User-Guided Volumetric Approximation Using Swept Sphere Volumes for Physically-based Animation

MyungSoo Bae, Young J. Kim
Ewha Womans University, Korea
{msbae,kimy}@ewha.ac.kr

Jinwook Kim
KIST, Korea
jwkim@imrc.kist.re.kr

Abstract

We present an efficient, user-guided volumetric approximation algorithm, specifically designed for physically-based animation. Our method combines automatic and interactive segmentation methods to give users an intuitive and easy way to approximate 3D meshes. Our approach first constructs the simplified medial axis transform (MAT) of the input mesh object, and segments the medial axis (MA) into parts in terms of swept sphere volumes (SSVs) using a region growing method. Then, we decompose the object surface into regions based on the mapping between the segmented MA and the object surface. Each segmented region is approximated with a SSV. These decomposed surface regions can be interactively refined further by splitting and/or merging using a sketch-based input. Experimental results show that our approach produces good volumetric approximation results for different types of object shapes. Moreover, rigid-body dynamics simulation based on our volumetric approximation provides a visually pleasing result.

Keywords: volumetric approximation, swept sphere volumes, mesh segmentation

1 Introduction

Physics simulations are widely used in computer animation, games, haptics, and robotics. In general, the performance bottleneck of physics simulation lies in proximity queries such as distance or collision queries between objects. The performance of these queries depends on the number of objects involved as well as the objects' complexities including the number of boundary polygons.

One way to accelerate proximity queries in physically-based animation is the use of simplified objects for the queries instead of the complicated original geometry. In particular, for real-time applications such as computer games and interactive entertainment, the given mesh models (often called rendering mesh) are typically approximated with a small number of geometric primitives such as spheres, boxes, cylinders, etc., and proximity queries are performed on these simple primitives. Therefore, the quality of approximation will heavily impact on the simulation performance and results. However, such approximation is often done manually in practice, especially in gaming industry, and can be very time-consuming and tedious.

A volumetric approximation for 3D models can be performed automatically or interactively. For automatic approximation methods, one may decompose a 3D object into meaningful parts (part-type segmentation), and approximate each part with a primitive volume. Many automatic segmentation algorithms have been developed in the past [2, 5, 31] and some of them work successfully. However, every mesh segmentation algorithm has its own advantages and disadvantages, so no single segmentation approach is any better than others for various types of objects [2][5]. Also, users may want different semantic-part segmentation depending on their applications or purposes. Thus, determining the semantic parts of a shape can be subjective, and an automatic segmentation for arbitrary shapes is very challenging. For this reason, recently many interactive segmentation methods have been introduced including sketch-based approaches which generate smart cuts based on users strokes [16][26][33]. However, these interactive approaches may be sensitive to user's input or cumbersome to deal with complicated shapes. Therefore, we propose a hybrid approach to combine the automatic and manual mesh decompos-

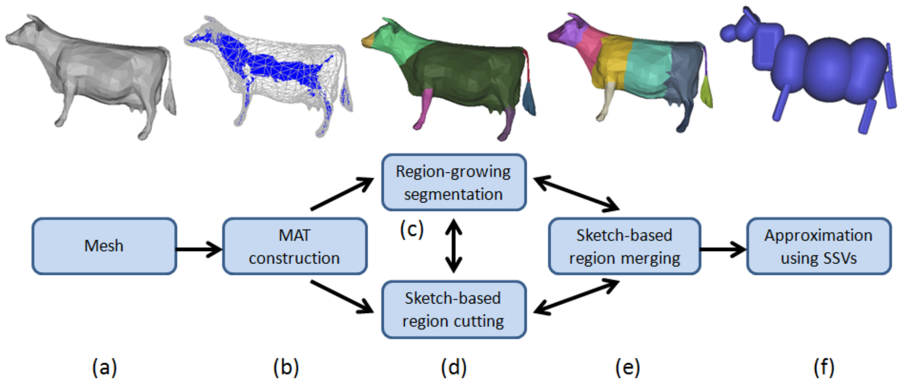

Figure 1: Overview of our user-guided volumetric approximation process.

ing methods together to complement each other to overcome the drawbacks of each approach.

Main Results In this paper, we propose a new sketch-based, semi-automatic approach that approximates 3D shapes with swept sphere volumes for physically-based animation. Our approach provides an intuitive and easy way to tightly approximate 3D meshes, which enables fast proximity queries for physics simulation. The essence of our approach is based on an observation that the MA of SSVs corresponds to linear objects such as a point, a line, and a plane, and such that the segmentation based on the MA of a mesh object is an effective way. In detail, as shown in Figure 1, our approach first constructs a medial axis transform (MAT) that represents the given shape (Figure 1 (b)). Then the medial axis (MA or medial surface) is segmented using a region growing method (Figure 1 (c)), and the segmented medial axis is mapped to the surface boundary. These initially segmented surface regions can be refined by merging or dividing surface regions using a sketch-based method (Figure 1 (d) and (e)). Alternatively, depending on shapes, users can initiate segmentation with the sketch-based region-cutting instead of using the automatic region growing method (Figure 1 (d)). All the segmented regions are finally approximated with swept sphere volumes (Figure 1 (f)).

