Automatic Generation of Coarse Bounding Cages from Dense Meshes

Chuhua Xian[1], Hongwei Lin[1,*], Shuming Gao[1]
[1]State Key Lab. of CAD & CG, Zhejiang University, Hangzhou 310058, P.R. China

Abstract—**The coarse bounding cage of a dense mesh plays important roles in computer graphics, computer vision, and geometric design. Specifically, in volume-based deformation, a coarse bounding cage is required to manipulate the dense mesh model it enclosed; in subdivision surface fitting, the fitting starts from a coarse cage bounding the fitted dense mesh or point set; and so on. However, the generation of a coarse bounding cage is mainly by interactive ways, which are very tedious and time-consuming. In this paper, we develop a fully automatic method to generate a coarse cage bounding a dense mesh model. The automatically generated coarse bounding cage can keep the topological structure and major geometric features of the original mesh model, which is validated by theoretical analysis and experimental data presented in this paper. Further more, we employ the automatically generated coarse bounding cage in some applications, such as deformation, and subdivision fitting, producing good results.**

Keywords—**bounding box, coarse bounding cage, automatic generation, deformation, subdivision surface fitting**

1. Introduction

The *Coarse Bounding Cages*(abbr. *CBC*) of dense mesh models are required in lots of applications in geometric design, computer graphics, and computer vision. For example, in volume-based deformation, a coarse cage bounding a dense mesh model needs to be extracted and is employed to manipulate the mesh model [1], [2], [3], [4], [5]. In geometric design, to fit a dense mesh by a subdivision surface, its coarse bounding cage is taken as the initial control mesh, and repeatedly subdivided to approximate the dense mesh [6], [7]. Moreover, the CBC is also used in collision detection to reduce the computation complexity [8], [9].

Although the CBC of a dense mesh is very useful, it is constructed mainly by interactive ways currently, either entirely manually [6], or by subdividing a bounding box [3], [10]. To make sure the generated coarse cage strictly enclose the dense mesh, users have to adjust the vertices of the cage by hand. Thus, the interactive CBC generation methods are very tedious and time-consuming. More seriously, when the topological structure of the mesh model is complex, as illustrated in Fig. 1, interactive methods are hard to generate a CBC homeomorphic to the original mesh model.

Specifically and evidentally, the CBC of a dense mesh can be generated by mesh simplification and offsetting (or extruding). However, mesh simplification can not guarantee the generated CBC encloses the original model, and offsetting the simplified mesh outwards may produce self-intersection. Thus, with mesh simplification and offsetting, lots of user interactions are required to adjust the CBC vertices to generate a valid CBC.

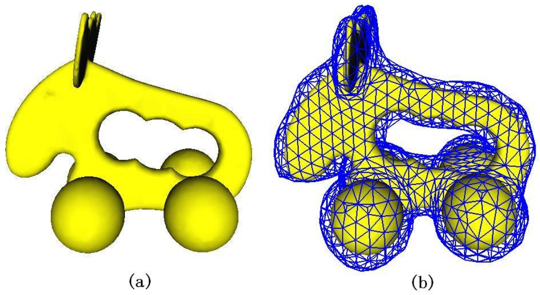

Fig. 1. Mesh model *elk*(left) and its automatically generated coarse bounding cage (right).

In this paper, we develop a fully automatic method to generate the coarse bounding cage from a dense mesh. The automatic CBC generation method and the CBC it generated have the following advantages(refer to Fig. 1):

1) **Homeomorphism:** The automatic method can generate a CBC homeomorphic to the original dense mesh (refer to Fig. 1, and section 3.6 for details).
2) **Shape preservation:** By selecting an appropriate size of the voxels, the constructed CBC can preserve the salient features of the original dense mesh (see section 4).
3) **Multi-resolution:** The automatic method is progressive and can generate multi-resolution CBCs by making the size of voxels smaller and smaller. Different applications can choose different CBCs with different resolutions as their cages.
4) **Full automation:** The method is fully automatic, and greatly reduces the interactive burden of users.
5) **Efficiency:** Since the method is fully automatic, the generation of a CBC is generally accomplished within several seconds.

The remainder of this paper is organized as follows. In section 2, we present some basic concepts and the overview of the algorithm. The automatic CBC genera-

*Corresponding Author
*Email:{xianchuhua,hwlin,smgao}@cad.zju.edu.cn

tion algorithm is described in detail in section 3. Section 4 presents some results and experimental data. Section 5 demonstrates the capability of the automatically generated CBC in some applications. Finally, section 6 concludes the paper.

