Real-time Curve-skeleton Extraction of Human-scanned Point Clouds
Application in Upright Human Pose Estimation

Frederic Garcia and Björn Ottersten
Interdisciplinary Centre for Security Reliability and Trust (SnT), University of Luxembourg, Luxembourg City, Luxembourg

Keywords: Curve-skeleton, Skeletonization, Human Pose Estimation, Object Representation, Point Cloud, Real-time.

Abstract: This paper presents a practical and robust approach for upright human curve-skeleton extraction. Curve-skeletons are object descriptors that represent a simplified version of the geometry and topology of a 3-D object. The curve-skeleton of a human-scanned point set enables the approximation of the underlying skeletal structure and thus, to estimate the body configuration (human pose). In contrast to most curve-skeleton extraction methodologies from the literature, we herein propose a real-time curve-skeleton extraction approach that applies to scanned point clouds, independently of the object's complexity and/or the amount of noise within the depth measurements. The experimental results show the ability of the algorithm to extract a centered curve-skeleton within the 3-D object, with the same topology, and with unit thickness. The proposed approach is intended for real world applications and hence, it handles large portions of data missing due to occlusions, acquisition hindrances or registration inaccuracies.

1 INTRODUCTION

Human pose estimation not only is one of the fundamental research topics in computer vision, but a necessary step in active research topics such as scene understanding, human-computer interaction and action or gesture recognition; a side-effect of the recent advances in 3-D sensing technologies.

In this paper, we address the problem of human pose estimation in the context of 3-D scenes scanned by multiple consumer-accessible depth cameras such as the Kinect or the Xtion Pro Live. To that end, we propose to use curve-skeletons, a compressed representation of the 3-D object. Curve-skeletons are extremely useful in computer graphics and increasingly being used in computer vision for their valuable aid to address many visualization tasks including shape analysis, animation, morphing and/or shape retrieval. Indeed, curve-skeletons represent a simplified version of the 3-D object with a major advantage of preserving both its geometry and topology information. This, in turn, allows to estimate the object configuration or object pose after approximating its skeletal structure.

The remainder of the paper is organized as follows: Section 2 covers the literature review on the most recent curve-skeleton extraction approaches for point cloud datasets. Section 3 introduces our real-time curve-skeleton extraction approach. In Section 4 we evaluate and analyze the resulting curve-skeletons from multiple scanned 3-D models. To do so, both real and synthetic data have been considered. Finally, concluding remarks are given in Section 5.

2 RELATED WORK

Curve-skeleton makes decisive contribution for human pose estimation as it enables to estimate the body configuration by fitting the underlying skeletal structure. Consequently, an extensive research can be found in the literature, with many approaches strongly dependent on the requirements of their applications. In the following we only review the most representative methods for curve-skeleton extraction from point cloud datasets. For a complete review, we refer the reader to the comprehensive survey of Cornea et al. (Cornea et al., 2007).

Skeletonization algorithms can be divided in four different classes: thinning and boundary propagation, distance field-based, geometric, and general/field functions (Cornea et al., 2007). However, recent approaches are providing excellent results by combining different techniques from different classes (Cao et al., 2010) (Sam et al., 2012) (Tagliasacchi et al., 2009). Au et al. (Au et al., 2008) proposed a curve-skeleton extraction approach by mesh contraction using Lapla-

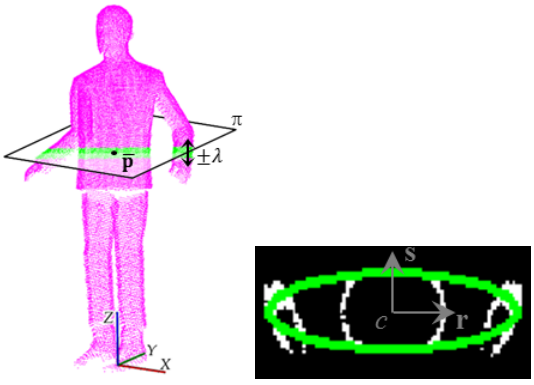
Garcia F. and Ottersten B..
Real-time Curve-skeleton Extraction of Human-scanned Point Clouds - Application in Upright Human Pose Estimation.
DOI: 10.5220/0005298300540060
In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (VISAPP-2015), pages 54-60
ISBN: 978-989-758-090-1

