Real-time Human Pose Estimation from Body-scanned Point Clouds

Jilliam María Díaz Barros[1,2], Frederic Garcia[2] and Désiré Sidibé[1]
[1]*Le2i - UMR CNRS 6306, Université de Bourgogne, 12 rue de la Fonderie, Le Creusot, France*
[2]*SnT, University of Luxembourg, 4 rue Alphonse Weicker, Luxembourg, Luxembourg*

Keywords: Human Pose Estimation, Point Cloud, Skeleton Model.

Abstract: This paper presents a novel approach to estimate the human pose from a body-scanned point cloud. To do so, a predefined skeleton model is first initialized according to both the skeleton base point and its torso limb obtained by Principal Component Analysis (PCA). Then, the body parts are iteratively clustered and the skeleton limb fitting is performed, based on Expectation Maximization (EM). The human pose is given by the location of each skeletal node in the fitted skeleton model. Experimental results show the ability of the method to estimate the human pose from multiple point cloud video sequences representing the external surface of a scanned human body; being robust, precise and handling large portions of missing data due to occlusions, acquisition hindrances or registration inaccuracies.

1 INTRODUCTION

Human pose estimation is indispensable in very active research areas such as scene understanding, human-computer interaction and action or gesture recognition. Among the vast literature on this fundamental research topic, many authors have considered predefined human models to simplify the pose estimation task when using conventional 2-D cameras. For instance, Ke *et al.* in (Ke et al., 2011) retrieve the human pose from a monocular camera, using downhill simplex algorithm to match 2-D feature points to a predefined 3-D human model. Other approaches specifically parameterize the pose in a lower dimensional space, using skeleton models. An example is the work of Li *et al.* in (Li et al., 2009), where the authors estimate the 2-D human pose in a video sequence using a predefined human skeleton model to fit the silhouette of a body shape. Distance Transform (DT) and Principal Component Analysis (PCA) are used to identify the skeleton base point and to initialize the skeleton. Then, they perform an iterative process to cluster the body parts to which they fit the predefined skeleton model. However, a renewed interest has arisen as a side-effect of the recent advances in 3-D sensing technologies. Indeed, recent consumer-accessible depth cameras such as the Kinect or the Xtion Pro Live provide remarkable advantages, such as easily overcoming the background matting problem, *i.e.,* segmenting foreground objects from the scene background. Although recent approaches based on depth cameras provide very promising human pose estimates, most of them are intended for mono-view systems and thus limited to applications in which the user is facing to the camera. Approaches intended for multi-view systems and thus to estimate the human pose from a full body scan, usually extract a curve-skeleton representation of the shape to which the skeleton model can be fitted and hence, estimate the pose. However, these approaches are impractical for applications in which the human pose must be estimated in real-time.

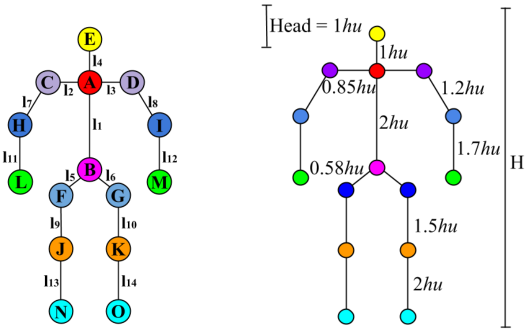
In this paper, the problem of human pose estimation is addressed in the context of 3-D scenes scanned by multi-view systems composed of multiple depth cameras. To enable for real-time applications, prior knowledge such as a predefined human body skeleton model is also incorporated, from which its skeletal joints will define the configuration and thus the pose of the scanned body. The remainder of this paper is organized as follows: Section 2 presents a review of human pose estimation based on depth sensing. In Section 3 a detailed description of the current approach is presented. Section 4 evaluates the proposed approach on both synthetic and real data. Finally, concluding remarks are given in Section 5.