2. Preliminaries

2.1 Basic Concepts

Given a dense mesh model with connected mesh surface, the automatic CBC generation method developed in this paper first voxelizes the mesh model, and then classifies the voxels into three types, that is, the voxels which intersect with the mesh surface, called *feature voxels*; the voxels out of the model, called *outer voxels*; the voxels in the model, called *inner voxels*. We assign value 0 to the feature voxels, 1 to the outer voxels, and -1 to the inner voxels. They make up of the *tri-values distance field.*

To measure the degree of sparseness of a CBC, we define the *sparse factor*, denoted by s_F:

$$s_F = \frac{N_c}{N_d}, \tag{1}$$

where, N_d and N_c are the numbers of vertices of a dense mesh and its CBC, respectively. The sparse factor is also used to determine the initial resolution of voxelization (See section 3.5).

On the other hand, to check the shape preservation capacity of a CBC, we employ the *shape descriptor* proposed in Ref. [11] to measure the similarity between a dense mesh and its CBC (See section 4).

2.2 Overview of the Algorithm

As illustrated in Fig. 2, the automatic CBC generation algorithm from a dense mesh model mainly includes the following steps:

1) Compute the bounding box of the dense mesh by Principal Component Analysis (PCA);
2) Voxelize the mesh model, identify feature voxels, and calculate the tri-value distance field;
3) Extract and triangulate the outer faces of the feature voxels;
4) Smooth the CBC by an improved mean curvature flow method.

3. Coarse Bounding Cage Generation

Given a dense mesh model M with *connected* mesh surface, denoted by ∂M, a closed and connected mesh surface separates the space in two parts: inside and outside. In this section, an automatic coarse bounding cage (CBC) generation algorithm will be described in detail.

3.1 Bounding Box Computation by PCA

The first step of the CBC generation algorithm is to compute the bounding box of the initial dense mesh model M by principal component analysis (PCA) [12]. Specifically, the (p, q, r)-th moment m_{pqr} of mesh M is given by,

$$m_{pqr} = \int_{\partial M} x^p y^q z^r dxdydz. \tag{2}$$

In the discrete form, denoting the coordinates of vertices of mesh M by $\{x_i, y_i, z_i\}_{i=1}^N$, its (p, q, r)-th moment can be approximated by

$$\hat{m}_{pqr} = \frac{1}{N} \sum_{i=1}^{N} x_i^p y_i^q z_i^r. \tag{3}$$

Thus, the covariance matrix of the second-order moment is constructed with Eq. (3). By performing singular value decomposition to the covariance matrix, we get three characteristic direction vectors as the axes of a new Cartesian coordinate system with the origin at the centroid of mesh M. Hence, a bounding box of M is built in the new Cartesian system. In our implementation, the bounding box is slightly enlarged to ensure the generated CBC encloses the dense mesh model.

The so constructed bounding box captures the major geometric shape of the mesh M, and can reduce the computation complexity in the following voxelization.

3.2 Voxelization and Tri-value Distance Field Calculation

In this step, the bounding box generated in section 3.1 is discretized into a series of voxels, feature voxels are identified, and the tri-value distance field is computed.

After the bounding box is discretized into voxels at an initial resolution, which is determined by a user specified or predefined sparse factor (please refer to section 3.5 for details), the feature voxels should be identified. Feature voxels are the ones intersecting with the mesh surface. Therefore, to identify feature voxels, we need to check which voxels the mesh M intersects with.

To do so, the *max-norm distance computation* method developed in [13] is employed. For clarity, we give a brief description of the method, and details can be found in Ref. [13].

As illustrated in Fig. 3, to calculate the max-norm distance between the center O of a voxel and a triangle $\triangle P_1P_2P_3$ (Fig. 3 b), twelve auxiliary partition triangles should be constructed (Fig. 3 a). Suppose t_1 and t_2 are the intersections between triangle $\triangle P_1P_2P_3$ and one of the partition triangles, the max-norm distance between O and $\triangle P_1P_2P_3$ is the minimum of the distances from O to P_1, P_2, P_3, t_1 and t_2. Let w_v, d_∞ be the length of an edge of a voxel and the max-norm distance from its center to a triangle of mesh M, respectively. If $d_\infty \leq \frac{w_v}{2}$, the voxel intersects with the triangle, and it is identified as a *feature voxel*, assigned value 0.

