Skeletal Graph Based Human Pose Estimation in Real-Time

Matthias Straka
straka@icg.tugraz.at
Stefan Hauswiesner
hauswiesner@icg.tugraz.at
Matthias Rüther
ruether@icg.tugraz.at
Horst Bischof
bischof@icg.tugraz.at

Inst. for Computer Vision and Graphics
Graz University of Technology
Inffeldgasse 16/II, 8010 Graz
Austria

http://www.icg.tugraz.at/

Abstract

We propose a new method to quickly and robustly estimate the 3D pose of the human skeleton from volumetric body scans without the need for visual markers. The core principle of our algorithm is to apply a fast center-line extraction to 3D voxel data and robustly fit a skeleton model to the resulting graph. Our algorithm allows for automatic, single-frame initialization and tracking of the human pose while being fast enough for real-time applications at up to 30 frames per second. We provide an extensive qualitative and quantitative evaluation of our method on real and synthetic datasets which demonstrates the stability of our algorithm even when applied to long motion sequences.

1 Introduction

Human pose estimation is a vivid topic in current literature [17] due to its wide-spread applications such as motion-capture, telepresence or object manipulation in virtual environments. The process of human pose estimation is concerned with finding the pose parameters of a human body model that best fit to the observations in one or more input images. While marker-based systems are already available in numerous commercial applications, marker-less pose estimation is still a challenging research topic. There exists a variety of algorithms that solve this task with high accuracy from multiple input images [6, 12] or even a single photograph [8, 10]. Unfortunately, these systems often require manual initialization and cannot process camera images at interactive frame rates. Promising methods for interactive human pose estimation use a volumetric model of the body [11, 14, 24] or utilize a depth camera based on the time-of-flight principle [7] or the newly released Microsoft Kinect camera [20]. A more detailed review of current literature can be found in Section 2.

A common disadvantage of many marker-less, real-time capable methods is that they require a learning phase in order to train exemplar based algorithms [12], regression models [11] or depth image segmentation [20]. When an unrestricted articulation of the human body is desired, huge training datasets are required considering that the pose space consists

BMVC 2011 http://dx.doi.org/10.5244/C.25.69

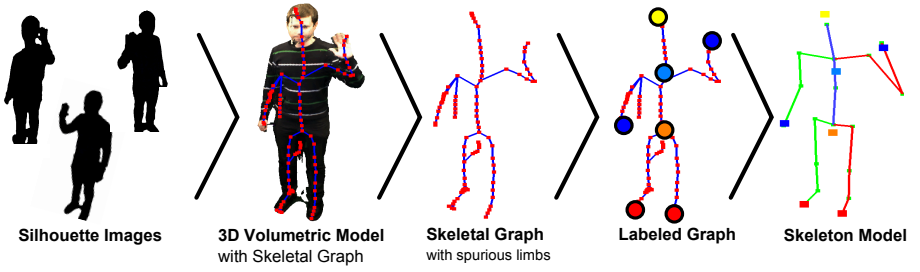


Figure 1: Process of extracting the human skeleton starting from silhouettes.

of 16 rotational degrees of freedom for arms and legs alone. In contrast, rigid part fitting methods such as [14] adapt a body model directly to image observations. However, they are easily distracted by missing body-parts or spurious limbs that often occur in volumetric body scans and depend on a good initialization. Therefore there is need for improvement in real-time body pose estimation methods to handle the full articulation space of the human body, support automatic single-frame initialization and tolerate outliers.

In this paper, we present a novel marker-less human pose estimation algorithm which uses a skeletal graph extracted from a volumetric representation of the human body. The skeletal graph is a tree that has the same topology as the human body (i.e. arms, legs and body) which can be generated efficiently using center-line tracing algorithms [19]. As the center-line extraction produces spurious branches, we employ a novel pose-independent graph matching algorithm which robustly classifies end-nodes of the graph into head, hands and feet while ignoring such end-nodes that do not correspond to any valid limb. Using these correspondences, we obtain a good initialization for fitting a human skeleton model onto the graph. Finally, we optimize all skeleton joint positions using a fast local optimization similar to [3] while enforcing that joints lie on the skeletal graph and bones maintain their lengths. A graphical summary of our algorithm is shown in Figure 1.