2 RELATED WORK

Related to single-depth-image pose estimation, Ye et

Díaz Barros J., Garcia F. and Sidibé D..
Real-time Human Pose Estimation from Body-scanned Point Clouds.
DOI: 10.5220/0005309005530560
In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (VISAPP-2015), pages 553-560
ISBN: 978-989-758-089-5

al. (Ye et al., 2011) proposed a pipeline to combine pose detection with pose refinement. To do so, the depth map is used to find a similar pose within a database of prior full-body surface mesh models. Lehment *et al.* (Lehment et al., 2010) considered 3-D point clouds extracted from depth maps to fit a mesh of a cylinder-based stickman model using Annealing Particle Filters (APF). However, the aforementioned methods require a GPU-based implementation. Shotton *et al.* (Shotton et al., 2013) introduced two super-real-time approaches to predict the positions of body joints using a large and varied synthetic set of training images. Decision forests and simple depth-invariant image features are implemented. In (Zhang et al., 2012), Zhang *et al.* considered a multi-view setup with depth cameras to perform human pose estimation and tracking. The method employs APF and partition sampling in point cloud models, handles occlusions and reduces ambiguities. The initial pose is estimated using a coarse-to-fine search paradigm. To the best of our knowledge, this is the only method using a multiple-depth-camera setup for human pose estimation. Curve-skeleton-extraction approaches have been successfully used for different kind of shapes besides human models. Since they preserve the geometry and topological information of the object, they can be implemented in human pose estimation by approximating the underlying skeletal structure (Garcia and Ottersten, 2014b). Au *et al.* at (Au et al., 2008) proposed a Laplacian-based contraction method intended only for watertight mesh surfaces. An extension of this work was proposed by Cao *et al.* (Cao et al., 2010) to handle surfaces with boundaries, polygon soups and point clouds. Although both methods are robust against noise and moderate miss of data, they are not optimized for real-time applications. Tagliasacchi *et al.* presented in (Tagliasacchi et al., 2009) a method to extract curve skeletons based on a generalized rotational symmetry axis (ROSA) of an oriented point cloud. A similar approach was proposed by Sam *et al.* (Sam et al., 2012) using the antipodes as a reference. Both methods can handle significant missing data, but require parameter tuning. A real-time curve-skeleton extraction method was proposed by Garcia and Ottersten (Garcia and Ottersten, 2014b) in which, inspired from (Sam et al., 2012), the skeletal candidates are extracted in the 2-D space and then back-projected to the 3-D space. The algorithm is robust against significant portions of missing data. Limitations are related to occluded body parts and limbs located very close to each other.



(a) (b)

Figure 1: Predefined human skeleton model. (a) Skeleton limbs and nodes. (b) Human body proportions, with *hu* the height of the head (head units).

3 PROPOSED APPROACH

A novel approach to estimate the human pose from a body-scanned point cloud $\mathcal{P}$ describing a set of 3-D points $\mathbf{p}_i = (p_x, p_y, p_z)$ representing the underlying external surface of a human body is outlined below. Similarly to (Li et al., 2009), the current approach has considered an articulated human skeleton model composed of 15 nodes and 14 edges, presented in Fig. 1 (a). By doing so, the complexity and flexibility of the human body as well as the high dimensionality of the pose space are reduced. The predefined skeleton model represents a simplified version of the geometry and topology of the human skeleton. Although there are subtle differences between people, human body proportions fit within a fairly standard range and thus, prior knowledge can be considered. Indeed, an average person uses to measure 7.5 times the height of his head (including the head). This in turn allowed to initialize the length of each skeleton limb as shown in Fig. 1 (b) (Hum, 2014).

In this work, $\mathcal{P}$ describes any possible body configuration of an upright person. Hence, the height of a person $(7.5 \times hu)$ is given by the difference between the maximum and minimum z coordinates within $\mathcal{P}$. That is, $hu = (p_{z_{max}} - p_{z_{min}})/7.5$, being the head the highest body part.

The human pose estimation results from the configuration of the skeletal joints after approximating the aforementioned skeleton model. This is achieved with a four-steps framework. First, both the base point of the skeleton model and the torso orientation are extracted. These two parameters allow the initialization of the torso whereas the remaining skeleton limbs are initialized by an iterative process in which the best initial skeleton limb configuration is selected. Next step concerns the clustering of the body parts, to which finally, their respective skeleton limbs are progressively approximated. The clustering and fitting are performed under a framework based on the theory of Expectation Maximization. Note that the 3-D point clustering to find the torso orientation (Section 3.1) and the initialization of the skeleton model using a predefined set of limb configurations (Section 3.2) are solely performed for the first frame. Indeed, the resulting fitted skeleton corresponds to the initial skeleton for the consecutive frame. By doing so, the time consumed during both stages is reduced, ensuring a better initial skeleton estimate for the following frames.