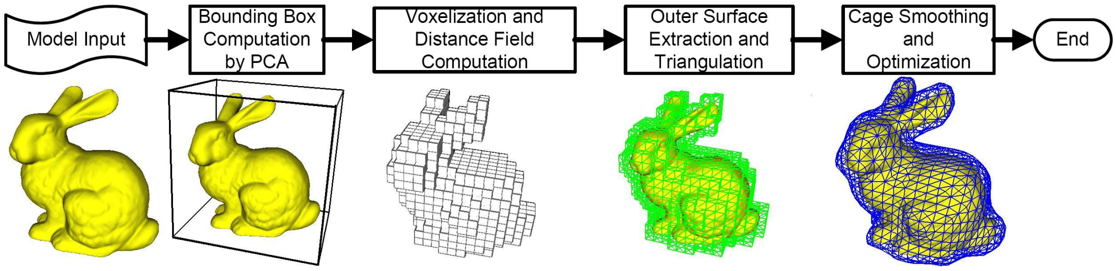
Model Input
Bounding Box Computation by PCA
Voxelization and Distance Field Computation
Outer Surface Extraction and Triangulation
Cage Smoothing and Optimization
End

Fig. 2. Flowchart of the automatic CBC generation algorithm.

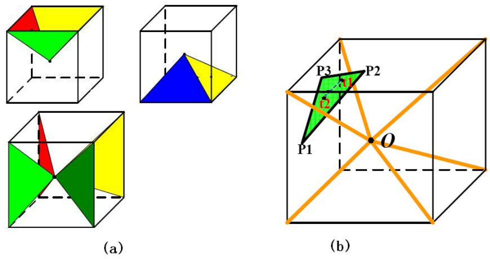

Fig. 3. Computing max-norm distance from the center of a voxel to a triangle.(a)Partitioning triangles; (b) max-norm distance computation for a triangle [13].

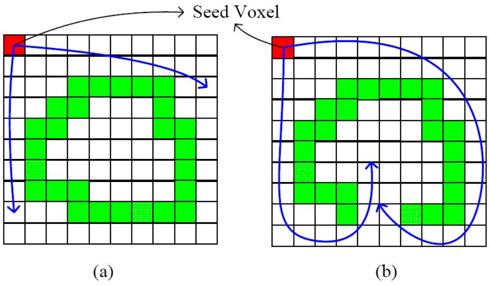

Fig. 4. Sketch of voxel classification by filling algorithm. (a)Closed, (b)open.

As stated above, the feature voxels are assigned value 0. If the mesh M is closed, the feature voxel set separates the discretized bounding box into inner part and outer part. We employ the *filling algorithm* [14] (see Fig. 4) to identify these inner voxels and outer voxels.

The outer voxels are first distinguished by the filling algorithm starting from a seed. According to the construction of the bounding box, the eight corner voxels are certainly outer voxels, and can be taken as the seed. All outer voxels are assigned value 1.

The remaining voxels, other than feature and outer voxels, are inner voxels, and assigned value -1. In this way, the tri-value distance field is obtained.

The tri-value distance field takes effect in extracting outer faces (see section 3.3), and in ensuring the CBC strictly enclose the dense mesh M in cage smoothing (see section 3.4).

It should be pointed out that, if mesh M is open, it is possible that the feature voxel set is also open, and there are only two kinds of voxels, feature and non-feature(see Fig.4b). In this case, we set the value 0 to the feature voxels, and 1 to the non-feature voxels.

Noticeably, the scan-conversion method [14] can also be employed to distinguish the outer and inner voxels. However, it will fail in some singular cases, and is not so efficient as the filling algorithm.

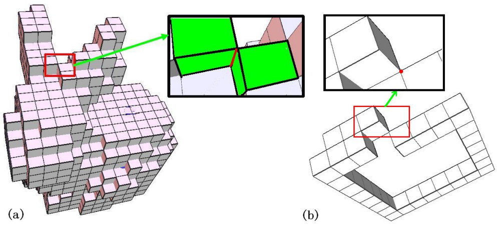

Fig. 5. Non-manifold edge and vertex. (a)Non-manifold edge with more than two adjacent faces, (b)non-manifold vertex where two sheets touch.