cian smoothing. They used connectivity surgery to preserve the original topology and an iterative process as an energy minimization problem with contraction and attraction terms. Though special attention must be paid to the contraction parameters, the method fails in the case of very coarse models. Watertight meshes are required due to the mesh connectivity constraint from the Laplacian smoothing. Cao et al. (Cao et al., 2010) extended their work to point cloud datasets. To do so, a local one-ring connectivity of point neighborhood must be done for the Laplacian operation, which significantly increases the processing time. An alternative approach proposed by Tagliasacchi et al. (Tagliasacchi et al., 2009) uses recursive planar cuts and local rotational symmetric axis (ROSA) to extract the curve-skeleton from incomplete point clouds. However, their approach requires special attention within object joints and it is not generalizable to all shapes. Similarly, Sam et al. (Sam et al., 2012) use the cutting plane idea but only two anchors points must be computed. They guarantee centeredness by relocating skeletal nodes using ellipse fitting. However, although the resulting curve-skeletons from this work preserve most of the required properties cited by Cornea et al. (Cornea et al., 2007), they are impractical when real-time is required. We herein overcome this limitation by computing the skeletal candidates in the 2-D space.

3 PROPOSED APPROACH

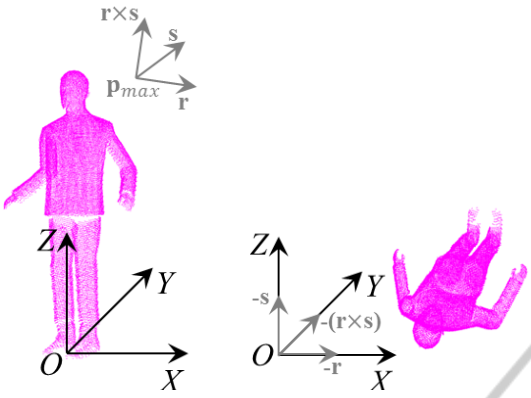
In the following we introduce our approach to extract the curve-skeleton from upright human-scanned point clouds. Our major contribution, in which we address the real-time constraint, is based on the extraction of the skeletal candidates in the 2-D space. We have been inspired by image processing techniques used in silhouette-based human pose estimation (Li et al., 2009). The final 3-D curve-skeleton results from back projecting these 2-D skeletal candidates to their right location in the 3-D space.

Let us consider a scanned point cloud $\mathcal{P}$ represented in a three-dimensional Euclidean space $\mathbb{E}^3 \equiv \{\mathbf{p}(x,y,z) | 1 \leq x \leq X, 1 \leq y \leq Y, 1 \leq z \leq Z\}$, and describing a set of 3-D points $\mathbf{p}_i$ representing the underlying external surface of an upright human body. The first step concerns the voxelization of the given point cloud to account for the point redundancy resulting from registering multi-view depth data. To that end, we have considered the surface voxelization approach presented in (Garcia and Ottersten, 2014). The resulting point cloud $\mathcal{P}'$ presents both a uniform point density and a significantly reduction of the points to be further processed. Our goal is to build a 2-D image representing the front view of the 3-D body in order to extract its curve-skeleton using 2-D image processing techniques. To do so, we define a cutting-plane π being parallel to x and y-axis of $\mathbb{E}^3$ and intersecting $\mathcal{P}'$ at $\bar{\mathbf{p}}$, the mean point or centroid of the point set $\mathcal{P}'$, *i.e.,* $\bar{\mathbf{p}} = \frac{1}{k} \cdot \sum_{i=1}^{k} \mathbf{p}_i$. From our experimental results, π crossing $\bar{\mathbf{p}}$ provides good body orientations in the case of upright body postures, *e.g.,* walking, running or working. Alternative body configurations are discussed in Section 4.1. We determine the body orientation by fitting a 2-D ellipse onto the set of 3-D points lying on π, *i.e.,* $\forall \mathbf{p}_i | \mathbf{p}_i(z) = \bar{\mathbf{p}}(z)$. We note that in practice and due to point density variations, 3-D points are not necessarily lying on π. Therefore, we consider those 3-D points in $\mathcal{P}'$ with $\mathbf{p}_i(z) \in [\bar{\mathbf{p}}(z) \pm \lambda]$, and λ being related to the point density of $\mathcal{P}'$, as depicted in Fig. 1a. We note that λ has to be chosen large enough to ensure a good description of the body contour. In turn, the higher the point density the smaller the λ value should be. Fig. 1b illustrates the resulting 2-D ellipse using the Fitzgibbon et al. (Fitzgibbon and Fisher, 1995) approach, implemented in OpenCV (Bradski and Kaehler, 2008). $\mathbf{r}$ and $\mathbf{s}$ are unitary vectors along the major and minor ellipse axes, respectively. c is the ellipse centroid. Once the body orientation is known, we build the 2-D image $\mathbf{I}$ representing the front view of the human body. To do so, we can project $\mathcal{P}'$ to the 2-D plane π' defined by the ellipse major axis $\mathbf{r}$ and with normal vector $\mathbf{s}$. Image dimensions are given by the projections of the minimum $\mathbf{p}_{min} = (x_{min}, y_{min}, z_{min})$ and maximum $\mathbf{p}_{max} = (x_{max}, y_{max}, z_{max})$ 3-D point coordinates from $\mathcal{P}'$. An alternative solution is to translate $\mathcal{P}'$ from the coordinate frame $\mathcal{C}_1 = (\mathbf{r}, \mathbf{s}, \mathbf{r} \times \mathbf{s})$, and origin at $\mathbf{p}_{max}$, depicted in Fig. 2a, to the coordinate frame $\mathcal{C}_2 = (-\mathbf{r}, -(\mathbf{r} \times \mathbf{s}), -\mathbf{s})$, and origin at $(0,0,0)$, de-



