Journal of Advanced Computer Science and Technology Research, Vol 8, No 3, September 2018 , 111 - 124 .

DSR Design for Scientific RENAISSANCE

JACSTR

Slinky-based Segmentation on 3D Polygonal Model for Functional Parts in Computer Graphics

Kok-Why Ng and Junaidi Bin Abdullah
Faculty of Computing and Informatics,
Multimedia University (MMU),
Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia.
kwng@mmu.edu.my; junaidi@mmu.edu.my

Article Info

Received: 1 August 2018
Accepted: 1 September 2018
Published online: 30 September 2018

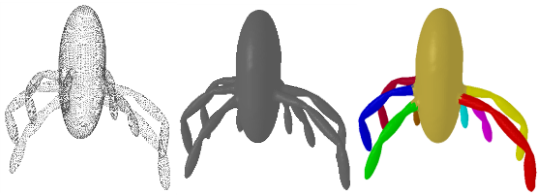
Abstract Segmenting three-dimensional (3D) polygonal model into meaningful functional parts is greatly debated in year 2010. Yet, there is no robust segmentation algorithm as to satisfactorily match the result to the human desired segments. This paper breaks the model into human definable functional parts (or features). The proposed method first shrinks the model into triangular-skeleton by Laplacian-based contraction method. The edge-collapsed and vertex-merging of simplification are applied to turn the skeleton into single-connected edges and merge the linear skeleton nodes for fast processing respectively. Slinky-based segmentation method is applied to investigate the contour and segment both the periphery and centered features. The generated result is compared to the seven well-known segmentation methods and the results are consistent and stable throughout all the models.

Keywords: **Segmentation, Clustering; Pattern Recognition; Functional Parts; Computer Graphics.**

1. Introduction Two-dimensional (2D) segmentation is a key process in the Imaging and Computer Vision research fields. Three-dimensional (3D) segmentation is an extension of 2D segmentation to tackle with volumetric models. This process receives great attention from researchers around the world to strike for producing accurate and consistent human definable segmented features.

An unorganized geometric model data demonstrates only the structure of a model (e.g. Fig. 1a and 1b). It does not display any direct connotation to human eyes. Segmented model classifies the raw data into related functional parts (Fig. 1c) and presents meaningful detail to human eyes.

Various segmentation methods have been proposed such as region grow, multiple source region grow, water-shed, hierarchical clustering, top-down hierarchical, iterative clustering, spectral analysis methods, implicit methods, graph-cut, skeleton method and etc. This paper will discuss seven popular segmentation methods (K-Means (Shlafman et al., 2002), Core Extraction (Katz et al., 2005), Fitting Primitives (Attene et al., 2006), Random Walk (Lai et al., 2008), Randomized Cut (Golovinskiy et al., 2008), Normalized Cut (Golovinskiy et al., 2008) and Shape Diameter Function (Shapira et al., 2008) and will compare them with the proposed method.



(a) (b) (c)

Figure 1: Octopus model in (a) point cloud, (b) skin and (c) segmented format.

Next section will briefly discuss the seven methods and their strengths and weaknesses. Section 3 will present the proposed algorithm to consistently segment the given model into meaningful functional parts. Section 4 will discuss the proposed slinky-based segmentation metrics. Section 5 will show the results and comparisons between the seven exiting methods and the proposed method. Last section will conclude this paper.

2. Seven Popular Segmentation Methods

K-Means was proposed by Shlafman et al. (2002) to create segmentation on two models for morphing purpose. A number of segments (say k) were first specified by the user from the model. The algorithm was then to create k seed faces of clusters, and assigned the adjacent faces to each associated cluster. The cluster assignment was judged on two thresholds: (1.) the difference in the dihedral angle α between two surfaces f_1 and f_2, and (2.) the sum of geodesic distance from the centroid of each face to the center of their shared edge, PhysDist (f_1, f_2). The segmentation process was iterated until all the faces were assigned. The k seed faces were then re-computed and re-located to the center of the faces in each cluster. This method requires user-input for the number of segmentation. The algorithm will not always produce accurate user-desired segmented boundary because the re-location of the seed faces does not always take into account of any borderline measure or mesh connectivity for the boundary feature. Though, this method till now is still very popular to be employed by many researchers (Zhang et al., 2016; Solaiyappan et al., 2017).

