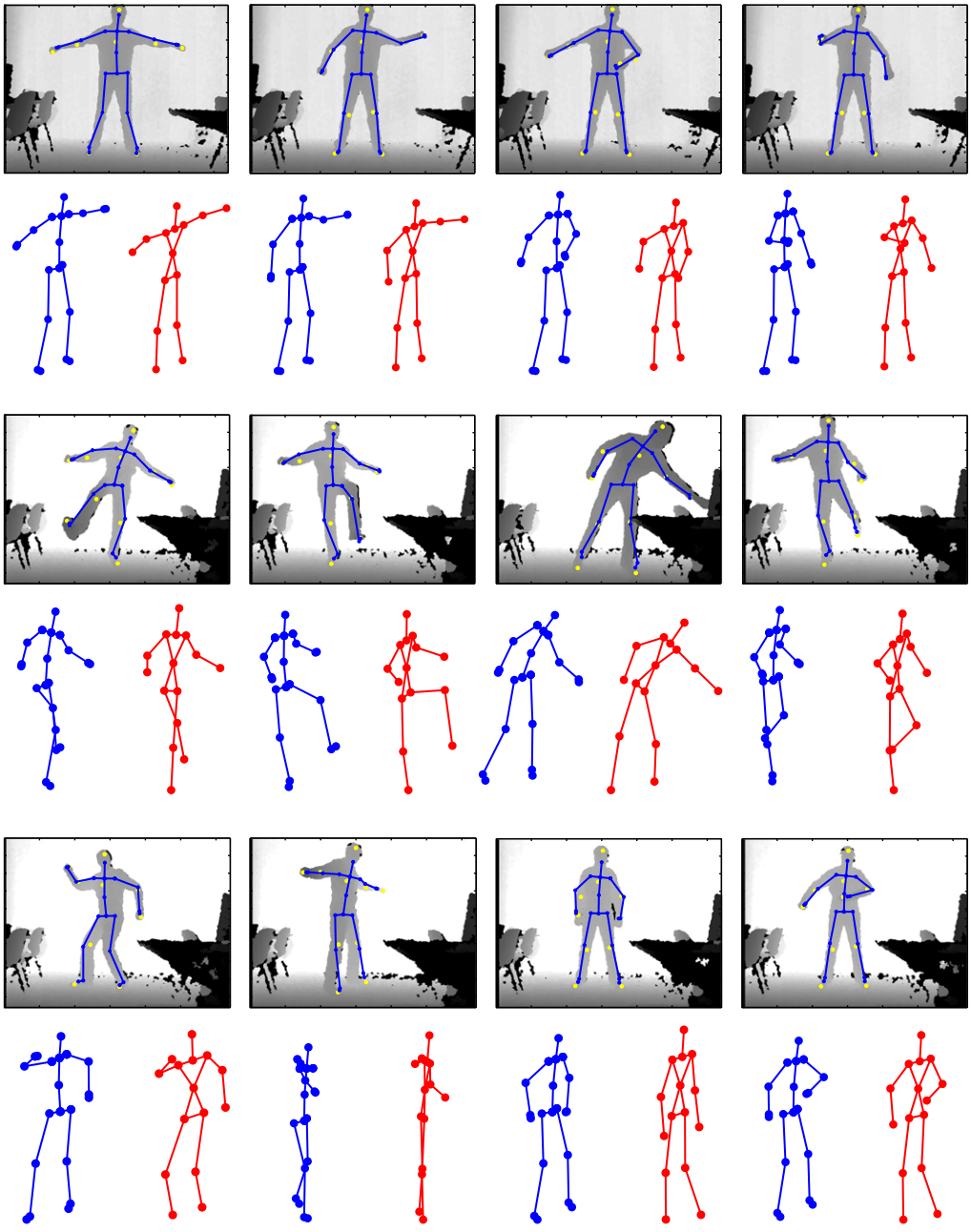

Fig. 8. Qualitative assessment of the proposed full-body pose estimation method on Kinect depth images. Depth images are overlaid with the detected anatomical landmarks (yellow markers) and the fitted skeleton (blue). Below each image is a 3D view of the predicted skeleton (blue) and of the corresponding NITE skeleton prediction (red).

two arms or legs crossed each other in front of the torso. In such cases, parts of one limb were cut off by the occluding limb, resulting in inaccurate landmark detections.