(a) (b)

Figure 1: (a) Selected 3-D points (in green color) for ellipse fitting. (b) Fitted ellipse using OpenCV (Bradski and Kaehler, 2008).



(a) $\mathcal{P}'$ (b) $\mathcal{P}''$

Figure 2: (a) $\mathcal{P}'$ at $\mathcal{C}_1$. (b) $\mathcal{P}''$ at $\mathcal{C}_2$.

picted in Fig. 2b, *i.e.,* $\mathcal{P}'' = \mathbf{T}_{\mathcal{C}_1}^{\mathcal{C}_2} \cdot \mathcal{P}'$. By doing so, the 2-D plane π' coincides with the two-dimensional Euclidean space with x and y axes coincident to $\mathbb{E}^3$, *i.e.,* $\mathbb{E}^2 \equiv \{\mathbf{I}(m,n) | 1 \leq m \leq M, 1 \leq n \leq N\}$. Therefore,

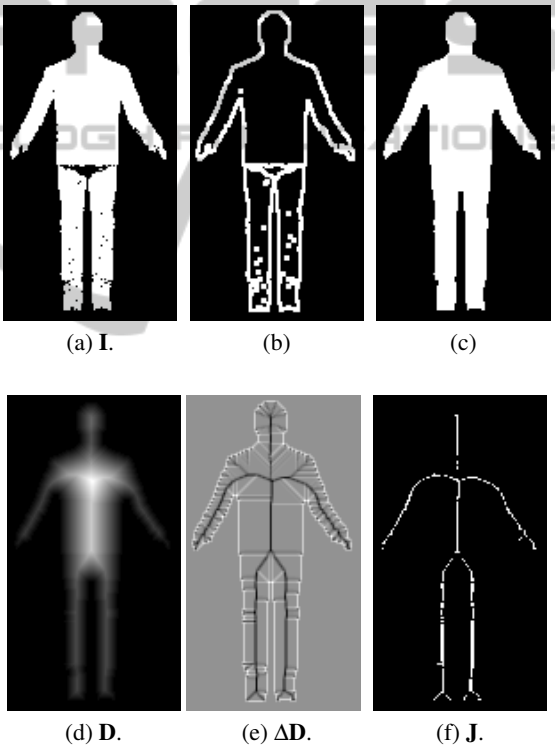
$$\mathbf{I}(m,n) = \begin{cases} 255 & \text{if } \mathbf{p}_i \in \mathcal{P}'' \quad \forall_i \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

with $m = \delta \cdot round\big(\mathbf{p}_i(x)\big)$ and $n = \delta \cdot round\big(\mathbf{p}_i(y)\big)$, and δ the conversion factor between metric units and pixels (herein we set $\delta = 100$, *i.e.,* 1 mm = 1 pixel). Fig. 3a shows the resulting image $\mathbf{I}$ after projecting $\mathcal{P}''$ onto $\mathbb{E}^2$.