3.1 Torso and Base Point Extraction

The node $\mathbf{A} = (A_x, A_y, A_z)$ in Fig. 1 (a) corresponds to the base point of the skeleton model whereas the segment between nodes **A** and **B**, skeleton limb $\mathbf{l}_1 = \mathbf{B} - \mathbf{A}$, to the torso. The direction of the torso results from the Principal Component Analysis (PCA) on $\mathcal{P}$. Indeed, the direction of the principal component $\mathbf{v} = (v_x, v_y, v_z)$ coincides with the direction of the torso, assuming that $\mathcal{P}$ describes an upright person. The equation of the 3-D line in which the base point **A** lies is thus defined from the centroid of $\mathcal{P}$, *i.e.,* $\bar{\mathbf{p}} = \frac{1}{k} \cdot \sum_{i=1}^{k} \mathbf{p}_i$, $\forall \mathbf{p}_i \in \mathcal{P}$ and the normalized vector **v**. The third coordinate of **A**, *i.e.,* A_z, is retrieved from the human body proportions denoted in Fig. 1 (b), whereas the two missing coordinates result from:

$$\begin{aligned} A_x &= v_x \cdot t + \bar{p}_x, \\ A_y &= v_y \cdot t + \bar{p}_y, \end{aligned} \tag{1}$$

with $t = (A_z - \bar{p}_z)/(v_z - \bar{p}_z)$. Note that the initialization of the skeleton model and hence, the body clustering, are strongly dependent on these two parameters. Indeed, a wrong direction of the torso entails to a wrong initialization of the model and thus to an erroneous pose in space. To increase the accuracy and robustness of the torso direction, only those 3-D points that belong to the torso are considered. To do so, the torso 3-D points are classified by fitting a cylinder, a simplified geometric model that can be quickly fitted to the dataset using Random Sample Consensus (RANSAC) (Fischler and Bolles, 1981). Alternative fitting algorithms with embedded heuristic hypotheses generators can be also considered.

3.2 Initialization of the Skeleton Model

The initial skeleton model results from the skeleton limb's configuration that minimizes the distances between the 3-D points and the set of skeleton limbs,

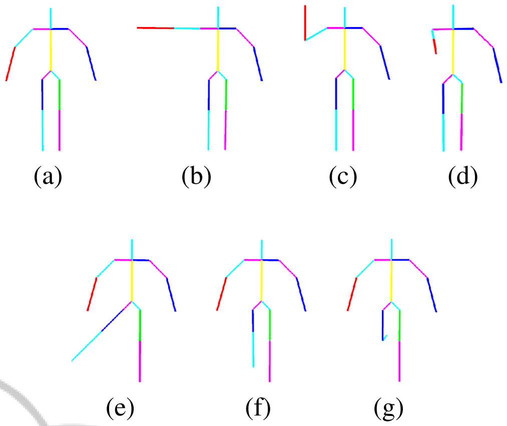


Figure 2: Considered skeleton limb and node configurations to initialize the skeleton model (only right body side configurations are shown).

i.e., the best matching between the predefined skeleton model configuration and the given point cloud $\mathcal{P}$. However, in contrast to alternative approaches to estimate the human pose from 2-D images (Li et al., 2009), this task is far from trivial when considering the additional degree of freedom in a 3-D space. First, the skeleton model is aligned to the estimated base point and torso directions. Then, the locations of the remaining skeleton nodes are progressively computed from the set of skeleton limb configurations, presented in Fig. 2, and using incorporated prior knowledge such as the radii of the skeleton limbs and the initial angles of the skeleton joints. Fig. 2 only shows the selected configurations to initialize the right body side. Nevertheless, as can be inferred, the mirrored versions correspond to the configurations of the left body side. Given a skeleton limb $\mathbf{l}_i$ with radius r_i, azimuthal angle θ_i, and polar angle ϕ_i, the 3-D coordinates of the end node $\mathbf{w} = (w_x, w_y, w_z)$ result from:

$$\begin{aligned} w_x &= u_x + r_i \cdot \cos\theta_i \cdot \cos\phi_i, \\ w_y &= u_y + r_i \cdot \sin\theta_i \cdot \cos\phi_i, \\ w_z &= u_z + r_i \cdot \sin\phi_i, \end{aligned} \tag{2}$$

with $\mathbf{u} = (u_x, u_y, u_z)$ the 3-D coordinates of the initial node. The spherical coordinate system and the right-hand rule are used to define the initial angles that generate the coordinates of each skeleton node. The angles θ and ϕ are fixed within the range of $[0, \pi]$ and $[0, 2\pi)$, respectively. First, the location of the skeleton nodes that are directly connected to the base point **A** are computed, *i.e.,* **B**, **C**, **D** and **E**. From them, the location of **F**, **G**, **H** and **I**, followed by **J**, **K**, **L** and **M** are computed. Finally, nodes **N** and **O** are computed.

