Feature Refinement Strategy for Extended Marching Cubes: Handling on Dynamic Nature of Real-time Sculpting Application

Chien-Chang Ho[†] Yan-Hong Lu[†] Hung-Te Lin[†] Shuen-Huei Guan[†]
Sheng-Yao Cho[†] Rung-Huei Liang[†] Bing-Yu Chen[‡] Ming Ouhyoung[*]

[†]*{murphyho,dino,piaip,drake,ven,liang}@cmlab.csie.ntu.edu.tw,*
[‡]*robin@ntu.edu.tw,* [*]*ming@csie.ntu.edu.tw*
Communication and Multimedia Lab.,
[†*]*Dept. of Computer Science and Information Engineering /* [‡]*Dept. of Information Management,*
National Taiwan University

Abstract

Digital sculpting is a new trend for creating 3D models, but its application in the manipulation of volumetric data raises several issues that need to be addressed. With the Extended Marching Cubes algorithm (EMC), sharp features of 3D models are well preserved. Additionally, the dynamic nature of modifying models in real time needs to be dealt with in sculpting applications: since the sampling of sharp features is implicit, direct modification on cell data will cause problems. A feature refinement strategy is proposed to preserve the dynamically modified model correctly and efficiently. Overall, the proposed methods provide an adaptive resolution and feature-preserved sculpting system that handles dynamic behavior in real-time performance.

1. Introduction

Designing convincing 3D models is a challenge for volumetric sculpting systems. Defects in the resulting models are unacceptable for artists. Several isosurface-extracting techniques have been proposed to present volume data with sharp features [8], adaptive resolution [7], and even progressive encoding [9]. However, directly editing the extracted isosurface without special actions still causes problems.

Basic sculpting systems today have the following 3 steps: 1) editing models by altering the value of voxels, 2) extracting the isosurface from volumetric data, and 3) rendering the isosurface to visualize the resulting model. In such application the isosurface-extracting techniques are independent of the model-editing. However, this is inappropriate for highly detailed models because large models require impractically large memory to store voxels. Additionally, repeated iterations at step 2 may waste time to produce a whole isosurface even if only part of it has been modified. These problems could be avoided if editing and isosurface-extracting steps could be combined together. In other words, an editable isosurface-extracting algorithm is needed for sculpting applications to reduce the memory requirements.

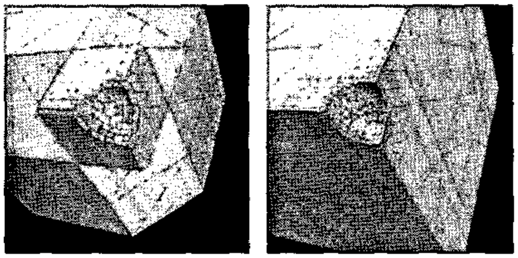
(a) (b)

Figure 1. Carving a volumetric model
(a) adaptive resolution. (b) sharp feature preserved

The major problem of editable isosurface-extracting algorithm comes from editing sharp features. In some cases, direct modification of sharp features is impossible because of implicit sampling. By converting all cases into editable cases, we propose a novel Feature Refinement Strategy based on EMC [11] with adaptive resolution to preserve features correctly in dynamic editing. This will be illustrated further in Section 3. In Figure 1, we demonstrate the resulting cube after a carve operation is applied.

Adaptive resolution and feature refinement have become critical to sculpting applications that handle volumetric data in dynamic way. The breakthrough of our work is that it allows sharp features to be dynamically manipulated in an efficient adaptive way.

2. Related Work

0-7803-8603-5/04/$20.00 ©2004 IEEE

Considering sculpting applications, many previous works [1,2,5,16] used a sampled scalar-field to represent models. [10,17] used Non-Uniform Rational B-splines to represent the isosurface. [3] adopted an adaptive resolution scheme based on the local detail of graver (the cursor to locate and apply operations as carving of filling etc.) to represent the sculpting data. Nevertheless, sculptures created by these systems do not preserve sharpness, still revealing rounded corners and smoothed edges. Perry et al. [14] proposed a sculpting system with *Adaptively Sampled Distance Fields (ADFs)* to preserve sharp features with an efficient refinement but this was limited to local modification. Recently, Lu et al.[12] developed a volumetric sculpting system using EMC in uni-resolution. Other groups used a mesh-based [4,6] approach with adaptive refinement and deformation operations with haptic interaction and texture painting.