5. Discussion

While our method requires an initialization phase, the user is only required to hold a T-pose for determining the approximate limb lengths and localizing the initial anatomical landmark locations. A similar calibration pose has to be taken by users of the NITE skeleton tracking method. In future, this initialization step could be overcome by adapting automatic body calibration methods, such as [29], to the present setting. An inherent assumption of our method is that persons are facing the camera and do not fully rotate around their vertical axis. We argue that this assumption is reasonable for gesture-based human-machine interaction. We also assume that the person is always visible from head to toe in the depth images. In other cases, the body part detection and propagation steps would fail. However, short phases of occlusions can be handled, as long as body parts re-appear at similar locations as before the occlusion. While the current implementation is close to providing real-time frame rates, we believe there is sufficient potential for improving computational efficiency, without requiring GPU acceleration.

Using ToF and Kinect data in parallel and running our algorithm on both types of depth information has lead to several observations. The higher resolution of Kinect depth images leads to a higher stability in detected landmark locations, as compared to the ToF data. This increase in resolution, however, also negatively affects runtime performance of our algorithm that depends on graph construction and on the Dijkstra algorithm. A positive aspect of the Kinect data is that regions with invalid measurements result in values that are clearly separable from valid depth measurements. Noise in ToF data appears often indistinguishable from regular depth measurements, requiring to make heavier use of filtering. Our future work will focus on improving the labeling process for detected anatomical landmarks that currently depends on matching to the locations of anatomical landmarks in previous frames.

6. Conclusion

We have presented a method for tracking human full-body pose from sequences of ToF camera images. The approach does not require any training data and is able to track arbitrary movements for gesture-based human-computer interaction. Our method takes full advantage of the data provided by typical depth cameras by utilizing both, depth and intensity information. Based on the depth data, we segment the person in front of static background and construct a graph-based representation of the 3D points. This graph allows us to robustly identify anatomical landmarks in each frame by selecting points with a maximal geodesic distance from the body center of mass. In cases where body parts occlude each other, we rely on optical flow, computed on the intensity images, to disconnect occluding limbs from the body part behind. The experimental evaluation on ToF and Kinect data shows that our method can track various full-body movements, including self-occlusions, and estimate 3D full-body poses with a high accuracy.