In 2-D image processing, distance field or distance transform (DT) is commonly used to build image maps $\mathbf{D}$ from binary images. Pixel values indicate the minimum distance between the selected pixel and its nearest boundary pixel, *i.e.,* the closest zero pixel. That is, $\mathbf{D}(\mathbf{u}) = \min\{d(\mathbf{u},\mathbf{v}) | \mathbf{I}(\mathbf{v}) = 0\}$, with $d(\mathbf{u},\mathbf{v})$ being the Euclidean, Manhattan or Chessboard distance metric. We note that the resulting image map is highly sensitive to zeros pixels, and thus holes in the image will significantly alter the resulting image map. A valid approach to fill the body silhouette in the case of a constant density of points and without omission of data is the use of morphological operators such as dilation and erosion operators. However, filling the body silhouette becomes more intricate when treating incomplete point clouds with large portions of data missing, *e.g.,* around the pelvis in Fig. 3a. We therefore propose to extract the contours of $\mathbf{I}$ and fill the area bounded by the most external contour, *i.e.,* the body silhouette, which results in a dense body silhouette, as shown in Fig. 3c. To do so, we use the contour following algorithm proposed by Suzuki et al. (Suzuki and Abe, 1985) (see Fig. 3b). Among the aforementioned distance metrics, we herein have considered the Euclidean DT described in (Felzenszwalb and Huttenlocher, 2004). From the resulting image map $\mathbf{D}$ in Fig. 3d, we realize that higher-valued map pixels correspond to centered pixels within the silhouette boundary and thus, skeletal candidates. Indeed, skeletal candidates coincide with low-valued edges on the result of the Laplace operator on $\mathbf{D}$, *i.e.,* $\Delta\mathbf{D} = \partial^2\mathbf{D}/\partial^2 m + \partial^2\mathbf{D}/\partial^2 n$, as can be observed in Fig. 3e. The final 2-D coordinates of the skeletal candidates $\mathbf{J}$ are given by an adaptive thresholding on the low-valued edges in $\Delta\mathbf{D}$, *i.e.,*

$$\mathbf{J}(\mathbf{u}) = \begin{cases} 255 & \text{if } \Delta\mathbf{D}(\mathbf{u}) < \tau \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

being τ the adaptive threshold value. A valid τ value can be automatically obtained using Otsu's method (Sezgin and Sankur, 2004). The pixel positions of the skeletal candidates from (2) correspond to the x and y coordinates of the 3-D points $\mathbf{q}_i$ that will constitute the final curve-skeleton $\mathcal{J}$, as depicted in Fig 4, *i.e.,* $\mathbf{q}_i(x) = m$ and $\mathbf{q}_i(y) = n \quad \forall \mathbf{J}(m,n) = 255$. We determine the missing $\mathbf{q}_i(z)$ coordinate for each skeletal candidate $\mathbf{J}(\mathbf{u}_i)$ by defining the line l_i with unit vector the z-axis of $\mathbb{E}^3$ and intersecting at the corresponding skeletal candidate $\mathbf{J}(\mathbf{u}_i)$. The missing coordinate corresponds to the middle point between the



(a) $\mathbf{I}$. (b) (c)

(d) $\mathbf{D}$. (e) $\Delta\mathbf{D}$. (f) $\mathbf{J}$.

Figure 3: (a) $\mathbf{I}$ from (1). (b) Contours of $\mathbf{I}$. (c) Body silhouette. (d) Euclidean DT $\mathbf{D}$. (e) Laplace operator $\Delta\mathbf{D}$. (f) $\mathbf{J}$ from (2).

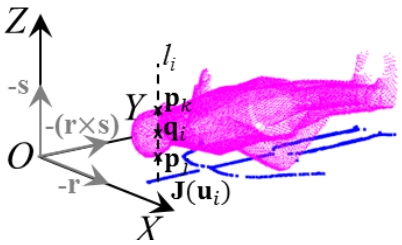


Figure 4: (a) Skeletal candidates **J** (in blue color) and the location of a skeletal candidate $\mathbf{J}(\mathbf{u}_i)$ in the 3-D space, *i.e.,* $\mathbf{q}_i(z)$.

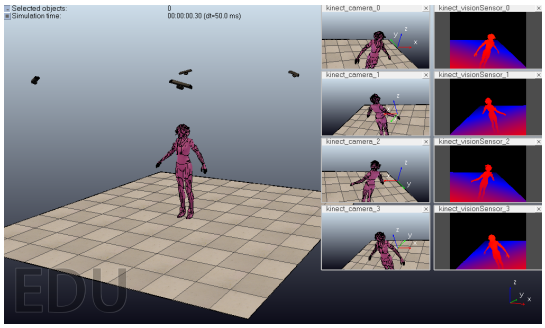


Figure 5: Experimental environment using V-Rep.

lower $\mathbf{p}_j$ and upper $\mathbf{p}_k$ intersected points on the external contour of the selected slice, also computed using the contour following algorithm in (Suzuki and Abe, 1985), *i.e.,* $\mathbf{q}_i(z) = \mathbf{p}_j(z) + \big(\mathbf{p}_k(z) - \mathbf{p}_j(z)\big)/2$. The final 3-D curve-skeleton $\mathcal{J}$ results from projecting back the 3-D points $\mathbf{q}_i$ from $\mathcal{C}_2$ to $\mathcal{C}_1$, *i.e.,* $\mathcal{J} = \mathbf{T}_{\mathcal{C}_2}^{\mathcal{C}_1} \cdot \mathcal{J} = (\mathbf{T}_{\mathcal{C}_1}^{\mathcal{C}_2})^{-1} \cdot \mathcal{J}$.