When considering 3-D models that are differently oriented with respect to the z-axis, the 2^{nd} principal component is used to rotate the initialized skeleton model. In the current system configuration, the eigenvector of the 1st component corresponds to the z-axis whereas the eigenvector of the 2nd one corresponds to the x-axis. Thereby, a 3×3 rotation matrix $\mathbf{R}$ is computed by rearranging the eigenvectors obtained from PCA. The new location of the skeleton nodes is given by $\mathbf{w} = \mathbf{R} \cdot \mathbf{u}$.

3.3 Body Parts Clustering

After initializing the skeleton model, the clustering of the body parts to which each skeleton limb will be further approximated is performed. To do so, each 3-D point $\mathbf{p}_i$ is assigned to the skeleton limb $\mathbf{l}_k$ to which the distance is minimum, *i.e.,* $\mathbf{l}_k = \arg\min_{k \in [1,14]}(d(\mathbf{p}_i, \mathbf{l}_k))\ \forall \mathbf{p}_i \in \mathcal{P}$.

Nevertheless, close distances between a 3-D point and two or more skeleton limbs may induce to ambiguity in the clustering. Hence, these 3-D points are not considered within the clustering process.

3.4 Skeleton Limb Fitting

Next step concerns the medial axis estimation of each clustered body part, to which their corresponding skeleton limb will be fitted. It is important to recall that the nodes of the torso, **A** and **B**, are fixed in this step, and the remaining nodes to be computed are only connected to two limbs. From each cluster, the mean point and the three principal components are extracted using PCA. Note that the length of each skeleton limb is known from Fig. 1 (b). To decide which of the three eigenvectors corresponds to the medial axis, they are stretched from both sides starting from the mean point of the cluster and by half of their known length. As a result, three potential medial axes that are not connected to each other, but oriented towards the directions of the principal components, are obtained. The connectivity between the skeleton limbs is ensured by selecting the medial axis candidates of adjacent clusters with the shortest distances between the end node of the previously fitted limb and the initial node of the limb to be fitted. Fig. 3 details the fitting process between the left hip and left thigh skeleton limbs. Red, blue and green segments are the candidate medial axes of each cluster, depicted as cylinders. The three candidates of each cluster have exactly the same length, which corresponds to the length of their respective skeleton limb. The yellow point in Fig. 3 (a) denotes the end node of the hip limb $\mathbf{G}'$, whereas the orange dashed line is the shortest distance to the skeletal node candidates of the adjacent cluster, *i.e.,* $\mathbf{G}''$. Next, the centroid $\bar{\mathbf{q}}$ between $\mathbf{G}'$ and $\mathbf{G}''$ is calculated, shown in purple color in Fig. 3 (b), and the fitting of $\mathbf{l}_6$ is refined by reorienting it towards $\bar{\mathbf{q}}$, as shown in Fig. 3 (c). The new 3-D coordinates of $\mathbf{G}$ result from:

$$\mathbf{G} = \mathbf{B} + \lambda, \tag{3}$$

with $\lambda = d(\mathbf{B}, \mathbf{G}) \cdot (\bar{\mathbf{q}} - \mathbf{B}) / ||\bar{\mathbf{q}} - \mathbf{B}||$. Note that node **G** corresponds to the initial node of the adjacent skeleton limb, *i.e.,* the thigh. Therefore, the skeleton limb $\mathbf{l}_{10}$ is translated to its respective location, which gives an initial location for the node **K**, to be refined when fitting its adjacent skeleton limb, *i.e.,* $\mathbf{l}_{14}$.

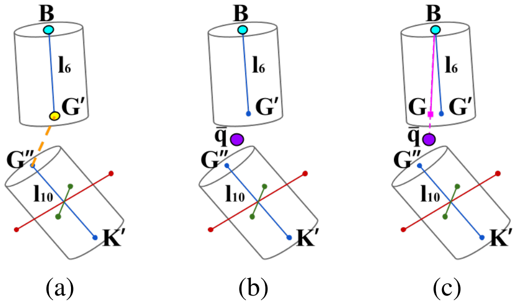


Figure 3: Fitting process of the hip limb. The medial axis candidates corresponding to the first, second and third principal components (in blue, red and green colors, respectively).