The key benefits of our method are the robustness of limb-labeling and its ability to perform frame-by-frame pose estimation at low computational cost due to early reduction of the input data. It does not require any learning phase nor a database with training images which makes it particularly easy to implement. In Section 3 we explain the details of our algorithm. An extensive evaluation in Section 4 shows that our algorithm outperforms other methods in both speed and robustness. Finally, Section 5 provides conclusions as well as suggestions for future enhancements.

2 Related Work

In this section we give an overview of related work concerning human pose estimation in general as well as skeleton based methods in particular. A recent survey [17] has identified two main approaches for human pose estimation: bottom-up and top-down. Top-down methods match the projection of the human body with image observations by searching the pose-space for a global (or local) optimum. These methods allow very accurate pose estimation from multiple views [6] and even monocular input data [8, 10]. However, optimizing for the correct pose is not trivial due to the high dimensionality of the problem. The evaluation of the cost function is therefore often computationally expensive. Bottom-up methods on

the other hand try to find body parts in one or more input images and combine them to a complete human body. For example, part-based approaches [4, 20, 25] generate probability maps for each body part in one or multiple input images and determine the best fitting body pose by filtering out the most probable positions.

An established way to combine bottom-up and top-down methods for human pose estimation is to use the underlying skeleton structure of the body. This has the advantage that image data is simplified and processing of the information is generally very fast and robust. In [23], Thome *et al*. use 2D silhouette images to generate a skeletal graph through Delaunay triangulation and assign limb labels to the end-nodes through graph matching. Menier *et al*. [15] state that using a single 2D silhouette only is prone to occlusion and that a 3D skeleton cannot be mapped to a 2D skeleton graph directly. Instead, they use multiple silhouette images to generate the visual hull of the body, extract medial axis points using Voronoi centers and fit a skeleton model using a probabilistic framework. A similar procedure is proposed by Vlasic *et al*. [26] who fit a skeleton model to the medial axis of the visual hull by minimization of an objective function. Moschini and Fusiello [16] approximate the 3D medial axis directly from 2D silhouettes and employ an Iterative Closest Point (ICP) algorithm for fitting a skeleton model. A hybrid 2D/3D approach is presented by Correa *et al*. [5] who fit a skeleton in multiple 2D silhouettes and combine them in a 3D fusion step. Sundaresan and Chellappa [22] have managed to solve skeletonization of the human body from 3D voxel data using the Laplacian manifold embedding framework to extract 1D limb lines. They even can handle cases where limbs form loops such as when hands touch the body. Similar to our approach, they generate a skeletal graph and identify limbs and fit a skeleton model.

A common weakness of most methods that process skeleton data is that they assume a very good skeletonization and a stable number of graph nodes and branches. Our observations have shown that such assumptions only hold when the input images are perfectly segmented and no ghosting limbs occur in the visual hull representation. Under such circumstances many skeleton extraction methods produce errors or spurious nodes which need to be handled in a robust manner.

3 Method

In this paper we show how to overcome the weaknesses of previous skeleton based methods using a fast skeleton graph extraction technique [19] combined with a novel end-node matching algorithm. We assume the availability of a calibrated multi-view camera setup which is able to provide a stream of background segmented silhouettes of a human body. In order to obtain the visual hull from these silhouette images, we perform space carving on a discrete voxel grid. A recent study by Straka *et al*. [21] shows that it is possible to provide such data in real-time on a single computer.

Processing of voxel data can be computationally inefficient when using a high resolution voxel space. The main idea of our method is to reduce the amount of data from roughly 10^6 voxels to a skeletal graph which consists only of about 10^2 connected nodes (see Figure 1 for an example). This can be achieved very efficiently by an algorithm called *voxel scooping* [19]. However, due to non optimal segmentation of silhouette images or space carving artifacts it is possible that the resulting graph contains spurious branches that need to be robustly detected and removed in order to find those branches that represent arms and legs. The cleaned graph can then be used to optimize a full-body skeleton model so that it represents the pose of the body in the original images.

