Self-Organizing Maps for Pose Estimation with a Time-of-Flight Camera

Martin Haker, Martin Böhme, Thomas Martinetz, and Erhardt Barth

Institute for Neuro- and Bioinformatics, University of Lübeck
Ratzeburger Allee 160, 23538 Lübeck, Germany
{haker,boehme,martinetz,barth}@inb.uni-luebeck.de
http://www.inb.uni-luebeck.de

Abstract. We describe a technique for estimating human pose from an image sequence captured by a time-of-flight camera. The pose estimation is derived from a simple model of the human body that we fit to the data in 3D space. The model is represented by a graph consisting of 44 vertices for the upper torso, head, and arms. The anatomy of these body parts is encoded by the edges, i.e. an arm is represented by a chain of pairwise connected vertices whereas the torso consists of a 2-dimensional grid. The model can easily be extended to the representation of legs by adding further chains of pairwise connected vertices to the lower torso. The model is fit to the data in 3D space by employing an iterative update rule common to self-organizing maps. Despite the simplicity of the model, it captures the human pose robustly and can thus be used for tracking the major body parts, such as arms, hands, and head. The accuracy of the tracking is around 5–6 cm root mean square (RMS) for the head and shoulders and around 2 cm RMS for the head. The implementation of the procedure is straightforward and real-time capable.

1 Introduction

A time-of-flight (TOF) camera [1] provides a range map that is perfectly registered with an intensity image (often referred to as an *amplitude* image in TOF nomenclature), making it an attractive sensor for a wide range of applications.

In this paper, we present a technique for estimating human pose in 3D based on a simple model of the human body. The model consists of a number of vertices that are connected by edges such that the resulting graph structure resembles the anatomy of the human body, i.e. the model represents the torso, the head, and the arms. The model is updated using an iterative learning rule common to self-organizing maps (SOMs) [2]. The position of certain body parts, such as the hands, can be obtained from the model as the 3D coordinates of the corresponding vertices, i.e. the position of the hands in 3D corresponds to the position of the vertex that terminates the chain representing an arm. Thus, body parts can be tracked in 3D space.

The estimation of 3D human pose has been addressed in a number of different publications. The majority of work focuses on the estimation of pose from

R. Koch and A. Kolb (Eds.): Dyn3D 2009, LNCS 5742, pp. 142–153, 2009.
© Springer-Verlag Berlin Heidelberg 2009

single images taken with a regular 2D camera, and a number of different algorithmic approaches have been presented. In [3] the pose is recovered from shape descriptors of image silhouettes. The authors of [4] map low-level visual features of the segmented body shape to a number of body configurations and identify the pose as the one corresponding to the most likely body configuration given the visual features. An approach based on a large database of example images is presented in [5]. The authors learn a set of parameter-sensitive hashing functions to retrieve the best match from the database in an efficient way.

Very accurate 3D reconstruction of human motion from multi-view video sequences was published in [6]. Based on a segmentation of the subject, the authors use a multi-layer framework that combines stochastic optimization, filtering, and local optimization to estimate the pose using a detailed model of the human body. However, the computational cost is relatively high and the system does not operate at camera frame rates.

Pose estimation based on 3D data has been addressed in [7]. The 3D volume of a person is estimated in a multi-camera setup using the shape-from-silhouette method. A skeleton model is then fit to a 2D projection of the volumetric data. The 2D projection is obtained by a virtual camera and the model is fit using certain features of the outer contour. The 3D coordinates of the model are finally reconstructed by inverting the 2D projection of the virtual camera, i.e. the vertices of the skeleton are projected back into 3D space using the intrinsic parameters of the virtual camera.

Another approach to obtaining a skeleton in 3D is to apply a thinning to volumetric data directly in 3D space [8,9]. The human pose can then be estimated from the skeleton [10].

Two related methods based on stereo imaging were presented in [11] and [12]. The authors introduce a hierarchical human body model database. For a given image the algorithm uses both silhouette and depth information to identify the model pose with the best match.

The work in [13] fuses 2D and 3D information obtained from a stereo rig and a TOF camera to fit a human body model composed of generalized cylinders. The system models body joints and uses kinematic constraints to reduce the degrees of freedom. The 3D data is obtained using a TOF camera and the system runs at frame rates of 10–14 frames per second.