3.5 Skeleton Refinement

Similarly to (Li et al., 2009), the fitting of the skeleton model is performed through an iterative process based on Expectation Maximization (EM). The Expectation (E) step comprises Section 3.3 and the first part of Section 3.4, where the expected human skeleton is calculated for the current pose. The Maximization (M) step corresponds to the last part of Section 3.3, where the parameters that maximize the expectation of the skeleton model during the fitting process are computed. From the experiments, the fitting process converges to a good pose estimation in only one or two iterations.

4 EXPERIMENTAL RESULTS

The proposed human pose estimation approach is evaluated on both real and synthetic data. All reported results have been obtained using a Mobile Intel® QM67 Express Chipset with an integrated graphic card Intel® HD Graphics 3000. The proposed approach has been implemented in C++ language using the OpenCV (Bradski and Kaehler, 2008) and PCL (PCL, 2014) libraries. Real data has been

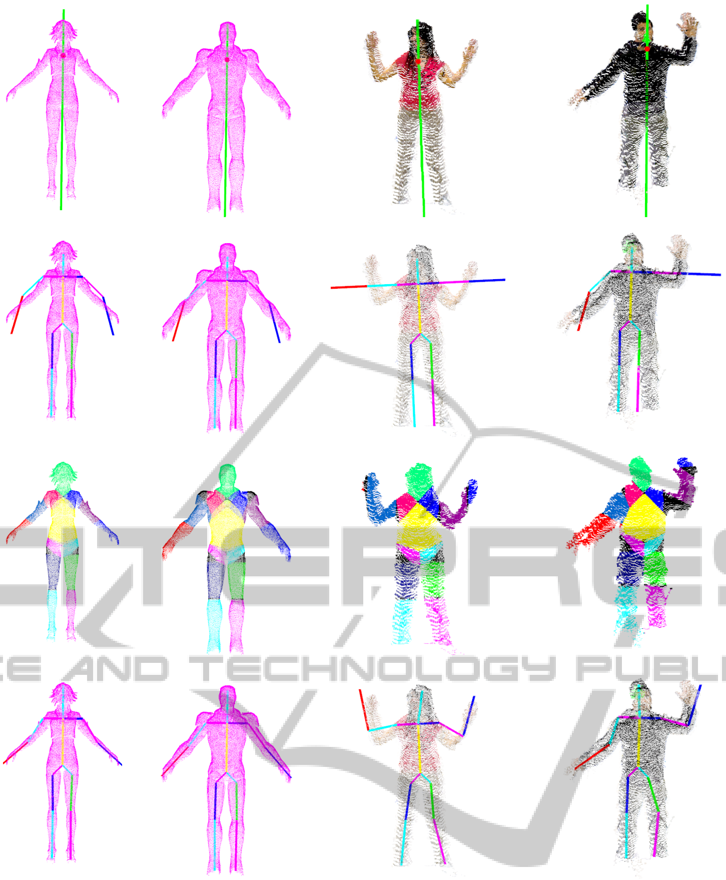


Figure 4: Human pose estimation on body-scanned point clouds using both real and synthetic data. 1^{st} row, input point cloud. 2^{nd} row, initial skeleton model. 3^{rd} row, body parts clustering. 4^{th} row, approximated skeleton model. 1^{st} col., Nilin Combat dataset. 2^{nd} col., Iron Man dataset. 3^{rd} col., Jilliam dataset. 4^{th} col., Frederic dataset.

recorded using a multi-view sensing system composed by 2 consumer-accessible RGB-D cameras, *i.e.,* the Asus Xtion Pro Live camera, with opposed field-of-views, *i.e.,* with no data overlapping. Nevertheless, the relationship between the two cameras was determined by a calibration step, using the stereo calibration implementation available in OpenCV (Bradski and Kaehler, 2008). Better registration approaches based on ICP, bundle adjustment or the combination of both can also be considered. However, the current approach perfectly estimates the human pose on such a coarse registered point clouds, handling large portions of missing data as well as registration inaccuracies. Synthetic data has been generated using V-Rep (VRE, 2014), a very versatile robot simulator tool in which the user can replicate real scenarios. A simulated scene has been created to generate the test cases with four virtual Kinect cameras installed in the top corners of the virtual scene. It can be observed that the data is perfectly registered since a full knowledge of the relationship between the cameras and their calibration parameters is known. Consequently, synthetic data has been considered as ground truth data in the next evaluations. All datasets have been voxelized to account for point redundancy after data registration. Voxelization stands for a discrete approximation of 3-D objects into a volumetric representation (Garcia and Ottersten, 2014a).