The rest of this paper is organized as follows. Section 2 presents related work, and section 3 explains how to use the swept sphere volumes (SSV) to approximate a mesh geometry. Section 4 describes our segmentation method using the medial axis transform, and Section 5 presents an interactive segmentation method. In Section 6, the experimental results of our approach are presented, and Section 7 concludes our paper.

2 Related Work

We briefly survey the works relevant to volumetric approximation of a $3D$ object.

Volume approximation Leonardis *et al.*[19] proposed segmentation and approximation methods for range images of a 3D object using superquadric models based on region growing from multi-seeds. Later, Chevalier *et al.*[10] extended this region- growing method to general 3D data points with superellipsoids. They also presented an approximation method based on a split-and-merge approach. This method approximates an object with superellipsoids using a fit-and-split method, then all the pairs of neighboring superellipsoids within some threshold value are merged. Zha *et al.*[35] proposed a fitting-and-splitting algorithm using superquadrics. This approach uses the fitting error and concavity of the data surface to extract dividing planes. Huebner *et al.*[15] also presented an approximation approach based on fitting-and-splitting for robot grasping. This approach decomposes a 3D shape by enclosing the data points using a minimal bounding box and iterative splitting. Mademlis *et al.*[24] presented an approximation approach based on medial surface segmentation, and used it for 3D object search and retrieval. This method decomposes a 3D object into meaningful parts by segmenting the medial surface, approximates those parts with suprellipsoids, and used for 3D object matching applications. Some algorithms used a set of bounding volumes such as spheres [3] or ellipsoids [23] for efficient collision detection. To the best of our knowledge, no prior research work have addressed the issue of approximating an object with SSVs.

Mesh segmentation One of the common methods for volumetric approximation is to decompose complicated 3D objects into parts and to fit primitives to them [24]. Thus, shape segmentation is crucial for a good approximation with an optional number of primitives. Since mesh segmentation is wildly used in many graphics areas, a high number of segmentation algorithms have been proposed in the literature. These algorithms are well surveyed and compared in [2][31][5]. The part-type segmentation which decomposes a shape into meaningful parts is particularly related to volumetric approximation. Many part-type segmentation algorithms are based on the curve skeleton [6][9][29][30]. The curve skeleton of a shape is extracted and segmented at the joint points and/or high curved points. Katz *et al.*[17] proposed a part-type segmentation method by extracting the core component and the feature points. Lee *et al.*[20] presented a segmentation approach based on the minima rule and the part salience theory. Lien and Amato proposed [21] a technique which decomposes a 3D model into approximate convex parts. However, no single algorithm can offer satisfactory results for different types of shapes.

Sketch-based segmentation For manual mesh segmentation, recently a few sketch-based cutting algorithms have been proposed. Some approaches split regions using strokes on the foreground and background regions [16][33]. Meng *et al.*[26] proposed a sketch-based 3D shape cutting tool. This tool intelligently finds the desired cut result and splits an object based on a user-stroke. No prior method has been proposed to combine automatic and sketch-based methods for object segmentation and approximation.

3 Swept Sphere Volume-based Approximation

In this section, we briefly review the swept sphere volume (SSV) [18] which is used as a volumetric primitive in our work, and then explain how we use SSVs to approximate a 3D mesh. The swept sphere volume is a volume swept by a sphere over core primitive shapes such as a point, a line and a rectangle; depending on these core primitives, the corresponding SSVs are the point swept sphere (PSS), the line swept sphere (LSS) and the rectangle swept sphere (RSS). Moreover, depending on the underlying geometry, one of the three SSVs is selected to provide the best fit. Moreover, the SSVs can be also described using medial axis transforms (MAT) in terms of the core primitives.

Since the SSVs have several advantages such as tight fit, efficient distance computation, and fast and simple intersection tests, we adopt SSVs as an underlying volume primitive for approximation. Note that most of existing works use SSVs for only bounding volume primitive (not approximation primitive) without taking into account the approximation error. Thus, the size of SSVs used as a bounding volume is bigger than when it is used as an approximation (Figure 2 (c) and (d)).