Another recent approach using TOF cameras was presented in [14]. The method tracks a number of anatomical landmarks in 3D over time and uses these to estimate the pose of an articulated human model. The model is in turn used to resolve disambiguities of the landmark detector and to provide estimates for undetected landmarks. The entire approach is very detailed and models constraints such as joint limit avoidance and self-penetration avoidance. Despite its complexity, the method runs at a frame rate of approximately 10 frames per second.

Our approach, in contrast, is a very simple one that demonstrates how effectively TOF cameras can be used to solve relatively complex computer vision tasks. A general advantage of TOF cameras is that they can provide both range and

intensity images at high frame rates. The combined use of both types of data was already used for tracking [15,16] and allows a robust segmentation of the human body in front of the camera. The range data, representing a $2\frac{1}{2}$D image, can then be used to obtain a point cloud in 3D representing the visible surface of the person. Thus, limbs extended towards the camera can still be easily identified while this proves to be a more difficult task in 2D projections of a scene.

Our approach takes advantage of this property and fits a simple model of the human body into the resulting point cloud in 3D. The model fitting algorithm is based on a SOM, can be implemented in a few lines of code, and the method runs at frame rates up to 25 frames per second on a 2 GHz Intel Core 2 Duo. The algorithmic approach to this procedure is discussed in Sect. 2. The method delivers a robust estimation of the human pose, as we show in Sect. 3 for image data that was acquired using a MESA SR4000 TOF camera.

2 Method

The first step of the proposed procedure is to segment the human body from the background of the image. We employ a simple thresholding approach that uses both range and intensity data. The thresholds for the two images are determined adaptively for each frame.

In case of the amplitude image, the pixel values correspond to the amount of light of the TOF camera's active illumination that is reflected back into the camera. Hence, the amplitude can be considered a confidence measure for the accuracy of the range measurement because it indicates the measurement's signal-to-noise ratio. The attenuation of the amplitude is proportional to the squared distance of an object to the camera. Thus, objects close to the camera appear generally much brighter than the background. We use the Otsu threshold [17] to determine an adaptive value for the threshold that separates the dark background from the brighter foreground. A more accurate segmentation using thresholding on amplitude data proves to be difficult because the objects may have different properties of reflecting infrared light.

In case of the range data, this simple assumption of a bimodal distribution does not hold if multiple objects are located at different distances in front of the camera. Thus, we construct a histogram of the range values in which every object can be assumed to result in a peak if the objects are truly at different distances from the camera. The threshold is determined as the one that separates the peak corresponding to the closest object from the peaks of the remaining objects.

The final segmented image is obtained as the one where the foreground pixels have been classified as foreground pixels with respect to both types of data. Furthermore, we identify the largest connected component of foreground pixels and consider all remaining pixels background. Thus, we obtain a clear segmentation of a single person closest to the camera in most cases. A sample TOF image and the resulting segmented image is shown in Fig. 1.

The identified foreground pixels can be assumed to sample the visible surface of the person in front of the camera. Since the intrinsic parameters of the

Fig. 1. Sample image taken with a MESA SR4000 TOF camera. The leftmost image shows the amplitude data. The range image is given in the center and the resulting segmentation is shown on the right.

camera, such as focal length and pixel size, are known, the surface pixels can be projected back into 3D space. As a result one obtains a point cloud in 3D that represents the 3-dimensional appearance of the person. This approach has two major advantages: (i) The representation is scale-invariant due to the fact that the size of the person in 3D space remains the same independently of the size of its image; (ii) body parts that are extended towards the camera in front of the torso can be easily identified due to the variation in distance, whereas this information is lost in 2D projections of the scene obtained with regular cameras.

Our method aims at fitting a simple graph model representing the anatomy of the human body into the resulting point cloud in 3D. To this end, we employ a SOM. We define a graph structure of vertices and edges that resembles a frontal view of the human body. Body parts, such as arms and torso, are modeled by explicitly defining the neighborhood structure of the graph, i.e. an arm is represented by a simple chain of pairwise connected vertices whereas vertices in the torso are connected to up to four neighbors forming a 2D grid. The resulting model structure is depicted in Fig. 2.