Some visual results on both synthetic and real body-scanned datasets are shown in the Fig. 4. The first row presents the considered datasets highlighting the estimated orientation of the torso (green line) and the extracted base point (purple dot). Second row shows the initialization of the predefined skeleton model. The clustering of all body parts is shown in the third row whereas the fitted skeleton model is shown in last row, from which results the human pose.

In Fig. 5, the estimated poses of the synthetic Bill model and a real dataset on some selected point-cloud video frames are shown. These results show that the method is able to accurately estimate the body pose of different body configurations of an upright person.

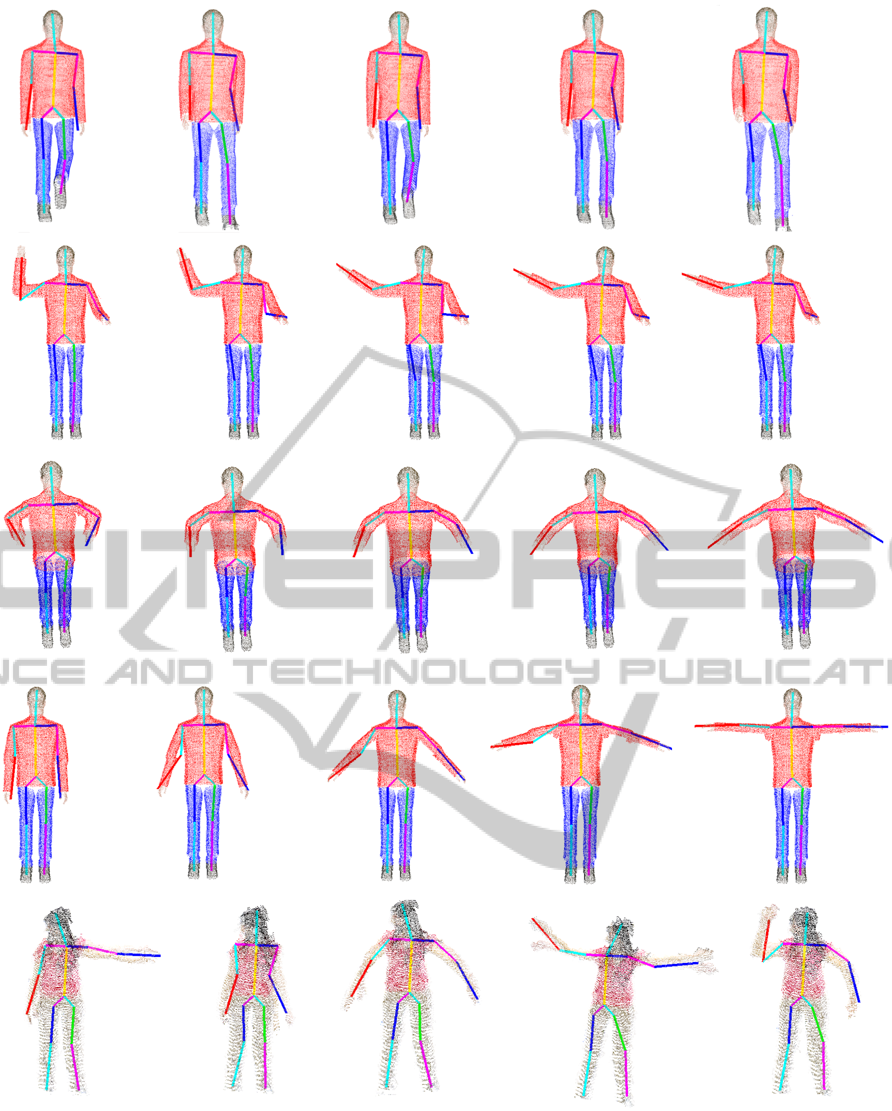


Figure 5: Human pose estimation on different frames from synthetic (1^{st} to 4^{th} rows) and real (5^{th} row) video sequences of body-scanned point clouds.