3.1 Generating the Skeletal Graph

We extract the skeletal graph G from a volumetric human body representation using *voxel scooping* [19] which extracts the center-line from voxel data. The main advantage of this algorithm is the quality of the extracted center-line as well as its speed. Compared to similar algorithms like [27] which are able to process only around 6000 voxels per second, [19] is able to generate a skeletal graph by processing up to 3 million voxels per second.

We assume that the person is standing upright and that the highest voxel in our volumetric voxel representation corresponds to the top of the head. If the person is allowed to raise the hands above the head, it is possible to track local height maxima after initialization or determine the head position by a face detection algorithm, which leads to a higher computational cost. We take the head position as the starting point for the skeletal graph which is then iteratively generated by spherical expansion and branching through the body volume [19]. The result of this operation is a graph similar to the human skeleton but with many spurious nodes and branches (such as in Figure 1). A pruning step is used to remove very short branches. Using a novel graph-matching algorithm, we can classify the remaining branches as outliers or valid hands and feet.

3.2 Skeleton Graph Matching using End-Nodes

We provide a template skeleton G' of the human body with approximately the same bone lengths as the user and limb-labels for each end-node but with no specific pose. By robustly matching end-nodes of the skeletal graph G with this template, it is possible to propagate limb-labels while ignoring spurious branches. The matching algorithm needs to be robust to pose changes and must not rely on the position of branching nodes (hips and neck), as their position can vary due to clothing or the body pose. We use a graph matching method inspired by Bai and Latecki [2]. They perform shape recognition from 2D silhouette images based on skeletal graph matching. Their main idea is to match two graphs by comparing geodesic paths between end-nodes of the skeletal representation of the objects. Hence matching is performed independent of the graph topology and the articulation of the object. Furthermore, there is no dependence on the global pose of the object in the input images.

We adopt this idea for finding correspondences between a 3D skeletal graph and a template graph with the properties of the human skeleton. One cannot apply the method of [2] directly to a 3D graph, as the their algorithm requires the end-nodes to be ordered along the outer contour of the silhouette image. In a voxel model such ordering along a 1D line is not possible as the volume is bounded by a 2D surface which does not allow for a deterministic sorting algorithm. We therefore propose a different strategy that takes advantage of the known head position as the root of the skeletal graph: we sort the end-nodes in ascending order of the length of the geodesic path to the head-node. This ordering preserves robustness in presence of articulated body movements and is independent from the graph topology.

3.2.1 Constructing Pairwise Distance Matrices

In contrast to [2], where a descriptor for the graph is calculated by sampling a distance field along the geodesic path between each two end-nodes, we describe a graph solely by geodesic distances. For the skeletal graph G with N end-nodes $\Omega = \{\omega_1, \omega_2, \ldots, \omega_N\}$ ordered in ascending geodesic distance to the root-node (i.e. the head-node), we define a pair-wise

distance matrix

$$\mathbf{D} = \begin{bmatrix} gd(\omega_1,\omega_1) & gd(\omega_1,\omega_2) & \dots & gd(\omega_1,\omega_N) \\ \vdots & \vdots & \ddots & \vdots \\ gd(\omega_N,\omega_1) & gd(\omega_N,\omega_2) & \dots & gd(\omega_N,\omega_N) \end{bmatrix} \tag{1}$$

where $gd(\omega_i,\omega_j)$ is the length of the geodesic path between end-nodes ω_i and ω_j of graph G (i.e. accumulated Euclidean distances between all nodes along the path). These distances can be efficiently determined using Dijkstra's shortest path algorithm. Each row of $\mathbf{D}$ can be seen as a descriptor for an end-node which contains distances to all other end-nodes of the same graph (including spurious end-nodes).

3.2.2 Robust Matching of Distance Matrices

In order to assign a label to each end-node of the skeletal graph G with the descriptor matrix $\mathbf{D}$, we need to find correspondences to end-nodes of a template skeleton graph G' with a descriptor matrix $\mathbf{D}'$ (in Figure 2(a) we show this template). This is done by comparing every end-node of graph G to all end-nodes of graph G', or more precisely calculating the matching cost for every pair-wise combination of row-vectors from both distance matrices $\mathbf{D}$ and $\mathbf{D}'$. Generally, the query graph G will not contain the same number of end-nodes N as the template graph G' ($N' = 5$) because some end-nodes are missing or spurious. Consequently the number of columns of $\mathbf{D}$ and $\mathbf{D}'$ will be different and standard comparison methods can not be used. We therefore need an algorithm to calculate the matching cost of feature vectors with different length.