Although many of these contributions deal with volumetric sculpting, few previous works have concentrated on feature-sensitive volumetric sculpting.

3. Feature-Sensitive volumetric sculpting on adaptive isosurface

The procedure of preserving features adaptively consists of three steps:

1. **Collision detection**: locates the cells which need to be modified (i.e., carving operations).
2. **Refinement**: subdivides the cells with features to avoid defects.
3. **Modification**: applies carving of filling operations and then generates new features.

The first and third steps can be performed just like in a Marching Cubes based sculpting system so we will only analyze step 2, the algorithm of feature preserved refinement.

3.1 Problems of editing sharp features

Due to the implicit sampling of sharp features with *Quadratic Error Function* (QEF) [7], some defects occur when making changes to feature preserving adaptive isosurfaces. The two major defects are Invalid Feature and Undetected Feature.

Feature points in EMC are sampled by hermite data (intersecting points and normal vectors), so **invalid features** may occur when some of the intersection points and normal vectors are changed while new feature points are incorrectly sampled. In Figure 2, the graver perturbs the feature point and then the newly sampled feature is found outside the original bounding box.

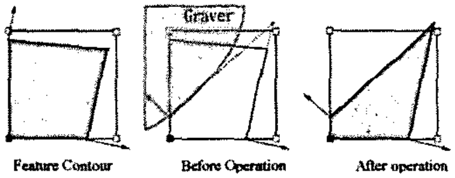

Figure 2. Invalid Features

In Figure 3, the feature is applied with the carving operation but the result is incorrect because the **undetected feature** was not calculated for intersection.

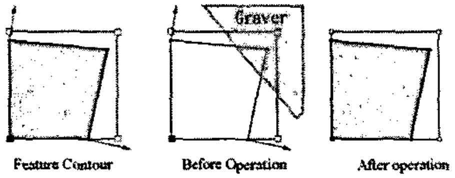

Figure 3. Undetected Feature

3.2 Feature refinement strategy

Our proposal to enable sharp features to be safely preserved under modification is to subdivide the cells. To the cells with sharp features two iterations of subdivision are applied, as shown in Figure 4. For each iteration the original feature is reconstructed in two steps. First, determine the binary value of voxels with ray casting and visible surface determination algorithm. Second, resample *hermite data* [7] by linear interpolation.

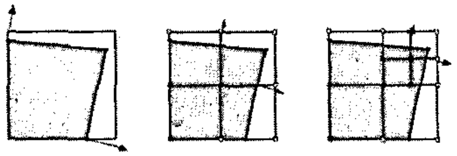

Figure 4. Feature preserving subdivision

The main idea of our strategy is to refine feature parts until the carving operations can be applied without intersecting undetermined data. We analyze and generalize all the cases of applying carving operations to sharp features in two categories. The handlers of each category are designed to solve both undetected and invalid conditions. We denote feature points as *S*, graver volume as *Gv*, hermite data as *h(e, n)* (exact intersection point e and normal n), and hermite data set as *H*, which forms the sharp feature *S*.

In a carving operation, the first category is that the feature point is outside graver volume. Feature point inside graver volume in a carving operation is classified as the second category, denoted as $S \subset G_v$. Two steps are chronologically executed for each category as in Table 1:

Feature S	Hermite set H	Handler
$S \not\subset G_v$ **Feature outside graver**	$\exists h(e,n) \in H \mid e \subset G_v$ **(Hermite data all or partially inside graver)**	(i) Feature refinement with **Iteration (A)** to prevent *invalid feature* (ii) Apply operations **Example in Figure 5**
	$\forall h(e,n) \in H \mid e \not\subset G_v$ **(All outside)**	Apply operations directly
$S \subset G_v$ **Feature inside graver**	$\exists h(e,n) \in H \mid e \not\subset G_v$ **(All or partially inside)**	(i) Feature refinement with **Iteration (B)** to avoid *undetected feature*. (ii) Apply operations **Example in Figure 6**
	$\forall h(e,n) \in H \mid e \subset G_v$ **(All inside)**	No operations required

Table 1. Feature Refinement Strategy

Iteration A: (see Figure 5)

For each cell that contains any feature point, recursively apply feature subdivision until hermite data which samples the feature point is outside graver.