3.3 *Outer Surface Extraction and Triangulation*

With the tri-value distance field, the outer faces of the feature voxels can be extracted easily. The outer face is the one adjacent to an outer voxel and a feature voxel, with value 1 and 0, respectively. Thus, by checking the values of adjacent voxels for each face of each feature voxel, the outer surface of the feature voxel set can be generated, which constitutes the initial CBC, a quadrilateral mesh.

However, it is possible that the initial CBC is not 2-manifold. That is, in the initial CBC, there may be non-manifold edges (Fig. 5a), whose adjacent faces are more than two, or non-manifold vertices(Fig. 5b), where two sheets touch.

A non-manifold CBC needs further process to make it be a valid 2-manifold mesh. As illustrated in Fig. 6, for a non-manifold edge (Fig. 6a), we perform an *voxel-attaching* operation by specifying one outer voxel adjacent to the edge as a feature voxel (Fig. 6b); similarly, for a non-manifold vertex, a *vertex-split* operation is employed to separate its adjacent feature voxels.

The voxel-attaching and vertex-split operations make a non-manifold CBC be a valid 2-manifold mesh. Since it is quadrilateral, its triangulation is straightforward, just by splitting one quadrangle to two triangles.

3.4 *Cage Smoothing*

The shape of the CBC so extracted is very scraggly, containing many step-like joints (See Figs. 2, 5). Hence,

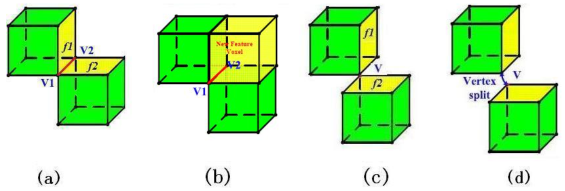

Fig. 6. Mesh correction. (a)Non-manifold edge, (b) voxel-attaching operation, (c)non-manifold vertex, (d)vertex-split operation.

it should be smoothed to generate a CBC with desirable shape. Differing from common mesh smoothing, in the CBC smoothing procedure, we should ensure that the CBC always encloses the dense mesh M. In this section, the mean curvature flow method [15] is improved to smooth the CBC while guaranteeing that it always encloses the mesh M during smoothing.

The mean curvature flow smoothing method performs iteratively by,

$$v_i^{k+1} = v_i^k - \mu dt H\mathbf{n}, k = 0, 1, 2, \cdots \tag{4}$$

where, v_i^0 are the vertices of the initial CBC, H is the mean curvature, and $\mathbf{n}$ is the unit normal vector. μdt is the time step satisfying $0 \le \mu dt \le 1$ [15]. In our implementation, we set $\mu dt = 1$.

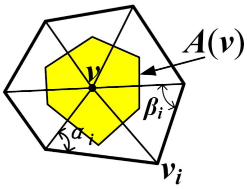

Fig. 7. Sketch for the discrete form of $-H\mathbf{n}$.

The discrete form of $-H\mathbf{n}$ can be computed by [15],

$$-H\mathbf{n} = \frac{1}{4A(v)} \sum_{v_i \in N(v)} (\cot\alpha_i + \cot\beta_i)(v_i - v), \tag{5}$$

where $N(v)$ is the set of 1-ring neighbor vertices around v, $A(v)$ denotes the Voronoi area around v, and α_i, β_i are two angles opposite to edge (v, v_i), respectively(see Fig. 7).

With the discrete form (5), at a convex vertex, the curvature vector $-H\mathbf{n}$ points inwards; at a concave vertex, it points outwards (see Fig. 8).

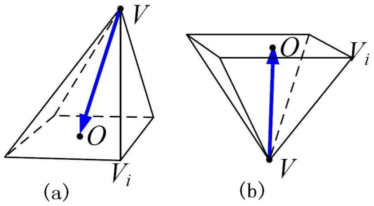

Fig. 8. (a) At a convex vertex, $-H\mathbf{n}$ points inwards; (b) at a concave vertex, $-H\mathbf{n}$ points outwards.

On the other hand, denoting by $d(i, j, k)$ the tri-value distance field defined in the voxels, the gradient vector,

$$\begin{aligned}\nabla d = (&d(i+1, j, k) - d(i, j, k),\\ &d(i, j+1, k) - d(i, j, k), d(i, j, k+1) - d(i, j, k)),\end{aligned}$$

always points outwards.