The SOM is updated by an iterative learning rule for each consecutive frame of the video sequence. The first frame uses the body posture depicted in Fig. 2 as an initialization of the model. During initialization the model is translated to the center of gravity of the 3D point cloud. The scale of the model is currently set manually to a fixed value that corresponds to an average-sized person. We can report that the scale is not a particularly critical parameter and that the same fixed scale works for adults of different height. Once the scale is set to an appropriate value, there is no need to adjust it during run-time due to the above mentioned scale-invariance of the method. The update of the model for each consecutive frame then depends on the model that was estimated for the previous frame.

The adaptation of the model to a new frame involves a complete training of the SOM, i.e. a pattern-by-pattern learning is performed using the data points of the 3D point cloud. This iterative procedure selects a sample vector $\boldsymbol{x}$ from the point cloud at random and updates the model according to the following learning rule:

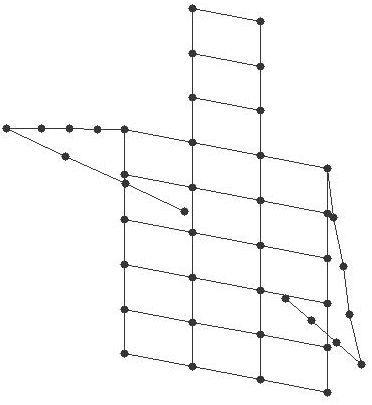

Fig. 2. Graph model of the human body. The edges define the neighborhood structure for the SOM.

$$\hat{\boldsymbol{v}}^{t+1} = \hat{\boldsymbol{v}}^t + \hat{\epsilon}^t \cdot (\boldsymbol{x} - \hat{\boldsymbol{v}}^t) \tag{1}$$

$$\tilde{\boldsymbol{v}}^{t+1} = \tilde{\boldsymbol{v}}^t + \tilde{\epsilon}^t \cdot (\boldsymbol{x} - \tilde{\boldsymbol{v}}^t). \tag{2}$$

Here, $\hat{\boldsymbol{v}}$ denotes the node that is closest to the sample $\boldsymbol{x}$ with respect to the distance measure $d(\boldsymbol{x}, \boldsymbol{v}) = \|\boldsymbol{x} - \boldsymbol{v}\|_2$. The nodes $\tilde{\boldsymbol{v}}$ are the neighbors of $\hat{\boldsymbol{v}}$ as defined by the model structure. The learning rates are denoted by $\hat{\epsilon}^t$ and $\tilde{\epsilon}^t$ for the closest node and its neighbors, respectively. The learning rate $\hat{\epsilon}^t$ was set to:

$$\hat{\epsilon}^t = \epsilon_i \cdot (\epsilon_f / \epsilon_i)^{t/t_{\max}}. \tag{3}$$

Here, $t \in \{0, \ldots, t_{\max}\}$ denotes the current adaptation step for this frame and $t_{\max}$ denotes the total number of adaptation steps performed for this frame. The initial learning rate ϵ_i and the final learning rate ϵ_f were set to 0.1 and 0.05. The learning rate for the neighbors was chosen to be $\tilde{\epsilon}^t = \hat{\epsilon}^t/2$. This choice of the learning rate was already proposed in previous work on self-organizing networks [18]. The initial and final learning rates were set to relatively high values in order to allow the network to handle fast movements of the person, i.e. if the limbs are moved quickly the correctional updates for the corresponding nodes have to be large so that the model can accurately follow.

This update rule does not always guarantee that the topology of the model is preserved. Here, we refer to topology with respect to the connectivity of the nodes within body parts such as the arm. Imagine the situation where the subject's hands touch in front of the torso. If the hands are separated again, it is possible that the model uses the last node of the left arm to represent samples that actually belong to the hand of the right arm. It can thus happen, that the last node of the left arm may continue to be attracted by the right hand although both hands have moved apart and, thus, the left arm will extend into empty space. In principle, the update rules resolve this problem over time. However, only a small number of updates are performed per frame and this may lead to a wrong estimation of the topology for a small number of frames.