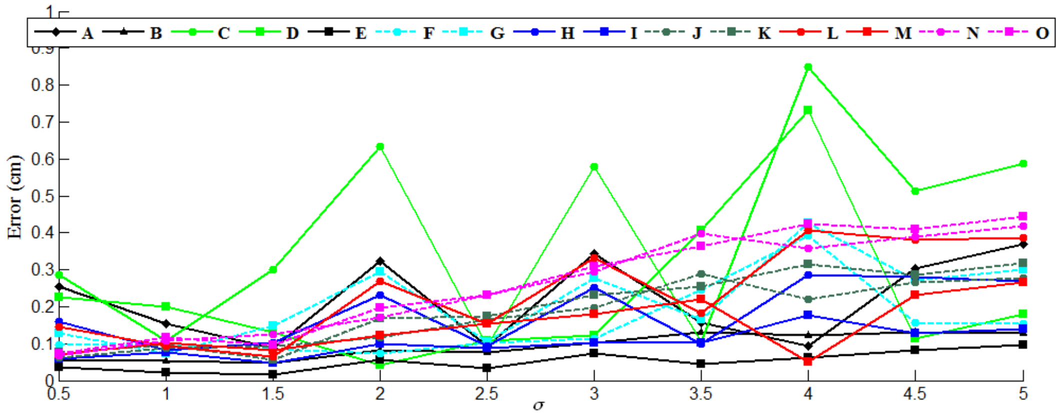


Figure 6: Fitting error of skeletal nodes for $\sigma \in [0.5, 5]$ cm.

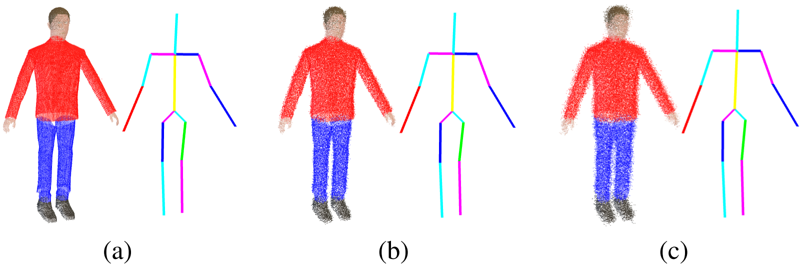


Figure 7: Bill pose estimation for (a) $\sigma = 0$ cm, (b) $\sigma = 2.5$ cm and (c) $\sigma = 5$ cm.

4.1 Robustness to Noise

Next, the robustness of the current approach against noise is evaluated. To do so, the synthetic dataset of Bill has been considered, to which it has been applied a zero-mean Gaussian noise with standard deviation $\sigma \in [0.5, 5]$ cm, *i.e.,* $\mathcal{N}(0, \sigma^2)$. In order to increase the reliability of this evaluation, the noise has been added to the depth maps acquired by each virtual Kinect camera, *i.e.,* before being transformed to point clouds. Fig. 6 depicts the error between the location of the resulting skeleton nodes from the noise-free 3-D model (considered as ground truth), and their respective ones from the noisy models. The labels of the nodes correspond to those presented in Fig. 1. As shown in Fig. 7, the proposed approach is able to estimate the Bill pose for all σ values. However, it can be observed that the fitting error of the skeletal nodes corresponding to body extremities slightly increases with amount of noise.

4.2 Runtime and Performance Analysis

Table 1 reports the time consumption to estimate the human pose for each of the datasets presented in Fig. 4. Note that most of the time is dedicated to cluster the 3-D points to estimate the direction of the torso. However, it is important to recall that this operation might be done only once in the first video frame, as discussed in Section 3.

If a better performance is required, one can increase the voxel size to represent the body-scanned point cloud. Thus, a voxel size of 1 cm^3 has been considered when evaluating the approach. However, the performance is significantly increased by increasing the voxel size, whereas the human pose remains accurately estimated, as shown in Table 2. The only constraint of this implementation is that there has to be a minimum distance of 5 cm between two 3-D points to be clustered within the same object. Indeed, this preliminary clustering is performed to cluster each individual in the scene. Another improvement may be achieved by parallelizing the initialization of the skeleton model, the clustering of body parts, and the fitting of the skeleton limbs. Indeed, the human body can be divided in four different regions, *i.e.,* left and right arms and left and right legs, that are independent and thus, can be performed in parallel.

Table 1: Time consumption analysis for human pose estimation on the datasets presented in Fig. 4 (units are in ms). Reported are the mean values taken over 100 iterations. 1st row, Nilin Combat dataset (12655 points). 2nd row, Iron Man dataset (21975 points). 3rd row, Jilliam dataset (14855 points). 4th row, Frederic dataset (19675 points).