Dynamic time warping (DTW) can be applied to various problems that consist of matching sequences of different lengths. It makes use of dynamic programming (DP) in order to efficiently calculate the minimal matching cost of a sequence $a = \alpha_1,\alpha_2,\dots,\alpha_A$ of length A when matching, skipping or deleting elements of another sequence $b = \beta_1,\beta_2,\dots,\beta_B$. DTW iteratively populates a $(A+1)\times(B+1)$ matrix $\mathbf{W}$. First, the matrix is initialized along the border:

$$\mathbf{W}(0,j) = \infty\big|_{j=1\dots B} \qquad \mathbf{W}(i,0) = \infty\big|_{i=1\dots A} \qquad \mathbf{W}(0,0) = 0 \tag{2}$$

By applying DP one can calculate its elements using the following procedure:

$$\mathbf{W}(i,j) = c(\alpha_i,\beta_j) + \min\{\mathbf{W}(i-1,j-1),\mathbf{W}(i-1,j),\mathbf{W}(i,j-1)\} \tag{3}$$

where $c(\alpha_i,\beta_j)$ denotes a cost function which compares α_i to β_j. In the simplest case, this is the absolute difference $c = |\alpha_i - \beta_j|$. The minimal matching cost for both series can be determined by evaluating the time warping matrix at $\mathbf{W}(A,B)$. For a more detailed discussion of dynamic time warping the reader is referred to [18].

For matching end-nodes of the skeletal graph, we make use of dynamic time warping in order to match sequences of ordered end-nodes of different length. In contrast to other applications of DTW, such as time-series processing, we are not interested in the optimal alignment of two series but rather in the total matching cost as a measure of their similarity. We define a cost matrix $\mathbf{C}$ that contains a matching cost for every pair of end-nodes of graphs G and G':

$$\mathbf{C}(\mathbf{D},\mathbf{D}') = \begin{bmatrix} mc(\mathbf{D}_1,\mathbf{D}'_1) & mc(\mathbf{D}_1,\mathbf{D}'_2) & \dots & mc(\mathbf{D}_1,\mathbf{D}'_{N'}) \\ \vdots & \vdots & \ddots & \vdots \\ mc(\mathbf{D}_N,\mathbf{D}'_1) & mc(\mathbf{D}_N,\mathbf{D}'_2) & \dots & mc(\mathbf{D}_N,\mathbf{D}'_{N'}) \end{bmatrix} \tag{4}$$

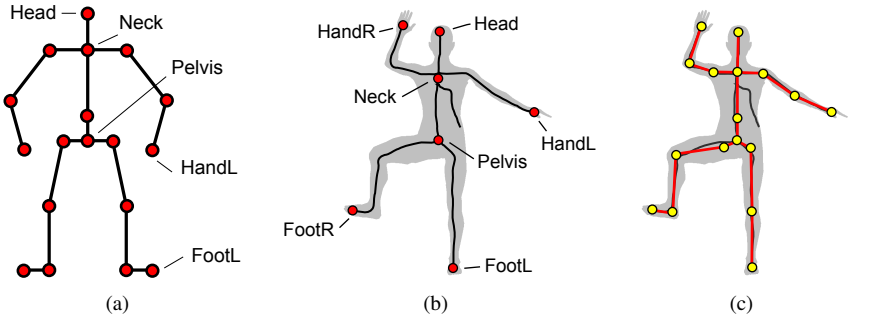


Figure 2: Template skeleton model (a). Skeletal graph with labeled end-nodes and nodes for neck and pelvis (b). Optimized skeleton model that fits to the skeletal graph (c).

where $mc(\mathbf{D}_i, \mathbf{D'}_j)$ denotes the minimal matching cost of the i-th row of matrix $\mathbf{D}$ and the j-th row of matrix $\mathbf{D'}$ which is obtained by evaluating the DTW matrix at $\mathbf{W}(N, N')$ when using both row vectors as input sequences for time warping. The optimal correspondence of end-nodes of the skeletal graph G to graph G' can be found using bipartite graph-matching based on the cost matrix $\mathbf{C}$, which can be efficiently performed by the Hungarian algorithm [13]. Note that it is not possible to distinguish left and right limbs due to the body symmetry. We determine the correct side of each limb in a post processing step.