Core Extraction was proposed by Katz et al. (2005) to hierarchically segment a model without being sensitive to the model pose and sizes. They first transformed the model vertices into an insensitive representation using multidimensional scaling (MDS). All the folded parts of a model were to be straightened. The prominent feature points were extracted from the canonical mesh based on the global conditions. Subsequently, the spherical mirroring was applied to extract the core components. At the end, the boundaries were refined by following the natural seams of the model vertices. Generally, this method produces a reasonable segmented result on models with obvious curvature surfaces and unconnected functional features (Shu et al., 2016). The initial transformation of the model by MDS process plays a very essential role. The straightened parts (or simplified parts) are claimed that will produce a pose-invariant representation. This will happen only if the topology of the new pose of the model is unchanged. The result after MDS process is believed to produce different result if the new pose is twisted or stretched. Therefore, the method may not always generate correct number of segmentation if the MDS process is not carefully handled.

Fitting Primitives was proposed by Attene et al. (2006) to apply multiple primitive objects to fit into a model. A binary tree was created with each triangle represented as an independent cluster filling up the lowest level of the tree. They fitted and tested each primitive to every triangle in separate segment hierarchically. The geometric primitive such as plane, cylinder or sphere were computed to find the best fitting primitive to approximate the faces in every pair of adjacent segments. Generally, a model with parts closely identical to plane, cylinder or sphere will accurately be clustered and segmented (Grilli et al., 2017). However, this method hardly segments the model with blending shapes or non-identical to any of the primitive objects.

Random Walk method was extended by Lai et al. (2008) from previous works (Grady, 2006; Sun et al., 2007) which were applied in image segmentation. The method reversed the approach of K-means method discussed in previous section. User would need to assign n faces as a seed in the initial step. This number was also the desired number of segments by the user in the final stage. The placement of the seeds was one seed per segment. A probability was computed on every edge of each non-seeded surface. A random walk would advance to each particular edge to the corresponding neighbor. Any surface with higher probability of reaching the initial seed would be clustered to the region associated to the seed. (Lai et al., 2008) had attempted the segmentation automatically and claimed that their algorithm would result more segmentations than desired.

Golovinskiy et al. (2008) made used of the idea proposed by (Karger et al., 1996) to find the most appropriate segment-able boundary for shape analysis. Karger et al. applied iterative edge contraction method to generate multiple set of randomized cuts. They computed the minimum cost among the cuts for the concave boundary. Golovinskiy et al. applied randomized cuts to create a partition function and produced a set of scored cuts to determine the most consistent cuts overlapped with the others in the randomized set. They built a dual graph from the input mesh, which the nodes were corresponding to the surfaces and the arcs are corresponding to edges. A traversal cost and a cut cost were computed on each arc. A

randomized set of K-way segmentations were then generated. This step outputted a scored set of segmentations that indicated the accuracy of the segmentations between two features. A partition function for each edge was estimated by taking the length of an edge and was divided by the total weight of all the segmentations. At the end, a consistency score for each of the segmentation was generated by computing the weighted average of the partition function values along edges on the boundaries of the segments Benczúr et al. (2015). The randomized seeds in the initial stage (e.g. finding the boundary feature) are randomly assigned. This may lead to inaccurate result and required more process for improvement.

Golovinskiy et al. also proposed an extended Normalized Cut method previously proposed by Shi and Malik (Shi et al., 2000) for 2D image segmentation. They merged the adjacent segments based on the area-weighted Normalized Cut cost, which took the total from all the ratio of the segment's cut cost to its area, to avoid dependence on surface tessellation. The computed error from the area-normalized cut of the segments was then contracted to each graph arc, and each cost was further increased to the power of *1/r*, where *r* was a randomization parameter. When *r* was close to zero value, their algorithm produced a better segmented result. The authors claimed that at $r = 0.02$, it gave a realistic tradeoff between quality and sufficiency of randomness for the segmented model. However, this does not guarantee for models with little concave edges (Herouane et al., 2016).

Shape Diameter Function (SDF) was proposed by Shapira et al. (2008) to measure the diameter of an object through volume for segmentation and skeletonization. A cone was centered at the inward normal direction of any point on the mesh. Rays were created inside the cone to the opposite meshes. Any intersection between a point to the normal of the meshes with the angle difference of more than 90 degrees was ignored. The SDF at the point was then recorded as the weighted average of all the rays length fell within a threshold from the median of all lengths (Jia et al., 2014). The smooth surface is believed to be able to produce more accurate diameter of distance from the inward normal direction to the antipodal mesh point. The piecewise linear surface will be a tedious one. Thus, the weighted average is taken (Shapira et al., 2008) to soften the error. Though, it is not a robust method and the error may increase especially at the crooked features.

3. Proposed Algorithm

Our algorithm begins with the input 3D mesh that consists of only the vertices and triangles (refer to Fig. 2). They will be voxelized to get rid of the alien points or noisy data before proceeding to surface reconstruction. Different levels of voxelized hand model will be created and only the third voxelized hand and onward (refer to Fig.3) provide sufficient detail for the next process. Note that if the input is a regular mesh, user may skip the Voxelization and surface reconstruction process.