To avoid this, we developed a modification of the update rule that speeds up the learning process by forcing neighboring vertices to stay close together. This is achieved by the following rule that is applied after the actual learning step if the distance $d(\hat{\boldsymbol{v}}, \tilde{\boldsymbol{v}}_a)$ exceeds a certain threshold θ:

$$\hat{\boldsymbol{v}} = \tilde{\boldsymbol{v}}_a + \theta \cdot \frac{(\hat{\boldsymbol{v}} - \tilde{\boldsymbol{v}}_a)}{\|\hat{\boldsymbol{v}} - \tilde{\boldsymbol{v}}_a\|_2}. \tag{4}$$

Here, $\tilde{\boldsymbol{v}}_a$ is a specific neighbor of $\hat{\boldsymbol{v}}$ referred to as an anchor. The rule enforces that the distance between the vertex $\hat{\boldsymbol{v}}$ and its anchor is always less than or equal to θ. The threshold θ depends on the scale of the model. The anchor of each vertex is defined as the neighbor that has minimal distance to the center of the torso with respect to the graph structure of the model, i.e. it is the vertex that is connected to the center of the torso by the smallest number of edges.

3 Results

3.1 Qualitative Evaluation

The proposed method was evaluated using a MESA SR4000 TOF camera. We operate the camera at a modulation frequency of 30 MHz for the active illumination. As a result the camera can disambiguate distances in the range of up to 5 meters. In the following sample images, the person was at a distance of roughly 2.5 meters from the camera. At that distance the range measurement has an accuracy of approximately 1 cm.

A sample result of the pose estimation is shown in Fig. 3. The figure depicts the point cloud of samples in 3D that represent the visual surface of the person in front of the camera shown in Fig. 1. The model that was fitted to the point cloud is imprinted into the data. One can observe that the model captures the anatomy of the person correctly, i.e. the torso is well covered by the 2-dimensional grid, a number of vertices extend into the head, and the 1-dimensional chains of vertices follow the arms. Thus, the position of the major body parts, such as the hands, can be taken directly from the corresponding vertices of the model in 3D.

The data from Fig. 3 is taken from a sequence of images. Further sample images from this sequence are given in Fig. 4. Each image shows the segmented amplitude image with the imprinted 2D projection of model. One can observe that the model follows the movement of the arms accurately, even in difficult situations where the arms cross closely in front of the torso. Note that the procedure does not lose the position of the head even though it is occluded to a large extent in some of the frames. The sample images are taken from a video sequence, which is available under http://www.artts.eu/demonstrations/

It is important to point out that the method may misinterpret the pose. This can for example be the case if the arms come too close to the torso. In such a case the SOM cannot distinguish between points of the arm and the torso within the 3D point cloud. We can report, however, that the method can recover the true configuration within a few frames once the arms are extended again in most cases.

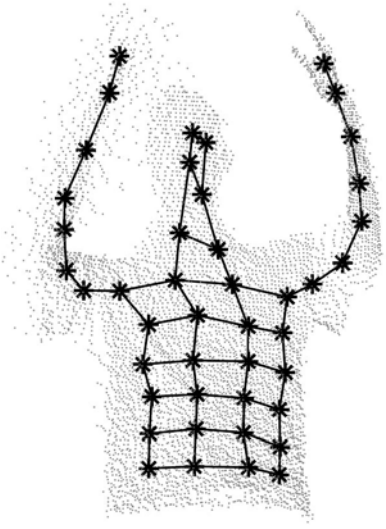

Fig. 3. Point cloud sampling the visible surface of a human upper torso in 3D. The graph represents the human model that was fitted to the data.