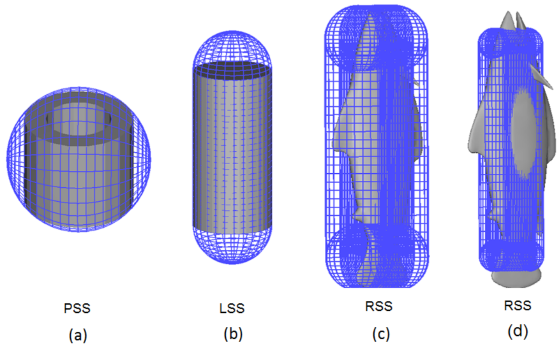

Figure 2: Examples of swept sphere volumes, shown in blue color, as bounding volumes (a)$\sim$(c) and an approximating volume (d).

3.1 SSV Approximation

The construction of SSVs to approximate a given mesh (or its substructure) consists of two steps: proper SSV selection and SSV approximation. For choosing a proper SSV, we first perform the principal component analysis (PCA) on the triangle mesh data. Then, depending on the number of similar lengths in the eigenvectors as a result of PCA, we choose a PSS (one), LSS (two), and RSS (three) just like in [18].

Once a proper type of SSV is selected, we need to set up its geometry to tightly fit the underlying geometry. First of all, the major axes as well as their location can be calculated based on the PCA similarly to [18]. However, in order to actually compute the size (*i.e.* radius r) of a swept sphere for an SSV, we minimize the sum of errors d_i between the vertices of the model and their closest points to the core primitive shape (*e.g.* point, line, rectangle) in SSV:

$$\arg\min_r (\sum_i (d_i - r)^2). \qquad (1)$$

Moreover, if we approximate an object O with a set of SSVs S, the average approximation error $\varepsilon(O,S)$ can be computed as:

$$\varepsilon(O,S) = \frac{1}{2}(\varepsilon_1(O,S) + \varepsilon_2(O,S)), \qquad (2)$$

$$\varepsilon_1(O,S) = \frac{1}{N}\sum_{j=1}^{m}\sum_{i=1}^{N_j}\sqrt{(d_{ji}-r)^2}, \quad (3)$$

$$\varepsilon_2(O,S) = \frac{1}{M}\sum_{j=1}^{m}\sum_{k=1}^{M_j}\parallel s_k - v_k \parallel, \quad (4)$$

where $N = \sum N_j$, N_j is the number of the vertices of the substructure O_j of O that is approximated by j^{th} SSV, m is the total number of SSVs used to approximate O, $M = \sum M_j$, M_j is the number of the sample points (s_k) on j^{th} SSV, and v_k is the closest vertex of O_j from s_k.

In the following sections, we show how to segment and decompose O into substructures O_j using MAT. Whenever a new segmentation O_j is introduced by our algorithm, we need to recompute its SSV approximation.

4 Mesh Segmentation

In this section, we present our part-type mesh segmentation method using a region growing based on the medial axis transform (MAT). The MAT is described briefly first, and then our region growing segmentation is explained.

4.1 Medial Axis Transform

The medial axis (MA) of a shape represents its skeleton, which is defined as a set of the centers of the maximal spheres contained in the 3D object. All the points on a medial axis have at least two closest boundary points. The medial axis transform (MAT) of a shape is also defined as a set of the pairs of sphere centers and radii of spheres [7][3]. Since the MAT represents a volumetric shape, the medial axis is useful in many applications such as object segmentation, animation, recognition, etc.

There are three well-known approaches to compute a medial axis: Voronoi-diagram based methods [1][11], thinning methods [25] and distance-field methods [8][13][32][34]. Although each approach has its own advantages and disadvantages [12], we employed a distance-field-based method to obtain the medial transform of a given model because of its simplicity and implementation robustness. The resulting MA based on distance fields is a discrete approximation of the true MA. Since the medial axis is very sensitive to boundary noises, the separation angle $S_\theta(x)$ [13] at a medial axis point x is used to extract the more stable medial axis. Figure 3 shows the MATs computed by a distance-field based method, and MA points with different separation angles.

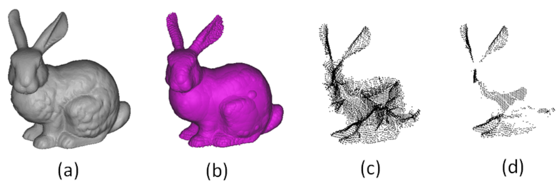

Figure 3: Examples of the medial axis. (a) original mesh, (b) the MATs in terms of spheres, (c) and (d) the medial axis with separation angles $S_\theta = 60°$ and $S_\theta = 100°$, respectively.

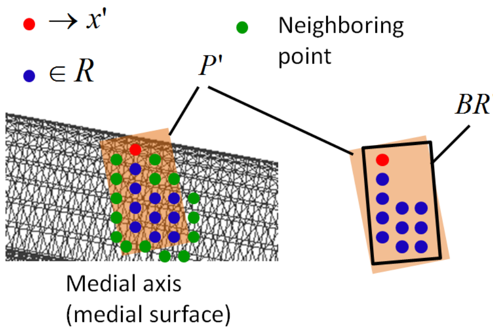

Figure 4: For the current region R, a fitting plane P' and a bounding rectangle BR' of $R' = R \bigcup x'$ are computed to add a new point x' to R.