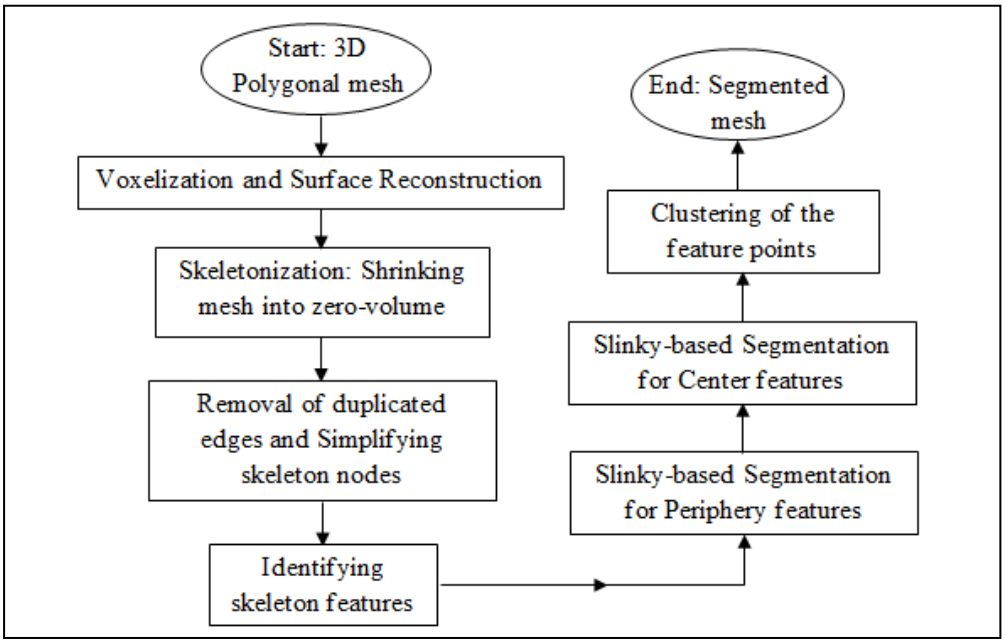


Figure 2: Proposed segmentation algorithm

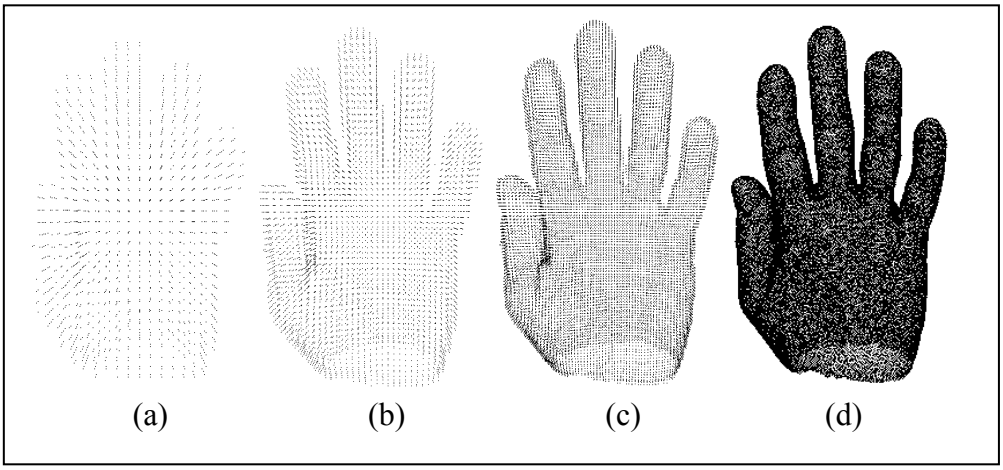


Figure 3: Different levels of voxelization output.

The mesh will then be skeletonized by turning the model into zero-volume of skeleton. This paper applied Laplacian-based contraction method [9] by shrinking the geometric contraction of the model. The proposed Laplace operator controls the diagonal weight matrices W_L and W_H to balance the contraction and attraction constraints respectively, so that

the skeleton will not be created outside the model mesh. However, the result of the shrinking has created many slim triangular skeletons and dense skeleton nodes. They need to be simplified so that the entire skeleton is simple, linear and non-cyclic graph exist. Vertex-merging of simplification process is proposed to merge one vertex (say *u*) to another vertex (say *v*) based on the shortest edge in a triangle. All the adjacent vertices of *u* will be connected to *v*. After the merging process, skeleton-node-removal process is applied to reduce the density of the skeleton nodes to ease the following segmentation process. The removal process will begin at the end node of each skeleton feature. The end node will be preserved so that it does not change the actual skeleton length.