3.3 Skeleton Fitting

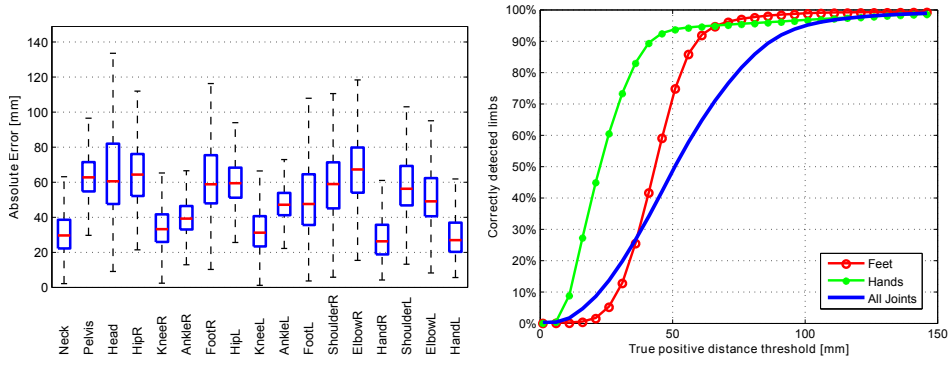
In addition to the end-nodes of the graph, we need to determine two additional interior nodes (pelvis and neck) in order to initialize our skeleton model (see Fig. 2(a)). We find these nodes by reusing the idea of the graph distance in order to define a discriminative interior-node descriptor:

$$F_g(v) = \left[gd_g(v, \text{Head}), gd_g(v, \text{HandL}), gd_g(v, \text{HandR}), gd_g(v, \text{FootL}), gd_g(v, \text{FootR})\right]^T \quad (5)$$

where v denotes a node in the graph g and $gd_g(v, \omega)$ the geodesic distance between the node v and an end-node ω of this graph. We use names for the nodes ω in order to make the procedure more comprehensible. For the pelvis and neck node we calculate a descriptor $F_{G'}(\text{Pelvis})$ and $F_{G'}(\text{Neck})$. We evaluate $F_G(v)$ for each node v in the skeletal graph G in order to find the best match for the pelvis and neck:

$$v_{\text{pelvis}} = \arg\min_{v \in G} \|F_G(v) - F_{G'}(\text{Pelvis})\|^2 \qquad v_{\text{neck}} = \arg\min_{v \in G} \|F_G(v) - F_{G'}(\text{Neck})\|^2 \quad (6)$$

As a result, the skeletal graph has seven labeled nodes as in Figure 2(b) which correspond to the human skeleton. We initialize our skeleton model with the positions of the nodes for head, limbs and inner joints. It is then possible to use any local optimization method to refine the skeleton model until it fits to the graph. We propose to use a method similar to [3], which optimizes joint positions until they fit nicely inside the body while ensuring that bones have the same length as in the template skeleton (see Fig. 2(c)). Our implementation, however, attracts bones towards the skeletal graph instead of the center of the medial body surface, thus we do not need to determine the medial surface at all.



(a) (b)

Figure 3: Boxplots showing the distance between estimated joint positions and ground truth data (a) and how many joints are correctly classified when increasing the confidence threshold (both charts are based on over 19K frames) (b).

4 Experiments

In this section, we show qualitative results on our own recordings and public datasets as well as a quantitative evaluation on synthetic scenes with ground-truth.