4.2 Region Growing based on MAT

In our work, SSV is the volume primitive chosen for object approximation. Since the MA of SSV corresponds to the SSV's core primitives such as a point, a line or a rectangle (all linear objects), our segmentation strategy is based on finding the regions on the MA that are *almost* linear and similar to the core primitives. In order to achieve this objective, we use a region-growing method.

Our new region-growing method incrementally expands a segmented region R by adding points on MA as long as a certain condition is satisfied; this condition is represented by four parameters in our work. More specifically, let P' be the fitting plane of $R' = R \bigcup x'$, where x' is a new point on MA that we wish to add to R, so that we can expand the segmented region from R to R' as shown in Figure 4. When the region R' is projected onto P', let BR' be a bounding rectangle of the projected region, that is computed using the first and second largest eigenvalues and associated eigenvectors of the covariance matrix of the region point distribution. Let R'_{sph} be a set of medial spheres in R'. Since the MA of SSV is a linear object, we search for a region

R whose MA is linear (or planar) so that it can be best approximated with an SSV. To detect a planar region on MA, we employ a variant of the fast plane detection algorithm [28]. The four parameters that constrained the incremental region growing include:

Region growing constraints:

1. The mean square error, MSE(R'), between R' and P', that controls the planarity of R'.

2. The distance, $d(x, P')$, between x' and P', that controls the planarity of R'.

3. The portion, $A_r(BR')$, of the uncovered medial surface region area in BR' to the total area of BR' that determines how close the projected region is to a rectangular shape.

4. The mean error, ME(R'_{sph}), between the MA and the surface boundary that determines the level of approximation based on the approximated medial ball on R'.

Then, we explain each step of our region-growing segmentation based on above constraints as follows:

Region growing algorithm:

1. Construct a list of MA points sorted in increasing order of the medial ball size.

2. A point from MA with the largest medial ball size but does not belong to any segmented regions that already exist is selected as a seed.

3. A segmented region R is created with the seed point, and their neighboring MA points are inserted into a neighbor list L_N with ME(R'_{sph}) $< \beta$.

4. The region R grows by retrieving a new MA point x' from L_N until $L_N \neq 0$ if MSE(R') $< \epsilon$ and $d(x', P') < \gamma$ and $A_r(BR') < \alpha$ and ME(R'_{sph}) $< \beta$. Also insert the neighboring points of x' to L_N.

5. Repeat steps (2) $\sim$ (4) for all unassigned MA points.

Note that the fitting plane P' is updated in step 4 by computing the three eigenvectors and their corresponding eigenvalues based on the covariance matrix for the region R'. To do this, we incrementally compute the covariance matrix and the mean squared error as described in [28].

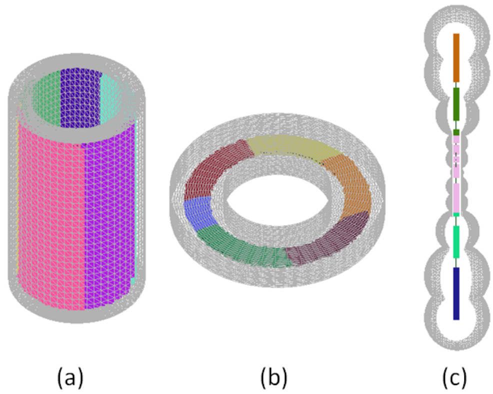

Figure 5: Mesh segmentation based on region growing on MAT. The mesh is displayed in gray color (in wire-frame) and the segmented medial surface is displayed in different colors. (a), (b) and (c) show a long tube, a short tube and a union of spheres, respectively.

Region merging After the region-growing segmentation, some regions can be further merged together to generate a more optimal segmentation. More specifically, for each pair of neighboring regions, if the three parameters of the region-growing constraints except for the distance constraint $d(x', P')$ are less than some thresholds, the two neighboring regions are merged. This process is applied to all pairs of neighboring regions. After segmentation, we map the surface boundary of the mesh to the segmented medial surface by associating each mesh triangle with the nearest MA points.

5 Sketch-based Segmentation

In this section, we present a sketch-based segmentation method that is able to cut and merge regions by allowing users to sketch a curve on the 2D viewport with a simple combination of mouse and keyboard strokes. This way, our system provides the users more control and adaptation of object segmentation.