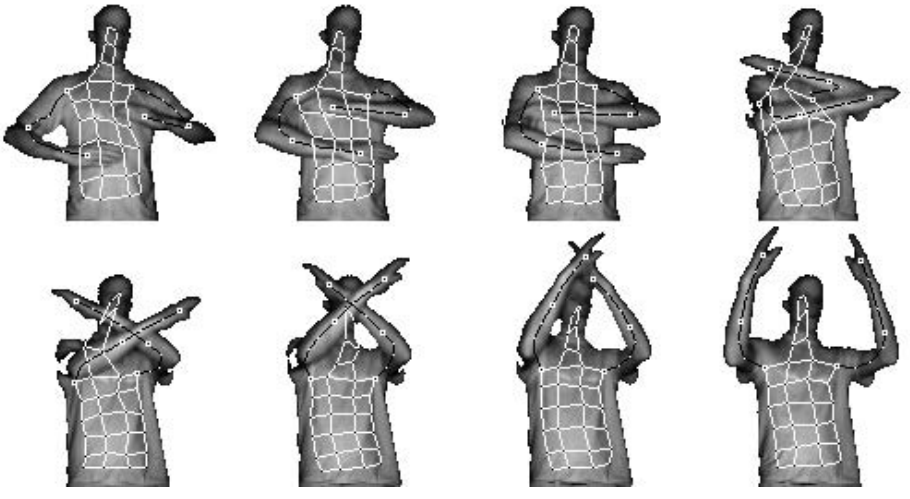

Fig. 4. A selection of frames from a video sequence showing a gesture. The model estimated by the pose estimation is imprinted in each frame. The edges belonging to torso and head are colored in white, whereas the arms are colored in black.

We assume that it is possible to detect and avoid such problems by imposing a number of constraints on the model, e.g. that the arms may only bend at the elbows and that the entire model should generally be oriented such that the head is pointing upwards. However, note that the current results were achieved without any such constraints.

3.2 Quantitative Evaluation

To evaluate the accuracy of the tracking quantitatively, we acquired sequences of 5 persons moving in front of the camera; each sequence was around 140 to 200 frames long. In each frame, we hand-labeled the positions of five parts of the body: the head, the shoulders, and the hands. To obtain three-dimensional ground-truth data, we looked up the distance of each labeled point in the range map and used this to compute the position of the point in space. This implies that the sequences could not contain poses where any of the body parts were occluded; however, many poses that are challenging to track, such as crossing the arms in front of the body, were still possible, and we included such poses in the sequences. (Note that the tracker itself can track poses where, for example, the head is occluded; see Fig. 4.)

The labeled positions can now be compared with the positions of the corresponding nodes in the tracked model. However, when assessing the accuracy of the tracking in this way, we run into the problem that we never define explicitly which part of the body each node should track. For example, though the last node in each of the arms will typically be located on or near the hand, we do not know in advance exactly which part of the hand the node will track. This means that there may be a systematic offset between the position that is labeled as "hand" and the position that the hand node tracks. To give a realistic impression of tracking accuracy, we should eliminate these systematic offsets.

We do this by measuring the average offset between the tracked position and the labeled position on ten "training" frames; this offset is then used to correct the tracked position in the remaining "test" frames, on which the accuracy is measured. Because the orientation of the respective parts of the body can change, we need to measure the offsets not in the world coordinate system but in a local coordinate system. For the head and shoulders, we use a coordinate system where the x-axis points from the left shoulder (of the tracked model) to the right shoulder, the y-axis is defined so that the head lies in the x-y-plane, and the z-axis is perpendicular to the other two axes to form a right-handed coordinate system. For the hands, it is not as easy to define a full coordinate system because the model only measures the direction in which the forearm is pointing but not the orientation of the hand. For this reason, we estimate and correct the offset between tracked and labeled position only along the direction of the forearm, which we define by the last two nodes in the arm; this is the direction that accounts for most of the offset. Any offset perpendicular to the direction of the forearm is not corrected.

Once the tracked positions have been corrected in this way, we can measure the tracking error. Fig. 5 shows a plot of tracking error over time for one of