Data set	Cluster 3-D torso points	Torso and base point extraction	Initialize skeleton model	Cluster body parts	Skeleton limb fitting	Total time
Nilin Combat	226.2	5.0	238.7	47.8	28.0	545.7
Iron Man	316.8	8.5	412.4	82.1	47.2	867
Jilliam	219.3	5.4	274.5	56.2	30.8	586.2
Frederic	220.2	5.0	275.8	54.7	32.0	587.7

Table 2: Time consumption and robustness depending on the voxel size to represent the Ironman dataset (units are in ms).

Voxel size	Cluster 3-D torso points	Torso and base point extraction	Initialize skeleton model	Cluster body parts	Skeleton limb fitting	Error (mm)
1 cm^3 (21975 points)	316.8	8.5	412.4	82.1	47.2	0.0
3 cm^3 (3516 points)	45.4	1.0	66.9	13.2	8.3	1.0
5 cm^3 (1305 points)	13.8	0.4	25.3	4.8	3.9	1.8

5 CONCLUDING REMARKS

A scheme to estimate the human pose from body-scanned point cloud datasets has been described. Using PCA and prior knowledge on human body proportions, a predefined skeleton model is initialized and then fitted to the given point cloud by an iterative process based on the theory of Expectation Maximization. From the experiments, it is shown that a good estimate is achieved in both synthetic and real datasets, and even in the presence of high noise within the depth measurements. As future work, the current approach will be extended to alternative body configurations other than upright. Alternative techniques to segment the 3-D points belonging to the torso (used to estimate its direction) will be further investigated in order to address the limitations given by the cylindrical model, such as the tuning of the model parameters.

ACKNOWLEDGEMENTS

This work was partially supported by the National Research Fund, Luxembourg, under the CORE project C11/BM/1204105/FAVE/Ottersten.

REFERENCES

(2014). Human Proportion Calculator. http://hpc.anatomy4sculptors.com/.

(2014). Point Cloud Library (PCL). http://pointclouds.org/.

(2014). Virtual Robot Experimentation Platform (V-REP). http://www.coppeliarobotics.com/.

Au, O. K.-C., Tai, C.-L., Chu, H.-K., Cohen-Or, D., and Lee, T.-Y. (2008). Skeleton extraction by mesh contraction. In *ACM SIGGRAPH 2008 Papers*, pages 44:1–44:10. ACM.

Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 1st edition.

Cao, J., A., T., M., O., Zhang, H., and Su, Z. (2010). Point cloud skeletons via laplacian based contraction. In *Shape Modeling International Conference (SMI)*, pages 187–197.

Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395.

Garcia, F. and Ottersten, B. (2014a). CPU-Based Real-Time Surface and Solid Voxelization for Incomplete Point Cloud. In *IEEE International Conference on Pattern Recognition (ICPR)*.

Garcia, F. and Ottersten, B. (2014b). Real-time Curve-Skeleton Extraction from Incomplete Point Clouds: Application in Human Pose Estimation. In *International Conference on Computer Vision Theory and Applications (VISAPP)*.

Ke, S.-R., Hwang, J.-N., Lan, K.-M., and Wang, S.-Z. (2011). View-invariant 3d human body pose reconstruction using a monocular video camera. In *Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on*, pages 1–6. IEEE.

Lehment, N. H., Arsic, D., Kaiser, M., and Rigoll, G. (2010). Automated pose estimation in 3d point clouds applying annealing particle filters and inverse kinematics on a gpu. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 87–92. IEEE.

Li, M., Yang, T., Xi, R., and Lin, Z. (2009). Silhouette-based 2d human pose estimation. In *International Conference on Image and Graphics (ICIG)*, pages 143–148.

Sam, V., Kawata, H., and Kanai, T. (2012). A robust and centered curve skeleton extraction from 3d point cloud. *Computer-Aided Design and Applications*, 9(6):969–879.

Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., et al. (2013). Efficient human pose estimation from single depth images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2821–2840.

Tagliasacchi, A., Zhang, H., and Cohen-Or, D. (2009). Curve skeleton extraction from incomplete point cloud. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pages 71:1–71:9. ACM.

Ye, M., Wang, X., Yang, R., Ren, L., and Pollefeys, M. (2011). Accurate 3d pose estimation from a single depth image. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 731–738. IEEE.

Zhang, L., Sturm, J., Cremers, D., and Lee, D. (2012). Real-time human motion tracking using multiple depth cameras. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2389–2395. IEEE.