5.1 Region Cutting

Even though there are a few approaches for interactive 3D mesh segmentation based on intelligent scissoring algorithms [14][16][26][33], they still may require laborious manual work for quality mesh segmentation. On the other hand, our interactive approach is quite simple compared to the existing work, but still can generate good segmen-

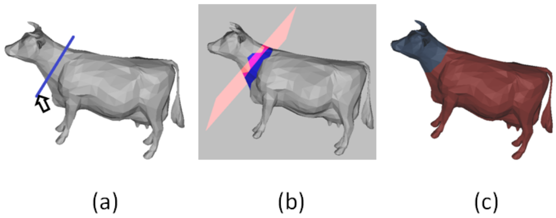

Figure 6: Our sketch-based region cutting. (a) Drawing a 2D line (blue) on the area of the mesh to be cut, (b) triangles (blue) intersected by the plane, that is an unprojected 2D line toward the view direction, and (c) splitting one region into two sub-regions.

tation results, combined with our automatic segmentation algorithm based on MAT. In our work, we split a region into two regions by drawing a 2D cutting line on the 2D viewport, and it is intuitive, simple and easy to learn. This approach is suitable for rough cutting.

More specifically, our approach asks the user to draw a line on the area that they wish to split (the blue line in Figure 6 (a)), and the 2D line in viewport is unprojected toward viewing direction and a cutting plane is constructed ((the pink plane in Figure 6 (b)). Then, the triangles intersected by the cutting plane forms a boundary curve (the blue curve in Figure 6 (b)), and the boundary curve split one region into two sub-regions, each of which can be closed or open (Figure 6 (c)). The cutting boundary curve may be jagged, and it can be repaired by using boundary smoothing methods [16][33]. If more than one boundary-curve is generated, then only the nearest curve to the camera is selected. Finally, when a region is split, its associated medial axis is also split to be used for automatic segmentation, explained in Section 4.

5.2 Region Merging

Our segmentation algorithm is a hybrid of the MA-based automatic region growing method and sketch-based interactive method. However, some regions may have been over-segmented, and the users may want to merge some of the overly segmented regions for a better approximation. Thus, our system allows region-merging by drawing a 2D curve on the regions of interest. Then, all the regions intersected by the 3D curve that are unpro jected toward the camera are selected and merged together as shown in Figure 7.

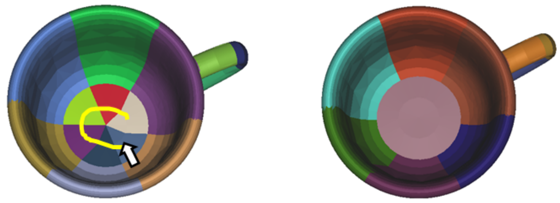

Figure 7: Our sketch-based region merging. (left) Drawing a curve on the regions to be merged, and (right) the result after merging the regions.

5.3 Hybrid Segmentation

Our segmentation algorithm relies on both manual and automatic methods, and each of these methods can precede another in the pipeline of our algorithm. Thus, some data shared by the two methods should be carefully managed in order not to interfere the functionality of other method. For instance, by the definition of MA, a point on MA should have at least two closest points to the object surface. Thus, we have many-to-one mapping between surface triangles and a MA point that our algorithm finds. Therefore, when a cutting operation is applied to object surfaces and some triangles sharing the same MA point are split, the MA point should be also duplicated.

6 Results

We have implemented our hybrid segmentation algorithm using C++ programming language (Visual Studio 6.0) under Windows7 operating system on a PC with an Intel 2.66GHz CPU and 2.7GB of memory. We used the distance filed implementation by [4] to extract the medial axis of an object. We also modified the public proximity package PQP [36] and the real-time dynamics simulation library, VirtualPhysics [37], to validate the applicability of volume approximation algorithm to physically-based animation.

We have tested our volumetric approximation algorithm with a set of closed 3D triangular meshes including rigid, articulated, and CAD models. Some of the results are shown in Figure 11. Figure 8 shows our system for semi-automatic volumetric approximation. During the segmentation process on the right window, users can check the SSV approximation result on the left window.

Figure 9 illustrates our interactive approximation process. Figure 9 (a) and (b) show the MAT extraction and the decomposition based on medial axis segmentation, respectively. Then, either the

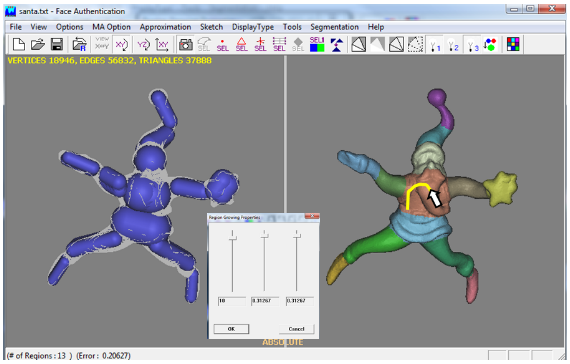

Figure 8: Our volumetric approximation system. The left window displays a SSV approximation result and the right window displays the user-guided segmentation process.