Skeleton features identification is proposed to guide the segmentation process. Skeleton is another representation of an object shape in one-dimensional (1D) aspect. In Fig. 4, the skeleton of a hand model clearly summarizes the important features in 1D. The skeleton nodes at each feature are simplified to ease the proposed segmentation method. The skeleton will be dismantled at each skeleton joint, such as signposted at s_n and s_m in Fig. 4. The separated skeleton (or draft features) will be divided into three categories. The first category is the skeleton with fewer nodes. This category will set the skeleton to dummy as it does not contribute obvious functionality to human eyes. It is not denied that this is hard to tell the exact number of nodes that is not contributing to any functionality. It subjects to individual's perspective. The default number of skeleton nodes in this category is two or less than two, which is simply a maximum of two edges. Alternative, one can base on the length of the skeleton to justify the effectiveness of a skeleton-piece. This measure has to be proportional to the entire model size for its effectiveness.

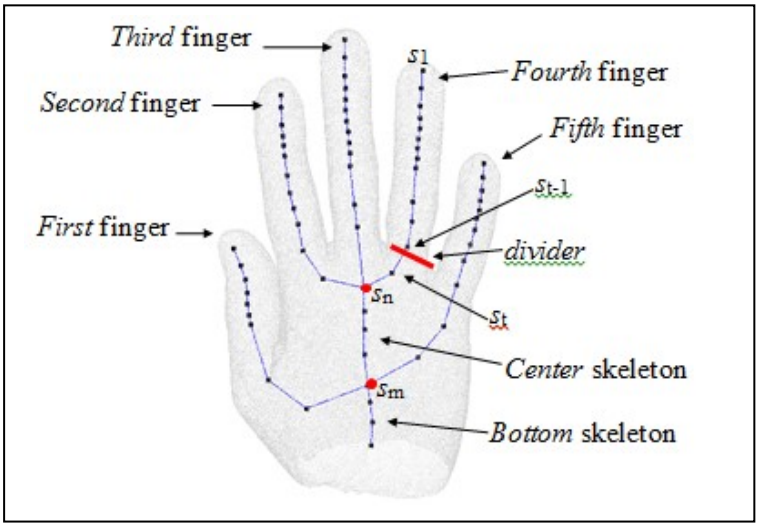


Figure 4: Skeleton of a hand model.

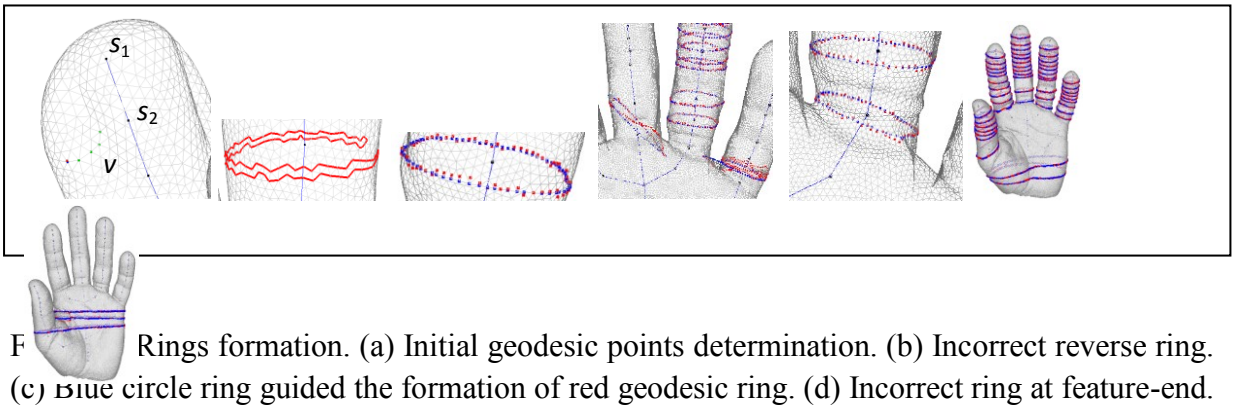
The second category is the skeleton with one end node to be connected with an edge; and the other end node is to be connected with more than two edges. This category consists of all

the periphery skeleton features as they are normally surrounding the center skeleton. For example, in Fig. 4, the skeleton from node s_1 till node s_n will fall into this category.

The third category is the skeleton with both end nodes to be connected with more than two edges. This category consists of all the center skeletons. For example, in Fig. 4, the skeleton from node s_n till node s_m will fall into this category. This category will most likely to merge with the model functional parts as many other skeletons are connecting to both the end nodes.