4 EXPERIMENTAL RESULTS

In the following we evaluate the proposed curve-skeleton extraction approach on both real and synthetic data. In addition, we present a visual comparison against the curve-skeleton extraction via Laplacian-based contraction approach presented in (Cao et al., 2010), and the runtime analysis. 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.

Four different 3-D models have been considered to evaluate the proposed curve-skeleton extraction approach, *i.e.,* the standing Bill from the V-Rep repository (VRE, 2014), and the Nilin Combat, Iron Man and Minions, from the publicly available TF3DM™ repository (TF3, 2014). The vast majority of available 3-D models are built as textured polygonal models, being flexible and easily to be rendered by computers. However, the number of polygons is far below the number of 3-D points of a real scanned object; mainly within regions with uniform geometry. Consequently, we have transformed the representation of the considered 3-D models from polygonal meshes to real 3-D point sets. To do so, we have simulated a multi-view sensing system composed of 4 RGB-D cameras using V-Rep (VRE, 2014), a very versatile robot simulator tool in which the user can replicate real scenarios, as shown in Fig. 5. Synthetic data results from the registration of the depth data acquired by each virtual RGB-D camera.

Fig. 6 shows the scanned 3-D models (in purple color), after data registration and surface voxelization, and their respective extracted curve-skeletons. We note that the resulting curve-skeletons present the same topology and geometry as the 3-D object; being unit wide and centrally placed within the underlying scanned surface.

We show, in Fig. 7, a visual comparison against Cao et al. approach (Cao et al., 2010). One of the limitations of Cao et al. approach is the need of 1-ring neighborhood to perform the Laplacian-based contraction. As a result, their approach fails in the presence of large portions of data missing, as occurs around the pelvis in Fig. 7c and Fig. 7f. Furthermore, contraction and attraction weights must be manually tunned in order to obtain a unit thickness curve-skeleton. Herein, we have considered the default settings provided by the authors. We also note that the consumption time to extract each curve-skeleton is above 1 minute.

Table 1: Running time to extract the curve-skeleton of the 3-D models presented in Fig. 6 (units are in ms).

3-D Model	Running time					
	Orientation	Silhouette	**D**	**ΔD**	$\mathbf{p}_i(z)$	Total
Standing Bill (23650 points)	2.5	0.2	0.3	0.0	193.0	252.2
Nilin Combat (18730 points)	1.6	0.0	0.1	0.1	159.6	206.0
Iron Man (33358 points)	2.7	0.1	0.4	0.2	390.6	473.0
Minions (18426 points)	1.5	0.3	0.1	0.2	105.4	150.8
Man 1 (18053 points)	1.3	0.7	0.3	0.1	232.7	279.5
Man 2 (21263 points)	0.9	0.4	0.4	0.1	287.5	341.6
Woman 1 (12986 points)	0.4	0.1	0.3	0.1	109.3	142.1