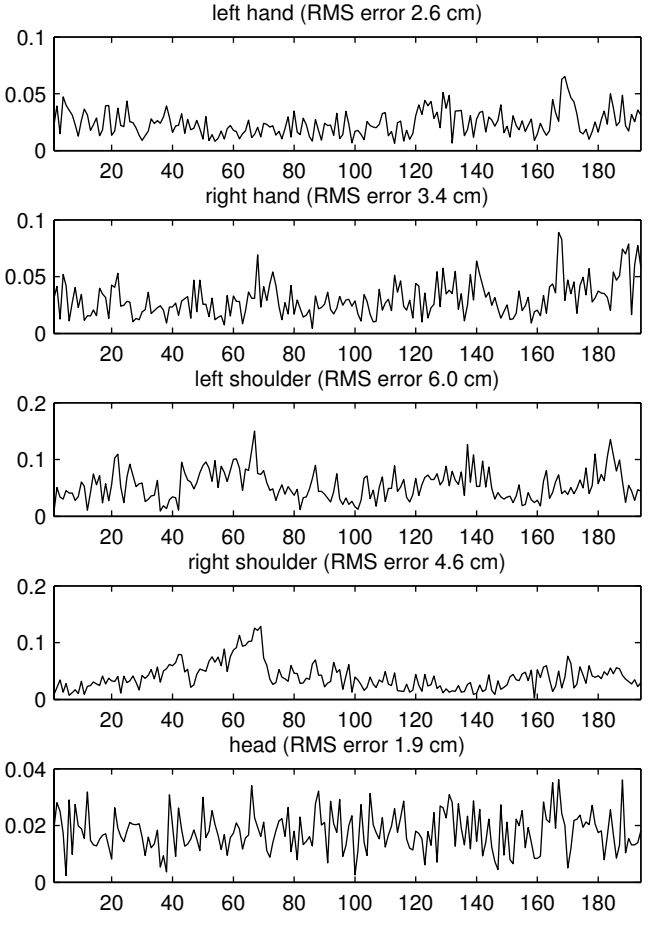

Fig. 5. Plots of tracking error over time for one of the sequences. The horizontal axis plots frame number, the vertical axis plots tracking error in meters.

the recorded sequences. It is obvious that there is little to no systematic error remaining; instead, most of the error is due to tracking noise.

Table 1 shows the root mean square (RMS) tracking error, averaged over all frames and subjects. The average error is around 5 to 6 cm for the hands and shoulders and around 2 cm for the head. While this degree of accuracy is not sufficient for tracking very fine movements, it is more than adequate for determining overall body posture and for recognizing macroscopic gestures. Also, consider that no smoothing of the tracked positions over time was carried out.

A major advantage of the proposed method is that the training of the model converges very fast for each new frame. Thus, only a small number of the samples of the 3D cloud need actually be considered during the update even when the person performs very fast movements in front of the camera. The sample image

Table 1. Root mean square (RMS) error between tracked and labeled positions, averaged over all frames and subjects

body part	RMS error
left hand	5.90 cm
right hand	5.29 cm
left shoulder	5.32 cm
right shoulder	5.15 cm
head	2.21 cm

in Fig. 1 contains roughly 6500 foreground pixels. However, we use only 10% of these samples for updating the model, i.e. we select roughly 650 points in 3D in random order from the point cloud and use these for updating the model by pattern-by-pattern learning. As a result the computational complexity is very low, and we achieve frame rates up to 25 frames per second on a 2 GHz PC while robustly tracking the human pose in scenarios such as the one depicted in Fig. 4. The use of a higher number of samples for training will further increase the robustness while at the same time the frame rate will decrease.

4 Discussion

We have presented a simple procedure to estimate human pose from a sequence of range images. The procedure is especially suitable for TOF cameras as they can deliver range data in combination with intensity images at high frame rates. These cameras can be assumed to be available at relatively low costs in the near future.

The use of a SOM results in a very simple, yet very efficient implementation. In principle the procedure can be extended easily to any other kind of deformable object.

A major shortcoming of the current implementation is that the method cannot deal with multiple persons in front of the camera, i.e. the system always assumes that the segmented foreground pixels correspond to a single person. This approach fails for example if two people are at the same distance in front of the camera and very close together. In that case the segmented foreground pixels sample the visual surface of both persons. Since the SOM attempts to represent all samples equally the resulting pose estimation fails. Using the current approach, this problem must be solved by an improved method for segmentation that can handle multiple objects in front of the camera. Then, a SOM can be trained for each segmented object and thus multiple people can be tracked.

This in turn can lead to a related problem that occurs when the segmentation fails to detect parts of the body due to occlusion, e.g. when the lower part of an arm is occluded by a second person. In that case the SOM will use the entire chain of arm nodes to represent the upper part of the arm. Thus, the node for the hand will be misplaced. To tackle this problem the system needs to identify the presence of certain body parts based on pose estimates from previous

frames. In case occluded body parts have been identified, the corresponding nodes of the SOM must be excluded from training. Instead their location could be predicted based on the posture of the remaining model. These two issues need to be addressed in future work.