References

[1] R. Johnson, K. O'Hara, A. Sellen, C. Cousins, A. Criminisi, Exploring the potential for touchless interaction in image-guided interventional radiology, ACM Conference on Computer-Human Interaction (CHI), 2011.
[2] R. Urtasun, T. Darrell, Sparse probabilistic regression for activity-independent human pose inference, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[3] T. Jaeggli, E. Koller-Meier, L.V. Gool, Learning generative models for multi-activity body pose estimation, Int. J. Comput. Vis. 83 (2) (2009) 121–134.
[4] R. Kehl, L. Gool, Markerless tracking of complex human motions from multiple views, Comput. Vis. Image Underst. 104 (2) (2006) 190–209.
[5] J. Bandouch, F. Engstler, M. Beetz, Accurate human motion capture using an ergonomics-based anthropometric human model, Articulated Motion and Deformable Objects (AMDO), 2008.
[6] R. Hu, A. Hartfiel, J. Tung, A. Fakih, J. Hoey, P. Poupart, 3D pose tracking of walker users' lower limb with a structured-light camera on a moving platform, Computer Vision and Pattern Recognition Workshops, 2011.
[7] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[8] A. Kolb, E. Barth, R. Koch, R. Larsen, Time-of-flight sensors in computer graphics, EUROGRAPHICS, 2009, pp. 119–134.
[9] V. Ganapathi, C. Plagemann, D. Koller, S. Thrun, Real time motion capture using a single time-of-flight camera, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.
[10] L.A. Schwarz, D. Mateus, V. Castaneda, N. Navab, Manifold learning for tof-based human body tracking and activity recognition, British Machine Vision Conference (BMVC), 2010.
[11] C. Plagemann, V. Ganapathi, D. Koller, Real-time identification and localization of body parts from depth images, IEEE International Conference on Robotics and Automation (ICRA), 2010.
[12] M. Mortara, G. Patane, M. Spagnuolo, From geometric to semantic human body models, Comput. Graph. 30 (2006) 185–196.
[13] Y. Sun, M. Bray, A. Thayananthan, B. Yuan, P. Torr, Regression-based human motion capture from voxel data, British Machine Vision Conference (BMVC), 2006.
[14] A. Fossati, M. Salzmann, P. Fua, Observable subspaces for 3D human motion recovery, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[15] Y. Zhu, B. Dariush, K. Fujimura, Controlled human pose estimation from depth image streams, Computer Vision and Pattern Recognition Workshops, 2008.
[16] G. Pons-Moll, A. Baak, T. Helten, M. Müller, H.-P. Seidel, B. Rosenhahn, Multisensor-fusion for 3D full-body human motion capture, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.
[17] V. John, E. Trucco, S. Ivekovic, Markerless human articulated tracking using hierarchical particle swarm optimisation, Image Vis. Comput. 28 (11) (2010) 1530–1547.
[18] H. Ning, T. Tan, L. Wang, W. Hu, Kinematics-based tracking of human walking in monocular video sequences, Image Vis. Comput. 22 (5) (2004) 429–441.
[19] A. Bleiweiss, E. Kutliroff, Markerless motion capture using a single depth sensor, ACM SIGGRAPH ASIA Sketches, 2009.
[20] S. Soutschek, J. Penne, J. Hornegger, J. Kornhuber, 3-D gesture-based scene navigation in medical imaging applications using time-of-flight cameras, Computer Vision and Pattern Recognition Workshops, 2008.
[21] R. Jensen, R. Paulsen, R. Larsen, Analyzing gait using a time-of-flight camera, Scandinavian Conference on Image Analysis, 2009, pp. 21–30.
[22] Y. Zhu, K. Fujimura, A bayesian framework for human body pose tracking from depth image sequences, Sensors 10 (5) (2010) 5280–5293.
[23] S. Malassiotis, M. Strintzis, Real-time hand posture recognition using range data, Image Vis. Comput. 26 (7) (2008) 1027–1037.
[24] M.B. Holte, T.B. Moeslund, P. Fihl, Fusion of range and intensity information for view invariant gesture recognition, Computer Vision and Pattern Recognition Workshops, 2008.
[25] S. Denman, V. Chandran, S. Sridharan, An adaptive optical flow technique for person tracking systems, Pattern Recognit. Lett. 28 (10) (2007) 1232–1239.
[26] R. Okada, Y. Shirai, J. Miura, Tracking a person with 3-D motion by integrating optical flow and depth, IEEE International Conference on Automatic Face and Gesture Recognition (FG), 2000.
[27] L.A. Schwarz, A. Mkhitaryan, D. Mateus, N. Navab, Estimating human 3D pose from time-of-flight images based on geodesic distances and optical flow, IEEE Conference on Automatic Face and Gesture Recognition (FG), 2011.
[28] PrimeSense, Inc., NITE Middleware, http://www.primesense.com/Nite/2010.
[29] J.F. Obrien, B. Bodenheimer, G. Brostow, J. Hodgins, Automatic joint parameter estimation from magnetic motion capture data, Graph. Interface (2000), pp. 53–60.