4.1 Data Acquisition

In order to generate a volumetric body scan, we perform GPU accelerated space carving of silhouette images obtained from multiple camera images. The silhouette images are generated by background subtraction using a colored background. Unless otherwise specified, we use a $96 \times 96 \times 128$ voxel grid with a resolution of about 15 mm per voxel. For qualitative evaluations we use our own hardware setup consisting of 10 synchronized color cameras [21] and publicly available datasets [6]. To obtain ground truth joint position data for our quantitative evaluation, we use motion capture data [1] to animate a human polygon model and render it from multiple views. The joint positions from the motion capture system are then compared to our estimations based on the synthetic images.

4.2 Quantitative Results

We quantify the joint position estimation accuracy as well as the robustness of end-node labeling. As an error measure we calculate the Euclidean distances of our joint pose estimates to the corresponding ground truth position given by the motion capture data in every frame of several sequences of the CMU motion capture database [1]. In total, we have evaluated our algorithm on almost twenty thousand frames of sequences containing a variety of movements.

In Figure 3(a) we show that the median of the distances stays below 80 mm for all joints while the distance for end-joints such as hands and feet is even smaller. Note that the 25th and 75th percentiles in the boxplot are less than twice the voxel size apart. This suggests that most errors are systematic due to structural differences between our template skeleton model and the skeleton used in the motion capture database. In Figure 3(b) we show the

Sequence	# Frames	FootR	FootL	HandR	HandL
01_01	2750	0 (0.0%)	0 (0.0%)	51 (1.9%)	36 (1.3%)
02_01	342	0 (0.0%)	0 (0.0%)	10 (2.9%)	47 (13.7%)
02_05	1854	1 (0.1%)	0 (0.0%)	5 (0.3%)	85 (4.6%)
03_01	431	0 (0.0%)	0 (0.0%)	11 (2.6%)	5 (1.2%)
05_02	1122	8 (0.7%)	5 (0.4%)	5 (0.4%)	7 (0.6%)
06_04	395	1 (0.3%)	0 (0.0%)	6 (1.5%)	10 (2.5%)
13_17	4839	29 (0.6%)	28 (0.6%)	119 (2.5%)	48 (1.0%)
13_18	2999	37 (1.2%)	41 (1.4%)	162 (5.4%)	129 (4.3%)
13_29	4591	75 (1.6%)	96 (2.1%)	172 (3.7%)	90 (2.0%)
16_11	533	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Average	19856	151 (0.8%)	170 (0.9%)	541 (2.7%)	457 (2.3%)

Table 1: Evaluation of the foot/hand classification errors on some sequences of the CMU motion capture database [1]. Each limb that is not within a 100 mm radius of the ground truth joint position is counted as an error.

effect of increasing the classification threshold on true positives for our skeleton fitting. On average, all joints are within 50 mm of their ground truth position in more than half of all frames of our test sequences. When we increase this threshold to 100 mm, we estimate all joints correctly in more than 95% of all frames. Again, the performance for hands and feet is superior to other joints as their position is determined directly by the graph matching step (see Sect. 3.2.2), while other joints depend on the position of hands and feet.

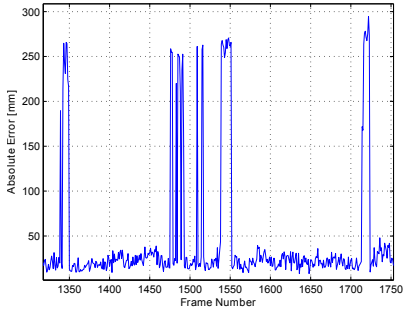
A major benefit of our algorithm is single-frame recovery. In Figure 4(a) we show the estimation error for the left hand over the period of some frames of a motion capture sequence. We deliberately have chosen a time interval with many errors that cause a jump of the hand position in the resulting skeleton. Even though the position can not be determined accurately in some frames, our single-frame estimation prohibits that the hand gets stuck in the erroneous position. These findings are also supported by Table 1 which gives a detailed count of how often each limb is away more than 100 mm from its ground truth position in several motion capture sequences. Even on long sequences the error rate stays below 3% on average.