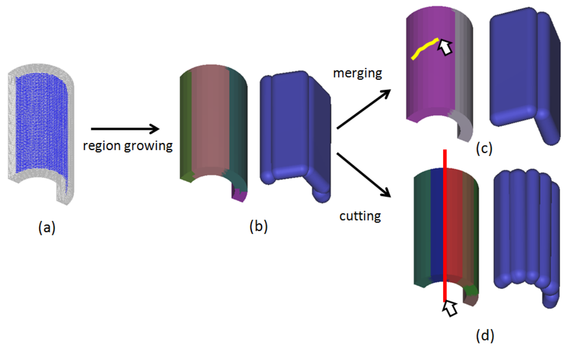

Figure 9: Our interactive approximation process with segmented objects (different colors) and SSV approximation (blue color). (a) Simplified MAT extraction (blue sheet), (b) medial axis segmentation based on region growing, (c) sketch-based region merging, (d) sketch-based region cutting.

sketch-based merging (Figure 9 (c)) or the cutting (Figure 9 (d)) can be followed to produce different approximation results. Figure 10 shows different levels of approximation in terms of tightness using our hybrid system.

We approximated rigid-body dynamics simulation using the results of our volumetric approximation algorithm as shown in figure 12. In Figure 12(a), we drop 15 rigid bodies whose total triangle counts are 524,490, but approximated with 3∼18 SSVs per model. In Figure 12(b), 50 rigid tori models approximated with 8 SSVs per torus are stacked on top of one another. We use the impulse-based dynamics [27] to perform the contact dynamics between SSVs. Our approximated volume provides a visually plausible result for different types of rigid-body dynamics including colliding and resting contact. In general, the quality of simulation depends on the number of approximate primitives; the more SSVs are used, the more realistic the simulation becomes. However, since our hybrid approximate algorithm provides a tight approximation, with a small number of SSVs, the simulation results are still convincing, that can be readily usable for games and movies.

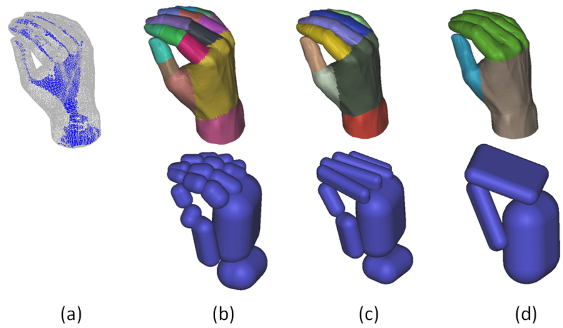

Figure 10: Example of different levels of approximation in terms of tightness. (a) 3D mesh with the medial axis (blue sheet), and (b)∼(d) are segmented objects (top) and approximation with SSVs (bottom) with errors (b) ε=0.023 (c) ε=0.028 (d) ε=0.046.

7 Conclusions

In this paper, we have proposed a semi-automatic volumetric approximation approach for physically-based animation. Our approximation method segments a 3D shape into parts and approximates them with swept sphere volumes. To segment a 3D shape in our method, an automatic method and an interactive method are interwoven to complement each other's weakness. Our automatic method is based on medial surface segmentation and region growing, and our interactive method relies on sketch-based region cutting/merging. Our experimental results have shown that our method generates good volumetric approximations for different types of shapes and the physically-based animation based on our approximation results show a visually plausible result compared to the full simulation based on the original mesh geometry.

Limitations and future work In theory, our approach is applicable to only closed 3D meshes since the MAT is defined for such objects. Our sketch-based approach may have a difficulty to