Among the three categories, only the second and third categories will be processed for different segmentations. Next section will discuss the proposed segmentation method on the periphery and center features, and the clustering of the feature points.

Slinky is a toy with pre-compressed helical spring (Wikipedia, 2015). Each ring transforms energy to next ring to move from top to bottom. This paper constructs rings (or contour) along each skeleton in second and third categories. Each ring will compute its circumference as a parameter. The parameter will be passed on to the next ring for comparison and average circumference finding. The comparison is to determine whether the following ring is more appropriate for different feature. In this paper, the default threshold for the comparison is 1.5 times of the previous ring size. This is an observed value from most of the experiment models. User may reduce the value to allow smaller changes or increase the value to cover more shapes in a feature. Ring is initially constructed at the second node s_2 of a skeleton (Fig. 5a). Nearest point v (in red color) from the model mesh is to be searched. The point is then made a copy and re-adjusted to a new position orthogonal to the skeleton, and to be a guide (in blue color) searching for the next point to form a complete ring.



F Rings formation. (a) Initial geodesic points determination. (b) Incorrect reverse ring. (c) Blue circle ring guided the formation of red geodesic ring. (d) Incorrect ring at feature-end. (e) Rectified the ring at feature-end. (f) Formation of all the rings at periphery features in a model. (g) Formation of rings at centered feature in a model.

The adjusted coordinate v' is given in Eq. 1:

$$v' = \begin{cases} v + dt * \hat{n}_s & \text{if } \theta = 90\text{-}\alpha > 0 \\ v - dt * \hat{n}_s & \text{if } \theta = 90\text{-}\alpha < 0 \\ v & \text{if } \theta = 0 \end{cases} \quad (1)$$

where the distant error $dt = d \sin \theta$, d is the distant from v to s_2, θ is the angle between point v to v' in the viewpoint of s_2, α is the angle between point v to the edge of s_1s_2, $\hat{n}_s$ is the unit skeleton vector at s_2.

The next adjacent point v'' is determined by computing the cost in Eq. 2.

$$\textbf{Cost } v'' = \min_{i \in v'_{adj}} (abs[\,90 - (\cos^{-1}(\hat{u}_i \cdot \hat{v}_s))\,]) \quad (2)$$

where v'_{adj} is the adjacent vertices of v', $\hat{u}_i$ is the unit vector to s_2, $\hat{v}_s$ is the unit skeleton vector. The minimum cost is sought so that the geometric modification is minimal. From third adjacent point and onwards, a prevention cost in Equation 3 is to be set up to prevent the reverse of the ring (as shown in Fig. 5b).

Prevention cost,

$$\mathbf{C_{pvt}} = \min_{i \in v_{2adj} \cap (\neg v_1)} (abs[\,180 - (\cos^{-1}(a_i \cdot b))\,]) \quad (3)$$

where $v_{2adj \cap (\neg v1)}$ is the adjacent points of v_2 but not including v_1, a_i is the adjacent edge to vertex v_2, b is the vector of v_1v_2.

With this, a complete ring is constructed (Fig. 5c). Another ring problem to solve is at feature-end (Fig. 5d), where the ring may include other unrelated feature or introduce duplicated rings. The previous computed circumference and average circumference are used to detect the end feature. If the created new circumference exceeds 1.5 times of the previous ring size or average circumference, it will stop constructing anymore ring. Interpolation will be applied to find the best fit ring (in Fig. 5e) in between the previous rings. The same computation costs are applied to all the other skeletons (Fig. 5f) in second category for the periphery features.

For the centered features (Fig. 5g), the same computation costs are to be utilized. This category is to be processed after the segmentation on periphery features is done, because the skeletons in this category are connected by many recognizable periphery features. If this is to be first processed, it will clinch up many other non-related periphery features together.

For clustering process, any points on the model mesh fall behind the last ring of each periphery feature will be clustered to the current feature. With this, there will be some un-clustered points in brown color (Fig. 6a). To solve this, the un-clustered points will be absorbed to the nearest centered feature. They will not be absorbed to the nearest periphery feature as the last ring in periphery feature has undoubtedly found the clear cut at its feature. The final result in the proposed method will output the segmented hand model as in Fig. 6b. However, due to the palm has two different colors which indicate two functional parts, this paper allows the user to combine the functional parts such as in Fig. 6c.