Iteration B: (see Figure 6)

For each cell that contains any feature point, apply feature subdivision until all refined inner edges are inside original feature grid and intersect with graver.

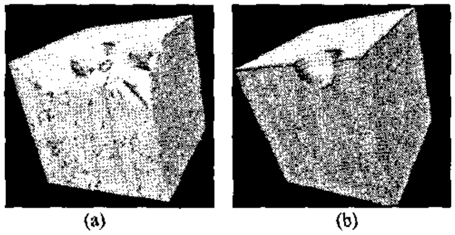

Figure 7. A comparison in low-level resolution (a) Carving features **without** feature refinement (b) Carving with feature refinement strategy applied.

The advantage of our feature refinement strategy is illustrated in Figure 7.

Concerns that, errors would be introduced from subdivision and refinement operations were answered by our data/experiments which showed that subdivision by linear interpolation is sufficient and did not introduce errors other than precision truncation or rounding errors. We applied Ju's method to minimize QEF for feature information while applying operations. The error estimation is the same as described in [7]. In brief, subdivision of cells will repeat until a predefined threshold is met if a pre-defined threshold was given.

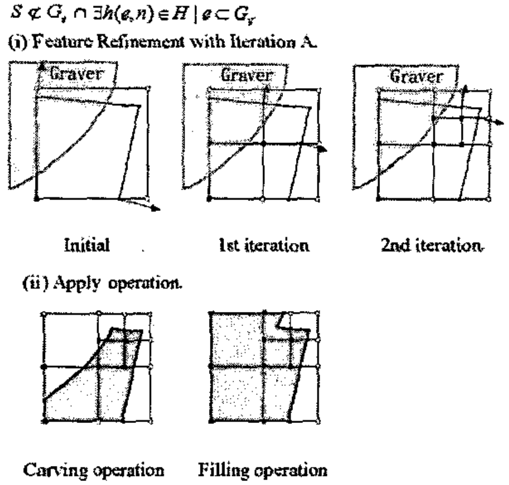

Figure 5. Example of Feature Refinement Strategy when features are outside graver

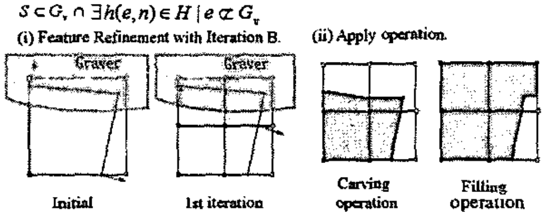
$S \subset G_v \cap \exists h(e,n) \in H \mid e \not\subset G_v$
(i) Feature Refinement with Iteration B.
(ii) Apply operation.
Graver
Graver
Initial
1st iteration
Carving operation
Filling operation

Figure 6. Example of Feature Refinement Strategy when features are inside graver

4. Results

Our system is running on a Pentium-4 1.4GHz PC and ELSA Gloria III Graphics Card. We use a force feedback device, PHANToM Desktop[1], to support the haptic system [13] as in Figure 8. Currently we provide carving, filling, painting operations when sculpting. It is so easy to use that an artist can create a dragon model with 71,233 triangles in Figure 9 in only 25 minutes, and only 11.2M memory is used with 28+ FPS. The internal data structure is derived from EMC

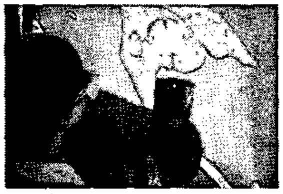

Figure 8. Using our system with PHANToM

[1] SensAble Technologies Inc. freeform™ modeling system, http://www.sensable.com/freeform, 1999.

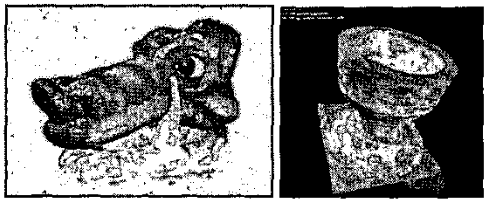

Figure 9. *Results. (a) Dragon, 71,233 triangles, 28+ FPS (b) Cup, 15,627 triangles, 63FPS*

and Adaptive Octree with *crack patching* [15].