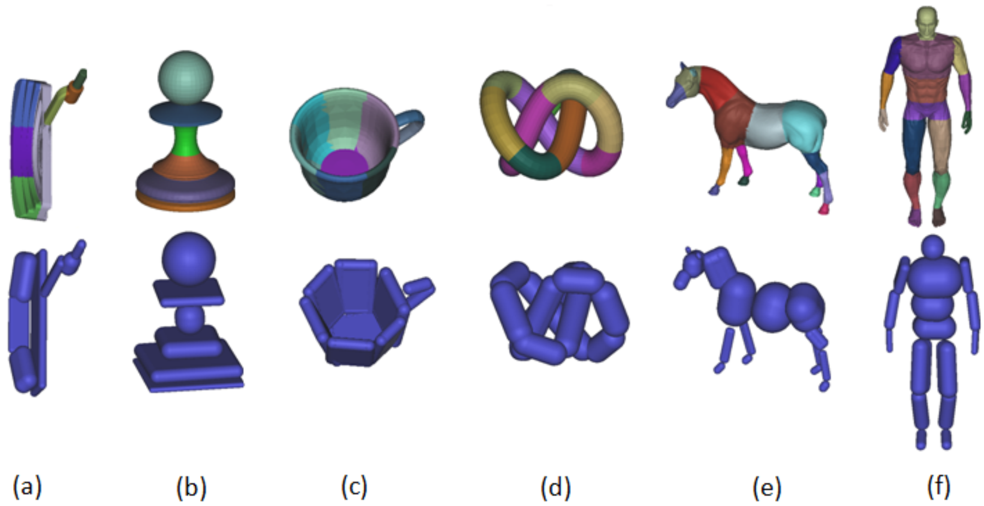

Figure 11: Approximations results using our method for various models. Each model displays the segmentation of the model (top) and its volumetric approximation with SSVs (bottom). The number of mesh triangles (T) and the number of SSVs (S) for each model are shown as T / S: (a) 72576 / 7, (b) 12192 / 6, (c) 8452 / 9, (d) 6210 / 10, (e) 39698 / 18, (f) 18110 / 16).

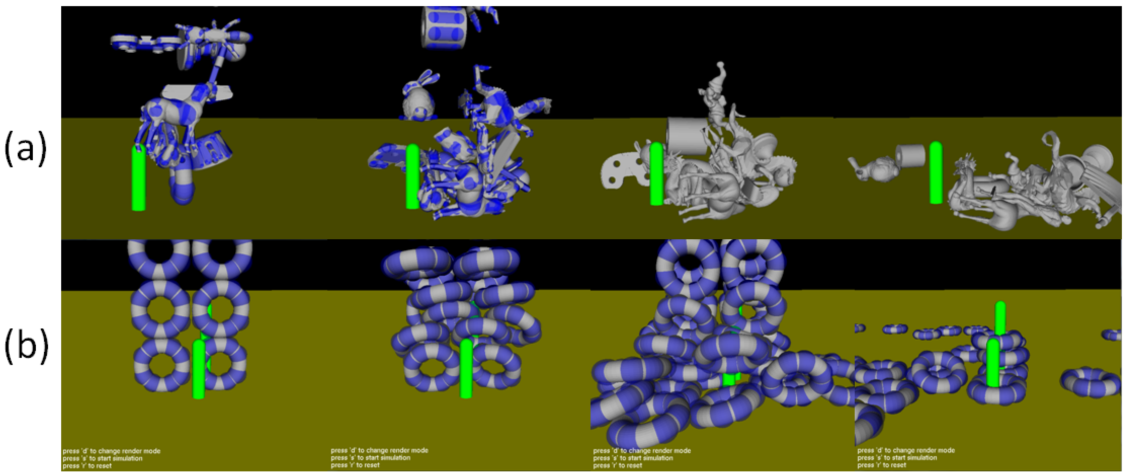

Figure 12: Approximated rigid-body dynamics with SSVs. The SSVs that approximate the underlying geometries are displayed in blue color.

deal with highly non-convex objects, since it would be hard to sketch on visually-occluded regions.

For future work, we would like to extend our algorithm to dealing with non-manifold objects with holes. We also would like to apply a smart cutting algorithm such as a feature-based method to handle cutting on highly curved boundaries in an easy way.

Acknowledgements

This research was supported in part by IT R&D program of MKE/MCST/KOCCA (KI001818) and NRF in Korea (NRF-2011-013-D00112).

References

[1] N. Amenta, S. Choi and R. Kolluri. The power crust, In *Proc. ACM Solid Modeling*, 249-260, 2001.

[2] M. Attene, S. Katz, M. Mortara, G. Patane,

M. Spagnuolo, and A. Tal. Mesh segmentation - A Comparative study. in: *SMI06 Proceedings of the IEEE International Conference on Shape Modeling and Applications*, 2006.

[3] G. Bradshaw and C. O'Sullivan. Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics*, 2004.

[4] B. Chang, D. Cha, and I. Ihm. Computing local signed distance fields for large polygonal models. *Eurographics/ IEEE-VGTC Symposium on Visualization*, 27(3), 2008.

[5] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM SIGGRAPH*, 28(3), 2009.

[6] Z.Q. Cheng, K. Xu, B. Li, Y.Z. Wang, G. Dang, and S.Y. Jin. A mesh meaningful segmentation algorithm using skeleton and minima-rule. *Advances in Visual Computing*, 671-680, 2007.

[7] H.I. Choi, S.W. Choi, and H.P. Moon. Mathematical theory of medial axis transform. *Pacific J. Math.*, 181(1):56-88, 1997.

[8] D. Coeurjolly and A. Montanvert. Optimal separable algorithms to compute the reverse Euclidean distance transformation and discrete medial axis in arbitrary dimension. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3), 2007.

[9] N.D. Cornea, D. Silver, X. Yuan, and R. Balasubramanian. Computing hierarchical curve-skeletons of 3D objects. *The Visual Computer*, 2005.

[10] L. Chevalier, F. Jaillet, and A. Baskurt. Segmentation and superquadric modeling of 3D objects. *Journal of WSCG*, 2003.