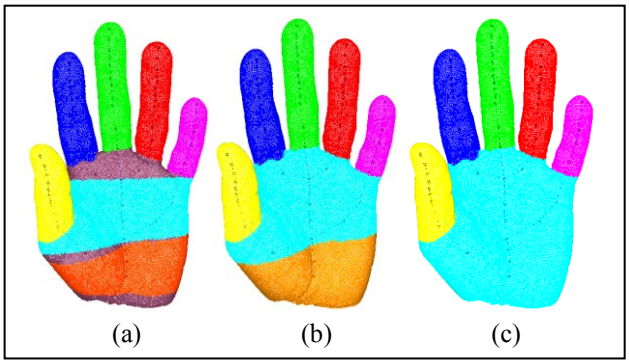


Figure 6: Segmented hand model. (a) Un-clustered brown color points. (b) Palm with two functional parts. (c) A complete clustered of points.

4. Results and Comparison

The proposed slinky-based segmentation (SB) method is computed in Intel® Core™i3 CPU, of 2 GB RAM, 64-bit Operating System. The input models are downloaded from Princeton university website. For fair comparison, this paper uses the results already generated by their original methods without re-implementing their methods.

Table 1 shows the number of vertices and faces for the experimented models. The differences in number of vertices or faces does not bring much significant impact to the segmentation process unless there are noises in the data. Though, these models are specifically selected for comparison because when they are segmented by the existing segmentation methods, their output results look so much different. Hence, it is good for comparison purpose.

Table 2 shows the number of segmented features by each segmentation method. Shape Diameter function always produces the highest number of segmented features; whereas Core Extraction method always produces the lowest number of segmented features. There is none existing definition to conclude that the number of segmented functional parts could determine the satisfactory to the human desired segments. In the table, one would say that only the Crooked hand makes obvious different in number of segmented functional parts.

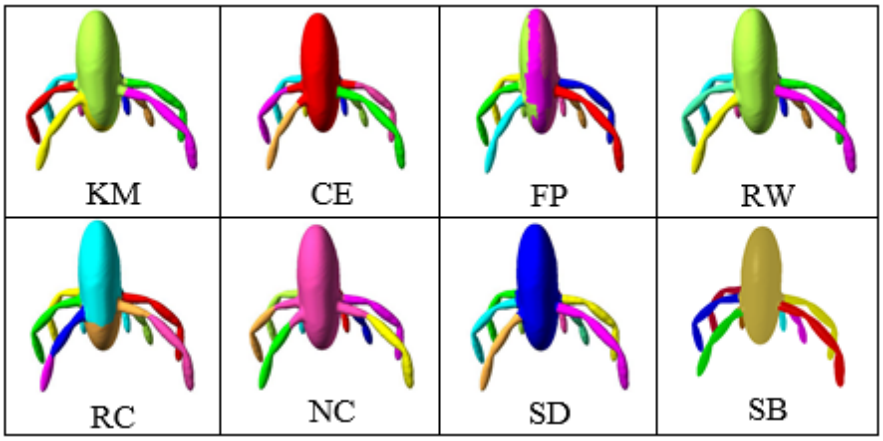
Tabel 1. Number of vertices and faces for each 3D model

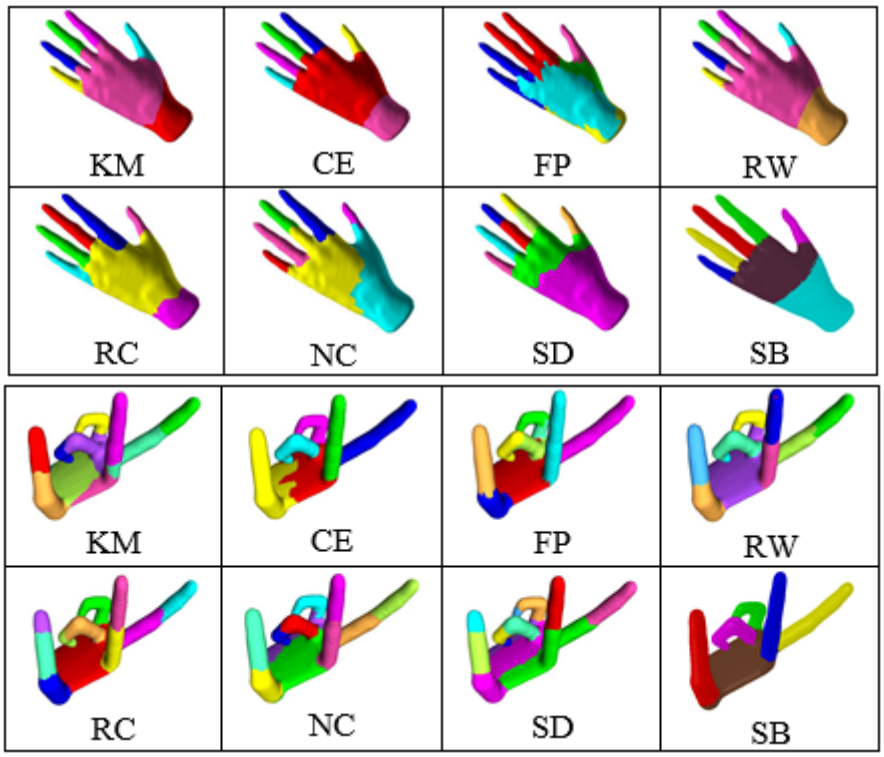
Model	No of vertices	No of faces
Octopus	15493	30982
Long hand	7242	14480
Crooked hand	7553	15102
Plier	4487	8970
Bird	7849	15694
Teddy bear	10096	20188