There exist other approaches that compute a more accurate estimate of human pose but our goal within this work was to develop a simple method that gives a rough but robust estimate of human pose at high frame rates.

We intend to use the proposed method of pose estimation for action recognition and gesture-based man-maschine interaction. Generally, the evaluation of certain spatio-temporal features for the analysis of video sequences is computationally expensive. We argue that rough knowledge of the position of landmarks, such as the hands, can greatly improve the runtime of feature-based action recognition systems, because the features do not have to be evaluated over the entire video sequence but only at those locations where certain important landmarks have been detected. Furthermore, these features can be put into a larger context if their relative location to each other is known.

Acknowledgments

This work was developed within the ARTTS project (www.artts.eu), which is funded by the European Commission (contract no. IST-34107) within the Information Society Technologies (IST) priority of the 6th Framework Programme. This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

References

1. Oggier, T., Büttgen, B., Lustenberger, F., Becker, G., Rüegg, B., Hodac, A.: SwissRanger^{TM} SR3000 and first experiences based on miniaturized 3D-TOF cameras. In: Ingensand, K. (ed.) Proc. 1^{st} Range Imaging Research Day, Zurich, pp. 97–108 (2005)
2. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1995)
3. Agarwal, A., Triggs, B.: Recovering 3D human pose from monocular images. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(1), 44–58 (2006)
4. Rosales, R., Sclaroff, S.: Inferring body pose without tracking body parts. In: Proceedings of Computer Vision and Pattern Recognition, pp. 721–727 (2000)
5. Shakhnarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: Proceedings of International Conference on Computer Vision, pp. 750–757 (2003)
6. Gall, J., Rosenhahn, B., Brox, T., Seidel, H.P.: Optimization and filtering for human motion capture. International Journal of Computer Vision (2008)
7. Weik, S., Liedtke, C.E.: Hierarchical 3D pose estimation for articulated human body models from a sequence of volume data. In: Klette, R., Peleg, S., Sommer, G. (eds.) RobVis 2001. LNCS, vol. 1998, pp. 27–34. Springer, Heidelberg (2001)
8. Palágyi, K., Kuba, A.: A parallel 3D 12-subiteration thinning algorithm. Graphical Models and Image Processing 61(4), 199–221 (1999)

9. Pudney, C.: Distance-ordered homotopic thinning: A skeletonization algorithm for 3D digital images. In: Computer Vision and Image Understanding, vol. 72, pp. 404–413 (1998)
10. Arata, M., Kazuhiko, S., Takashi, M.: Human pose estimation from 3D object skeleton using articulated cylindrical human model. IPSJ SIG Technical Reports 51, 133–144 (2006)
11. Yang, H.D., Lee, S.W.: Reconstructing 3D human body pose from stereo image sequences using hierarchical human body model learning. In: ICPR 2006: Proceedings of the 18th International Conference on Pattern Recognition, Washington, DC, USA, pp. 1004–1007. IEEE Computer Society, Los Alamitos (2006)
12. Yang, H.D., Lee, S.W.: Reconstruction of 3D human body pose from stereo image sequences based on top-down learning. Pattern Recognition 40(11), 3120–3131 (2007)
13. Knoop, S., Vacek, S., Dillmann, R.: Fusion of 2D and 3D sensor data for articulated body tracking. Robotics and Autonomous Systems 57(3), 321–329 (2009)
14. Zhu, Y., Dariush, B., Fujimura, K.: Controlled human pose estimation from depth image streams. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPRW 2008, June 2008, pp. 1–8 (2008)
15. Böhme, M., Haker, M., Martinetz, T., Barth, E.: A facial feature tracker for human-computer interaction based on 3D TOF cameras. In: Dynamic 3D Imaging – Workshop in Conjunction with DAGM (2007) (in print)
16. Haker, M., Böhme, M., Martinetz, T., Barth, E.: Deictic gestures with a time-of-flight camera. In: Gesture in Embodied Communication and Human-Computer Interaction – International Gesture Workshop GW 2009 (2009)
17. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics 9(1), 62–66 (1979)
18. Martinetz, T., Schulten, K.: A "Neural-Gas" Network Learns Topologies. Artificial Neural Networks I, 397–402 (1991)