The dual contouring [7] provides that a CSG operation takes approximately 30 milliseconds to compute. Our method takes approximately 18 milliseconds to perform a similar operation.

5. Conclusions and Future Work

In this paper we propose an algorithm based on EMC with feature preserving and adaptive resolution for volumetric sculpting systems that achieves real-time performance. By designing our feature refinement algorithm, problems such as invalid feature and undetected feature are prevented in applying EMC algorithm to adaptive resolution volumetric sculpting. Therefore, sharp features on adaptive resolution isosurface are always well preserved. In brief, our system makes volumetric sculpting more practical and usable. We anticipate this concept be evaluated for the next generation of commercial CAD products.

Currently our system supports carving, filling, and painting operations. There are more operations can be implemented like smoothing or deformation in the future. Furthermore, we can add support for more material simulation for both visual and haptic rendering.

6. Acknowledgment

We would like to thank Tien-Jung Huang (National Taiwan University of Art) for using our system to create models. We wish to thank Wan-Chun Ma *(National Taiwan University)* for his remarks and suggestions. This work was partially supported by National Science Council 90-2662-E-002.

7. Reference

[1] J. Baerentzen, "Octree-based volume sculpting", *Proc. of IEEE Visualization*, pages 9–12, 1998.

[2] E. Ferley, M.-P. Cani, and J.-D. Gascuel, "Practical volumetric sculpting", *The Visual Computer*, 16(7): pages 469–480, November 2000.

[3] E. Ferley, M.-P. Cani, and J.-D. Gascuel, "Resolution adaptive volume sculpting", *Graphical Models (GMOD)*, 63(6): pages 459–478, March 2002.

[4] M. Foskey, M. A. Otaduy, and M. C. Lin, "Touch-enabled 3d model design", *Proc. of IEEE Virtual Reality*, pages 119-126, March 2002.

[5] T. A. Galyean and J. F. Hughes, "Sculpting: An interactive volumetric modeling technique", *Proc. Of ACM SIGGRAPH*, 25(4): pages 267–274, 1991.

[6] A. D. Gregory, S. A. Ehmann, and M. C. Lin, "intouch: Interactive multiresolution modeling and 3d painting with a haptic interface", *Proc. of IEEE Virtual Reality*, pages 45–52, March 2000.

[7] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data", *Proc. of ACM SIGGRAPH*, 21(3): pages 339–346, 2002.

[8] L. P. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel, "Feature sensitive surface extraction from volume data", *Proc. of ACM SIGGRAPH*, pages 57–66, 2001.

[9] Haeyoung Lee, Mathieu Desbrun, Peter Schröder , "Progressive encoding of complex isosurface", *Proc. of ACM SIGGRAPH*, pages 471-476, 2003

[10] F. W. B. Li, R. W. H. Lau, and F. F. C. Ng, "Collaborative distributed virtual sculpting", *Proc. Of IEEE Virtual Reality*, pages 217–224, November 2001.

[11] W. Lorensen and H. Cline, "Marching cubes: A highresolution 3d surface construction algorithm", *Proc. Of ACM SIGGRAPH*, pages 163–169, 1987

[12] Y. Lu, W. Wang, R. Liang, and M. Ouhyoung, "Virtual sculptor: A feature preserving haptic modeling system", *Proc. of ACM International Workshop on Immersive Telepresence (ITP2002)*, 2002.

[13] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling", *Proc. of ACM SIGGRAPH*, pages 401–408, 1999.

[14] R. N. Perry and S. F. Frisken, "Kizamu: a system for sculpting digital characters", *Proc. of ACM SIGGRAPH*, pages 9–12, 2001.

[15] R. Shekhar, E. Fayyad, R. Yagel, and J. F. Cornhill, "Octree-based decimation of marching cubes surfaces", *Proc. of IEEE Visualization*, pages 335–342, 1996.

[16] S. W. Wang and A. E. Kaufman, "Volume sculpting", *Proc. of ACM Symposium on Interactive 3D Graphics*, pages 151–156, 1995

[17] J. P. Y. Wong, R. W. Lau, and L. Ma, "Virtual 3d sculpting", *The Journal of Visualization and Computer Animiation*, 11(3): pages 155–166, 2000.