Table 2. Number of segmented functional parts for each 3D model. The abbreviation is given as KM (K-Means), CE (Core Extraction), FP (Fitting Primitive), RW (Random Walk), RC (Randomized Cut), NC (Normalized Cut), SD (Shape Diameter Function) AND SB (Slinky-Based) segmentation method

Model	KM	CE	FP	RW	RC	NC	SD	SB
Octopus	9	9	9	9	9	9	10	9
Long hand	7	7	7	7	7	7	9	7
Crooked hand	11	6	11	11	11	11	12	6
Plier	5	5	5	5	5	5	5	5
Bird	5	5	5	5	5	5	5	5
Teddy bear	8	6	8	8	8	8	9	7

Fig. 7 shows the visual results of the seven existing popular segmentation methods and the proposed slinky-based segmentation method. For octopus model, FP and RC produce ugly segmented results. For long hand model, except the proposed method, almost all the existing methods do not segment the fingers accurately. Some of the existing methods produce two or more colors in a single finger. For crooked hand model, KM, RW, RC, NC and SD contain mixture of colors in some of the fingers. In contrary, the proposed method produces excellence segmentation result. For plier model, FP, RW, RC, NC and SD produce two colors either at the peak or the handle of the plier. For the bird model, KM, FP and NC output two colors at the bird wings. For the teddy bear model, KM and NC show two colors at the body of the teddy bear.