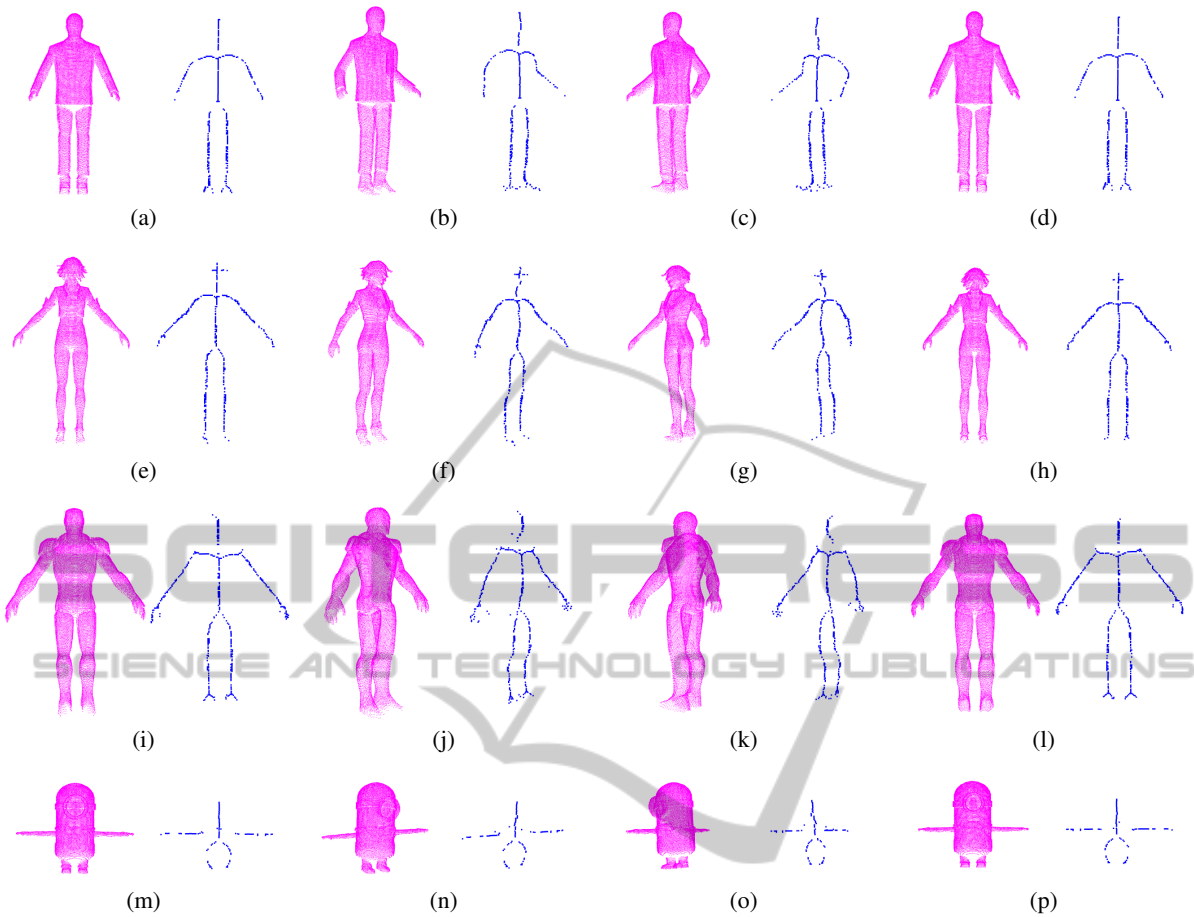


Figure 6: Curve-skeleton extraction from synthetic object-scanned point clouds. 1^{st} row, Standing Bill (23650 points). 2^{nd} row, Nilin Combat (18730 points). 3^{rd} row, Iron Man (33358 points). 4^{rd} row, Minions (18426 points).

Real data has been generated from a multi-view sensing system composed of 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. The relationship between the two cameras was determined using the stereo calibration implementation available in OpenCV (Bradski and Kaehler, 2008) and a transparent checkerboard (black squares are visible from both sides). Better registration approaches based on ICP, bundle adjustment or the combination of both can also be considered. However, it is shown in Fig. 8 that the current approach perfectly extracts the curve-skeleton on such a coarse registered point clouds, handling large portions of missing data as well as registration inaccuracies. Fig. 8 presents 3 coarse registered point clouds of two mans and one woman. Note that despite the large portions of data missing due to self occlusions and inaccurate registration of both views, our approach is able to extract an accurate curve-skeleton that preserves both topology and geometry of the scanned model.

Table 1 reports the running time to extract the curve-skeleton of the 3-D models presented in Fig. 6 and Fig. 8. We note that we have reported CPU-based values without using data-parallel algorithms or graphics hardware (GPU). Most of consumption time is invested on determining the missing 3-D coordinate of each skeletal candidate whereas 2-D processing time is almost negligible, as it was expected.