4.3 Qualitative Results

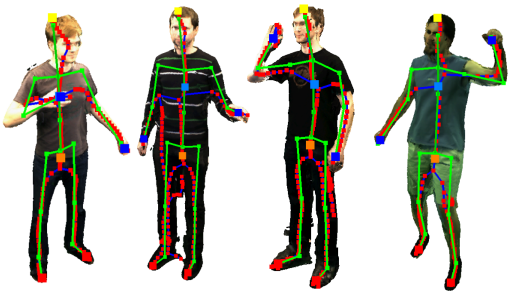
In Figure 4(b) we show qualitative results of our skeleton fitting algorithm. The input images are taken from our own multi-camera hardware setup [21] and publicly available datasets [6]. Colored squares mark the detected joint positions in the graph matching step that are used for initialization of the skeleton model. Note that even long spurious branches do not affect our node classification.

4.4 Comparison of Results

Top-performers in the field of human pose estimation from multi-view images achieve average joint position errors of around 50–100 mm [6, 12] at the cost of a processing time of more than one second per frame. Even then, such methods rely on tracking information and can get stuck in local minima for a long time if tracking information is wrong. Our system does not depend on temporal information and is capable of providing the same error rates



(a)



(b)

Figure 4: Distance between the left hand its ground truth position in an interval taken from the sequence 13_18 of [1] (a). Skeleton model automatically fitted to the human body (b). We used [3] for rendering the 3D model.

at up to 30 frames per second. Other methods such as [7, 20] rely less on tracking data and provide real-time performance at similar error rates but incur a substantial training effort for part detectors or require a database of exemplar images [25]. We do not require any training data but require only a skeleton model with known dimensions. Previous skeletal graph based methods that work on 2D images [5, 23] or 3D data [15, 22, 26] can either not operate at interactive frame rates or work only if users directly face the camera.

4.5 Runtime Evaluation

We performed our experiments on a state of the art PC system equipped with the Nvidia GTX 280 graphics card and an Intel Core i7 processor. It is possible to estimate a human skeleton model in real-time at 15 frames per second using our algorithm, limited only by the frame rate of our camera setup. Given silhouette images of the body from multiple views, we were able to generate a voxel model in 10 ms on the GPU and then use a single CPU thread to extract a skeletal graph within 6 ms and fit a skeleton to the graph in less than 1 ms. This allows for skeleton estimation at more than 30 frames per second. The main reason for this efficiency is the reduction of data early on: we compress the voxel model by expressing its structure with a graph consisting of at most a few hundred nodes. Processing of such graphs is very efficient, especially when only end-nodes are of interest.

5 Conclusions and Future Work

A novel method to fit a human skeleton to multiple image observations has been proposed. The technique uses silhouette images to build a voxel model of the human body and extracts a skeletal graph from this volumetric representation. By using a robust matching algorithm based on geodesic distances it is possible to assign labels such as hand and foot to the end-nodes of this graph as well as to determine important inner-nodes such as the neck or pelvis. These special nodes are used to initialize a local skeleton refinement step which ensures that the complete skeleton model fits the body in an optimal way. We have demonstrated that such a system can be implemented fast enough to operate at interactive frame rates, which is achieved by an early reduction of voxel/image data to the skeletal graph. The

algorithm estimates most poses accurately within 100 mm of their ground truth position and automatically recovers from an erroneous pose due to single-frame processing (we do not require any tracking). Experiments have shown that hands and feet are detected correctly in more than 97% of all frames in various test sequences.

Currently, we are working on handling cases where the skeletal graph becomes corrupt as a result of arms being too close to the upper body. Also, tracking over time can resolve ambiguities between two end-nodes with similar descriptors and therefore will reduce the number of false classifications.

Acknowledgements

This work was supported by the Austrian Research Promotion Agency (FFG) under the BRIDGE program, project #822702 (NARKISSOS).

References

[1] CMU Mocap Database. URL http://mocap.cs.cmu.edu/.

[2] Xiang Bai and Longin Jan Latecki. Path similarity skeleton graph matching. *Pattern Analysis and Machine Intelligence*, 30(7):1282–1292, 2008.

[3] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3D characters. In *Proc. of the ACM SIGGRAPH*, 2007.

[4] Martin Bergtholdt, Joerg Kappes, Stefan Schmidt, and Christoph Schnoerr. A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision*, 87:93–117, 2010.