KM
CE
FP
RW
RC
NC
SD
SB
KM
CE
FP
RW
RC
NC
SD
SB

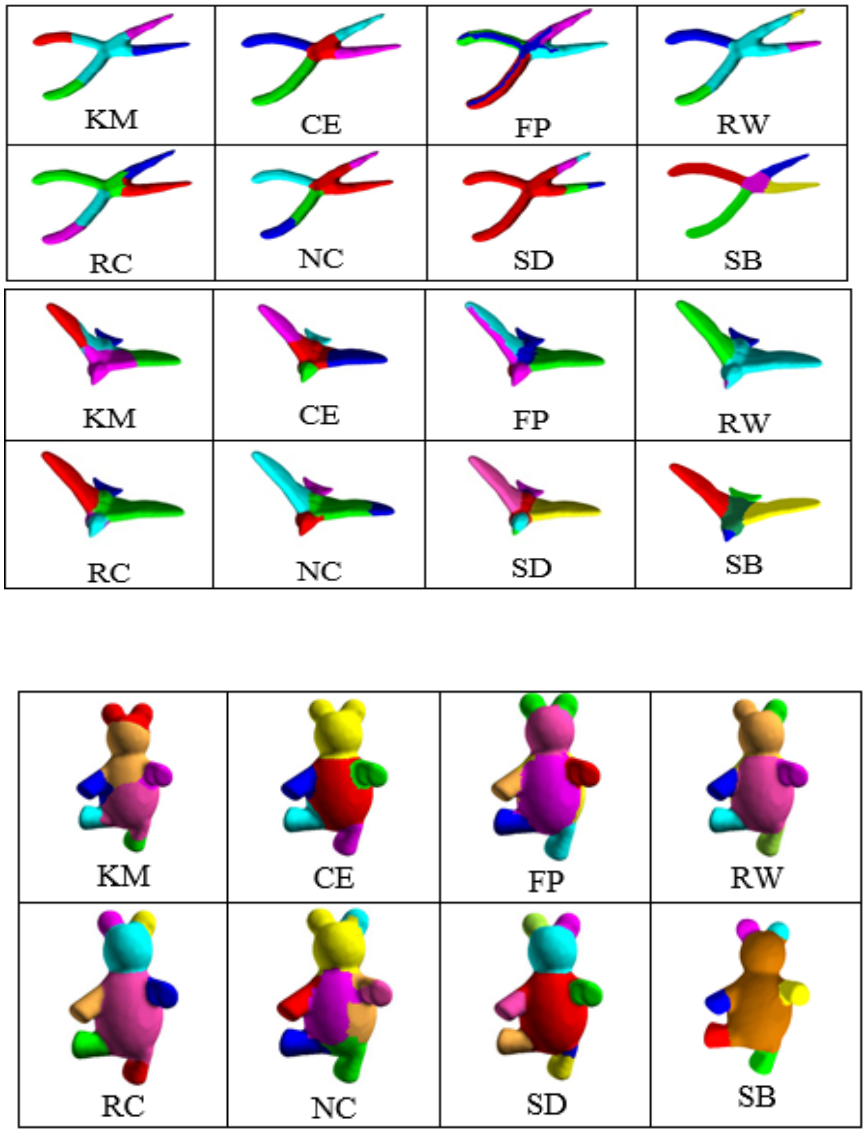


Figure 7: Visual results of the seven existing popular segmentation methods and the proposed slinky-based (SB) segmentation method.

5. Conclusions

Overall, the proposed slinky-based segmentation method has produced accurate and consistent functional parts. The results strictly follow the feature mesh definable by human eyes, regardless of the curvature shape of the model. This method is worth for a consideration to be extended to other research applications such as morphing and animation.

Acknowledgment Many thanks to MMU [Mini Fund, SAP ID: mmui/180152] for funding this project.

References

Attene M., Falcidieno B., and Spagnuolo M. (2006). Hirarchical mesh segmentation based on fitting primitive. Vis. Comput. 22(3), pp. 191-193.

Benczúr, A. A., & Karger, D. R. (2015) Randomized approximation schemes for cuts and flows in capacitated graphs. SIAM Journal on Computing, 44(2), pp. 290-319.

Cao J.J., Andrea T., Matt O., Hao Z., and Su. Z.X. (2010). Point Cloud Skeletons via Laplacian-Based Contraction. In: IEEE International Conference on Shape Modeling and Applications (SMI'2010), pp. 187 – 197.

Golovinskiy A., and Funkhouser T. (2008). Randomized cuts for 3D mesh analysis. ACM Transactions on Graphics (Proc. SIGGRAPH ASIA), 27(5).

Grady L. (2006). Random walks for image segmentation. IEEE Trans. Pattern Analysis and Machine Inbtelligence, 28(11), pp. 1768 – 1783.

Grilli, E., Menna, F., & Remondino, F. (2017). A Review of Point Clouds Segmentation and Classification Algorithms. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 339-344.

Herouane, O., Moumoun, L., Gadi, T., & Chahhou, M. (2016). An automatic framework for 3D objects-parts learning. In Information Science and Technology (CiSt), 4th IEEE International Colloquium on (pp. 481-486).

Jia, H., Geng, G. H., & Zhang, J. G. (2014). Consistent mesh segmentation based on shape diameter function and EM. In Advanced Materials Research (Vol. 1049, pp. 1417-1420).

Karger, D. R., and Stein, C. (1996). A new approach to the minimum cut problem. Journal of the ACM, 43(4), pp. 601–640.

Katz S., Leifman G., and Tal A. (2005). Mesh segmentation using feature point and core extraction. The Visual Computer, 21(8-10), pp. 649–658.

Lai Y.-K., Hu S.-M., Martin R. R., and Rosin P. L. (2008). Fast mesh segmentation using random walks. In Symposium on Solid and Physical Modeling, pp. 183–191.

Shapira L., Shamir A., and Cohen-Or D. (2008). Consistent mesh partitioning and skeletonisation using the shape diameter function. Vis. Comput. 24(4), pp. 249–259.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), pp. 888–905.

Shlafman S., Tal A., and Katz S. (2002). Metamorphosis of polyhedral surfaces using decomposition. Eurographics, pp. 219-228.

Shu, Z., Qi, C., Xin, S., Hu, C., Wang, L., Zhang, Y., & Liu, L. (2016). Unsupervised 3D shape segmentation and co-segmentation via deep learning. Computer Aided Geometric Design, 43, pp. 39-52.

Solaiyappan, C., & Prisilla, L. (2017). A Review on Clustering Techniques. International Journal of Computer Science, 5(1):19.

Sun, X., Rosin, P. L., Martin, R. R., and Langbein, F. C. (2007). Random walks for mesh denoising. In Proc. ACM Symposium on Solid and Physical Modeling, pp. 11–22.

Wikipedia (2015, May 19th). Slinky. Retrieved from http://en.wikipedia.org/wiki/Slinky

Zhang, C., & Mao, B. (2016). 3D Building Models Segmentation Based on K-means++ Cluster Analysis. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 42.