[11] T.K. Dey and W. Zhao. Approximate medial axis as a voronoi subcomplex. In:*Proceedings of the seventh ACM symposium on Solid modeling and applications*, 356-366, 2002.

[12] H. Du and H. Qin. Medial axis extraction and shape manipulation of solid objects using parabolic PDEs. In: *Proceedings of the ninth ACM symposium on Solid modeling and applications*, 25-35, 2004.

[13] M. Foskey, M. Lin, and D. Manocha. Efficient computation of a simplified medial axis. in: *Proc. of ACM Symposium on Solid Modeling and Applications*, 2003.

[14] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin1. Modeling by example. *ACM Transactions on Graphics*, 23(3):652-663, 2004.

[15] K. Huebner, S. Ruthotto, and D. Kragic. Minimum volume bounding box decomposition for shape approximation in robot grasping. *IEEE International Conference on Robotics and Automation*, 1628-1633, 2008.

[16] Z. Ji, L. Liu, Z. Chen, and G. Wang. Easy mesh cutting. in: *Eurographics*, 25(3):283-291, 2006.

[17] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *Visual Computer*, 2005.

[18] E. Larsen , S. Gottschalk , M. Lin, and D. Manocha. Fast proximity queries with swept sphere volumes. in: *IEEE International Conference on Robotics and Automation*, 3719-3726, 2000.

[19] A. Leonardis, A. Jaklic, and F. Solina. Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1289-1295, 1997.

[20] Y. Lee, S. Lee, A. Shamir, D. Cohen-Or, and H.-P. Seidel Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design*, 22(5):444 465, 2005.

[21] J.M. Lien and N.M. Amato. Approximate convex decomposition of polyhedra. In *Proc. of ACM Symposium on Solid and Physical Modeling*, 121 131, 2007.

[22] M.C. Lin and S. Gottschalk. Collision Detection Between Geometric Models: A Survey. In *Proc. of IMA Conference on Mathematics of Surfaces*, 1998

[23] S. Liu, C.C.L. Wang and K-C. Hui, J. Xiaogang, and H. Zhao. Ellipsoid-tree construction for solid objects. In: *Proceedings: Symposium on Solid and Physical Modeling*, 2007.

[24] A. Mademlis, P. Daras, A. Axenopoulos, D. Tzovaras, and M.G. Strintzis. Combining topological and geometrical features for global and partial 3-D shape retrieval. in: *IEEE Transactions on Multmedia*, 10(5):819 831, 2008.

[25] A. Manzanera, T. Bernard, F. Preteux, and B. Longuet. Medial faces from a concise

3d thinning algorithm. In *ICCV99*, 337-343, 1999.

[26] M. Meng, L. Fan and L. Liu. iCutter: A Direct cut out tool for 3D shapes. *Computer Animation and Virtual Worlds*, 22(4):335-342, 2011.

[27] B. Mirtich and J. Canny. Impulse-based dynamic simulation. In *in Proceedings of Workshop on Algorithmic Foundations of Robotics*, 1994.

[28] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak. Fast plane detection and polygonalization in noisy 3D range image. in: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[29] R. Raab, C. Gotsman, and A. Sheffer. Virtual woodwork: Making toys from geometric models, 2008.

[30] D. Reniers and D. Telea. Hierarchical part-type segmentation using voxel-based curve skeletons. *The Visual Computer*, 2008.

[31] A. Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539-1556, 2008.

[32] S. Stolpner , P. Kry, and K. Siddiqi. Medial spheres for shape approximation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.99, 2011.

[33] H. Wu, C. Pan, J. Pan, and Q. Yang. A sketch-based interactive framework for real-time mesh segmentation. in: *Computer Graphics International (CGI)*, 2007.

[34] H. Xia and P.G. Tucker. Distance solutions for medial axis transform. in *Proceedings of the 18th International Meshing Roundtable*, 247-265, 2009.

[35] H. Zha, T. Hoshide, and T. Hasegawa. A recursive fitting-and-splitting algorithm for 3-D object modeling using superquadrics. in *14th Int. Conf. on Pattern Recognition*, 1:658 662, 1998.

[36] http://www.cs.unc.edu/.geom/SSV

[37] http://virtualphysics.imrc.kist.re.kr/index.html

Myungsoo Bae is currently a research professor at Computer Graphics & Virtual Reality Center, Ewha Womans University, Korea. He received his PhD in Computer Science from Arizona State University in 2008. His research interests include geometric modeling, computer graphics, computer games and animaiton.

Young J. Kim is currently an associate professor of computer science and engineering at Ewha Womans University in Seoul, Korea. He received his BS and MS degrees in Computer Science and Statistics in 1993 and 1996, respectively, from Seoul National University, and received his PhD in computer Science in August 2000 from Purdue University. His research interests include interactive computer graphics, computer games, robotics, haptics and geometric modeling.