4.1 Limitations and Future Work

The resulting curve-skeleton depends strongly on the DT computed from the front view of the human body. Therefore, occluded body parts due to crossing arms or legs will not be considered and can provide wrong skeletal candidates. Furthermore, having one or both arms too close to the torso will make DT to consider them as part of the torso. Similarly, if both legs are too close, they can be considered as a single one. We plan to solve these body configurations by using both temporal and RGB information. Indeed, we herein pro-

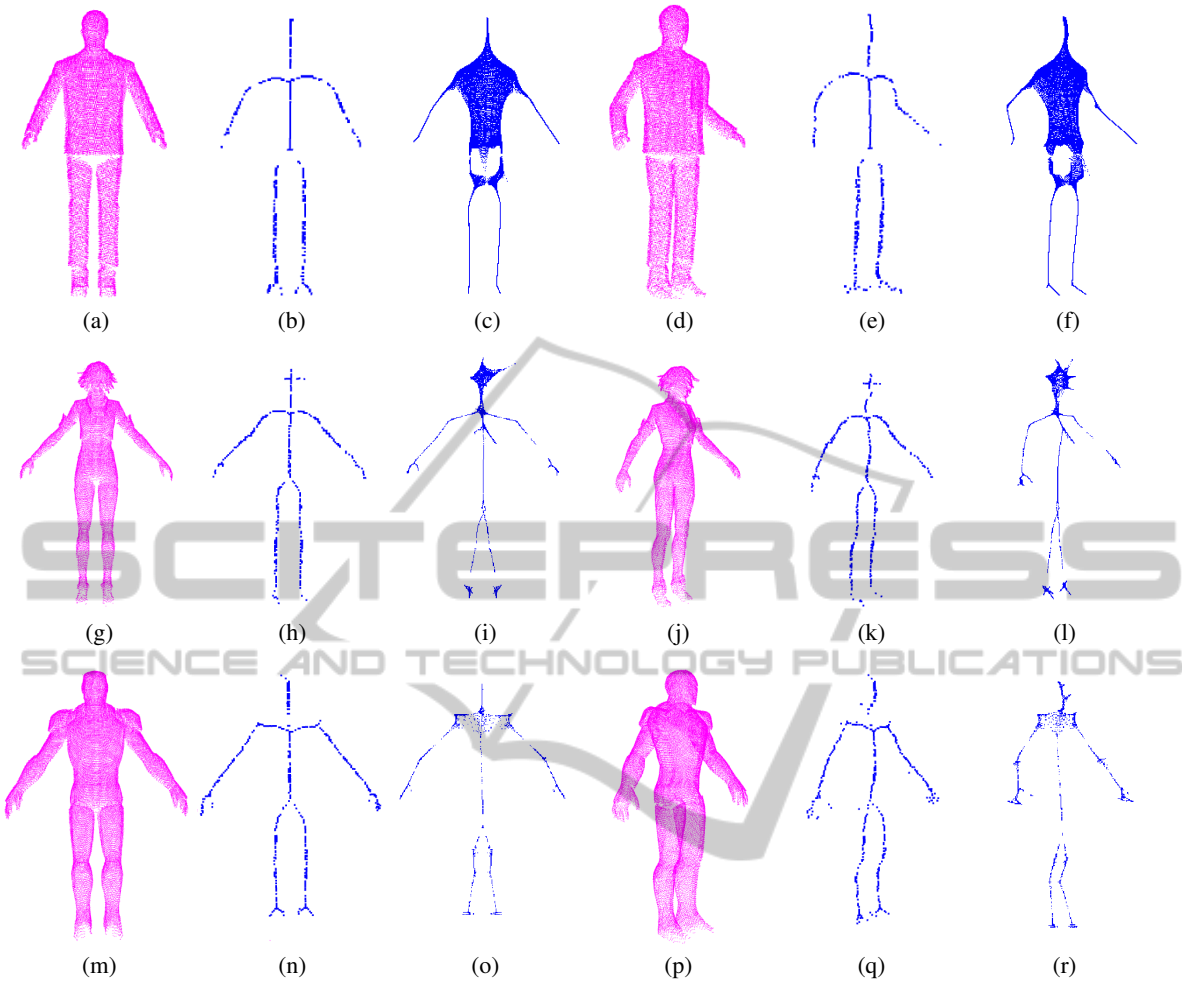


Figure 7: Visual comparison against curve-skeleton extraction via Laplacian-based contraction (Cao et al., 2010). 1^{st} row, Standing Bill. 2^{nd} row, Nilin Combat. 3^{rd} row, Iron Man. 1^{st} and 4^{th} col., Input dataset. 2^{nd} and 5^{th} col. Resulting curve-skeleton using our approach. 3^{nd} and 6^{th} col. Resulting curve-skeleton using (Cao et al., 2010).

pose a very fast approach to estimate a coarse curve-skeleton that will be the basis to fit a human model skeleton. We plan to use a progressive fitting of the skeleton model starting from the torso limb. In general, the torso limb corresponds to the lowest-valued edges from the Laplace operation output.