[5] Pedro Correa, Ferran Marqués, Xavier Marichal, and Benoit Macq. 3D posture estimation using geodesic distance maps. *Multimed. Tools Appl*, 38:365–384, 2008.

[6] Jürgen Gall, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel. Optimization and filtering for human motion capture. *International Journal of Computer Vision*, 87: 75–92, 2010.

[7] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. Real time motion capture using a single time-of-flight camera. In *Proc. of Computer Vision and Pattern Recognition*, 2010.

[8] Peng Guan, Alexander Weiss, Alexandru O. Bălan, and Michael J. Black. Estimating human shape and pose from a single image. In *Proc. of the International Conference on Computer Vision*, 2010.

[9] Stefan Hauswiesner, Matthias Straka, and Gerhard Reitmayr. Coherent image-based rendering of real-world objects. In *Proc. of the Symposium on Interactive 3D Graphics*, 2011.

[10] Yap Wooi Hen and Raveendran Paramesran. Single camera 3d human pose estimation: A review of current techniques. In *Proc. of the International Conference for Technical Postgraduates*, 2009.

[11] Michiro Hirai, Norimichi Ukita, and Masatsugu Kidode. Real-time pose regression with fast volume descriptor computation. In *Proc. of the International Conference on Pattern Recognition*, pages 1852–1855, 2010.

[12] Michael Hofmann and Dariu M. Gavrila. Multi-view 3D human pose estimation combining single-frame recovery, temporal integration and model adaptation. In *Proc. of Computer Vision and Pattern Recognition*, 2009.

[13] Harold Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[14] Xinghan Luo, Berend Berendsen, Robby T. Tan, and Remco C. Veltkamp. Human pose estimation for multiple persons based on volume reconstruction. In *Proc. of the International Conference on Pattern Recognition*, 2010.

[15] Celment Menier, Edmond Boyer, and Bruno Raffin. 3D skeleton-based body pose recovery. In *Proc. of the International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 389–396, 2006.

[16] Davide Moschini and Andrea Fusiello. Tracking human motion with multiple cameras using an articulated model. In *Proc. of the International Conference on Computer Vision/Computer Graphics Collaboration Techniques*, 2009.

[17] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1–2):4–18, 2007.

[18] Chotirat Ann Ratanamahatana and Eamonn Keogh. Everything you know about dynamic time warping is wrong. In *Workshop on Mining Temporal and Sequential Data*, 2004.

[19] Alfredo Rodriguez, Douglas Ehlenberger, Patrick Hof, and Susan L. Wearne. Three-dimensional neuron tracing by voxel scooping. *Journal of Neuroscience Methods*, 184 (1):169–175, 2009.

[20] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Proc. of Computer Vision and Pattern Recognition*, 2011.

[21] Matthias Straka, Stefan Hauswiesner, Matthias Rüther, and Horst Bischof. A free-viewpoint virtual mirror with marker-less user interaction. In *Proc. of the 17th Scandinavian Conference on Image Analysis*, 2011.

[22] Aravind Sundaresan and Rama Chellappa. Model driven segmentation of articulating humans in laplacian eigenspace. *Pattern Analysis and Machine Intelligence*, 30(10): 1771–1785, 2008.

[23] Nicolas Thome, Djamel Merad, and Serge Miguet. Human body part labeling and tracking using graph matching theory. In *Proceedings of the IEEE Conference on Video and Signal Based Surveillance*, 2006.

[24] Cuong Tran and Mohan M. Trivedi. Human body modeling and tracking using volumetric representation: Selected recent studies and possibilities for extensions. In *Proc. of the International Conference on Distributed Smart Cameras*, 2008.

[25] Michael Van den Bergh, Esther Koller-Meier, and Luc Van Gool. Real-time body pose recognition using 2D or 3D haarlets. *International Journal of Computer Vision*, 83: 72–84, 2009.

[26] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics*, 27(3):1–9, 2008.

[27] Yu-Shuen Wang and Tong-Yee Lee. Curve-skeleton extraction using iterative least squares optimization. *IEEE Transactions on Visualization and Computer Graphics*, 14 (4):926–936, 2008.