5 CONCLUDING REMARKS

A real-time approach to extract the curve-skeleton of an upright human-scanned point cloud has been described. Our main contribution is in estimating the 2-D skeleton candidates using image processing techniques. By doing so, we address the real-time constraint. The final curve-skeleton results from locating the previous computed 2-D skeleton candidates in the 3-D space. The resulting curve-skeleton preserves the centeredness property as well as the same topology and geometry as the 3-D model. Future work includes the treatment of alternative body configurations than upright. The use of temporal and RGB information will be also investigated in order to increase the robustness of the approach.

ACKNOWLEDGEMENTS

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

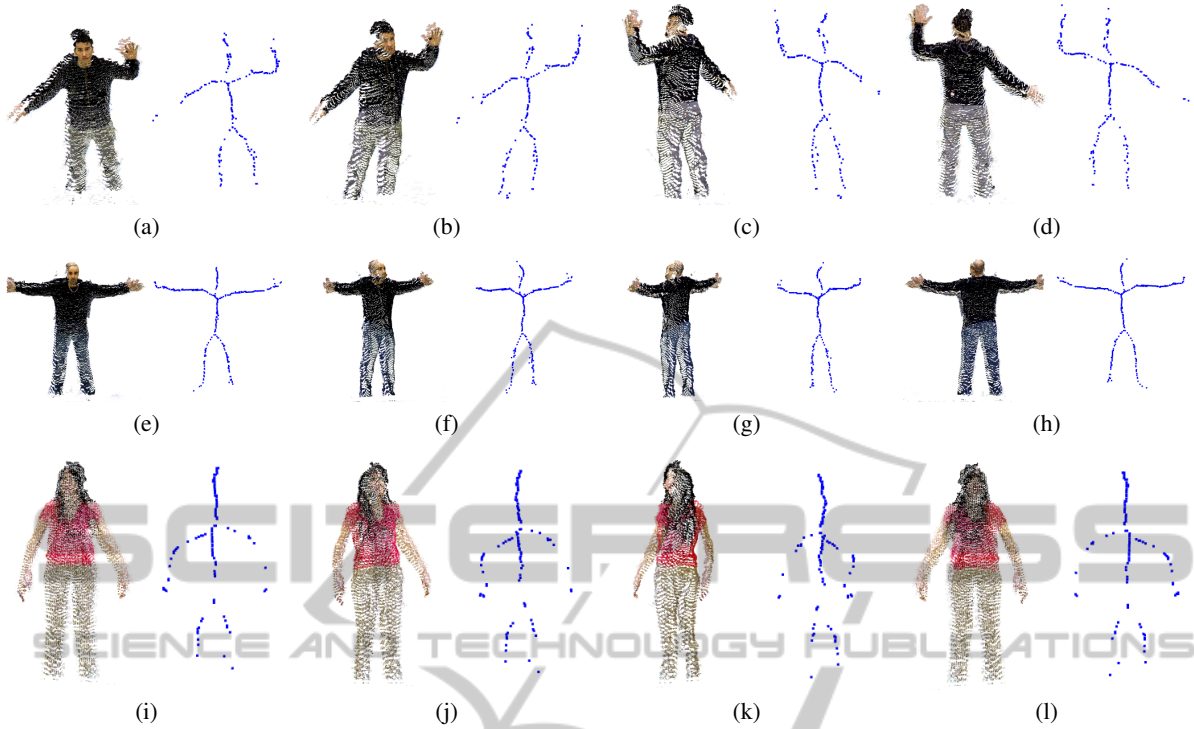


Figure 8: Curve-skeleton extraction from upright human-scanned point clouds. 1^{st} row, Man 1 (18053 points). 2^{nd} row, Man 2 (21263 points). 3^{rd} row, Woman 1 (12986 points).

REFERENCES

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

(2014). TF3DM^{TM}. http://tf3dm.com/.

(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., Tagliasacchi, A., Olson, M., Zhang, H., and Su, Z. (2010). Point Cloud Skeletons via Laplacian Based Contraction. In *Shape Modeling International Conference (SMI)*, pages 187–197.

Cornea, N., Silver, D., and Min, P. (2007). Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):530–548.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science.

Fitzgibbon, A. and Fisher, R. B. (1995). A buyer's guide to conic fitting. In *In British Machine Vision Conference*, pages 513–522.

Garcia, F. and Ottersten, B. (2014). CPU-Based Real-Time Surface and Solid Voxelization for Incomplete Point Cloud. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 2757–2762.

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.

Sezgin, M. and Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168.

Suzuki, S. and Abe, K. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46.

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.