Therefore, letting θ be the angle between $-H\mathbf{n}$ and ∇d at a CBC vertex, if $0 \le \theta < \frac{\pi}{2}$, both vectors point outwards. In this case, we make the vertex move along the direction ∇d. If $\frac{\pi}{2} \le \theta < \pi$, the curvature vector $-H\mathbf{n}$ points inwards (Fig. 9). To keep the current vertex out of the mesh M, we force the vertex moving along the direction $-H\mathbf{n}\sin\theta$, which lies in the plane perpendicular to the gradient vector ∇d. Hence, the improved iterative smoothing format is,

$$v_i^{k+1} = \begin{cases} v_i^k + \mu dt H \nabla \mathrm{d}, & 0 \le \theta < \frac{\pi}{2}, \\ v_i^k - \mu dt H\mathbf{n}\sin\theta, & \frac{\pi}{2} \le \theta < \pi. \end{cases} \tag{6}$$

Finally, we check the distance value at each cage vertex after each smoothing. If some vertex penetrates into the original model, we move it out of the original mesh along the direction of the gradient vector.

The improved mean curvature flow smoothing method stated above keeps the CBC out of the dense mesh M during smoothing, and produces good results in practice, as illustrated in sections 4 and 5.

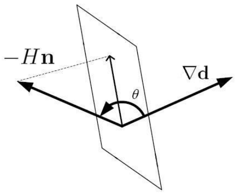

Fig. 9. The angle between the gradient vector and curvature vector is greater than $\pi/2$.

3.5 Initial Resolution

In section 3.2, the bounding box is discretized into voxels with an initial resolution. It can be computed approximately by a user-specified or pre-defined sparse factor s_F.

Suppose n is the initial resolution, that is, the bounding box is discretized into $n \times n \times n$ voxels. Then, on the six outer faces of the discretized bounding box, there are totally $6(n+1)^2 + 12n - 4 \approx 6n^2$ vertices. The six outer faces constitute a bounding cage, so the number of its vertices is a good approximation to that of the actual CBC. Thus, according to the definition of the sparse factor (Eq. (1)), we have,

$$s_F \approx \frac{6n^2}{N_d},$$

where N_d is the number of vertices of the dense mesh M. Therefore, the initial resolution n can be determined as,

$$n \approx \sqrt{\frac{N_d\ s_F}{6}}.$$

3.6 *Homeomorphic CBC Generation*

Some applications, such as subdivision surface fitting, require that the dense mesh M and its CBC are homeomorphic. In this section, we develop an algorithm to generate the CBC homeomorphic to the dense mesh it encloses.

As stated above, the automatically generated CBC keeps out of the dense mesh M. With larger and larger voxelization resolution, the generated CBC will approach closer and closer to the dense mesh, acquiring more and more topological and geometric features of the mesh model M. Hence, by increasing the resolution of voxelization, if we can get a CBC whose value of Euler formula, $G_c = V_c - E_c + F_c$, is equal to that of the dense mesh M, $G_d = V_d - E_d + F_d$, they are homeomorphic to each other. Here, V_c, E_c, and F_c are the numbers of vertices, edges, and faces of the CBC, respectively; similarly, V_d, E_d, and F_d are the numbers of vertices, edges, and faces of the dense mesh M, respectively.

Consequently, the following algorithm is developed to generate the CBC homeomorphic to the original mesh M.

```
Calculate $G_d = V_d - E_d + F_d$ (Euler formula for
the dense mesh $M$);
Voxelize the bounding box with an initial
resolution, and compute the initial CBC;
Calculate $G_c = V_c - E_c + F_c$ for the initial CBC;
**while** $G_c \neq G_d$ **do**
    Voxelize the bounding box with a larger
    resolution;
    Compute the CBC;
    Calculate $G_c = V_c - E_c + F_c$ for the CBC;
**end**
```

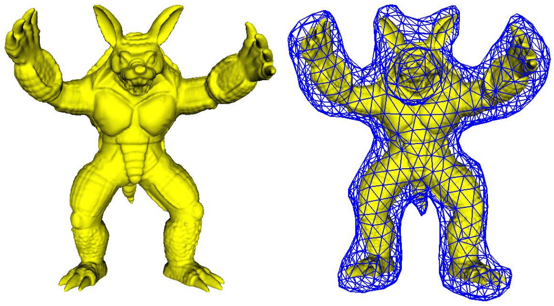

Fig. 10. Amarlido (left) and its CBC (right).

4. Results and Discussion

The automatic CBC generation algorithm developed in this paper is implemented with $VC++2005$ and $OpenGL$, and runs on the PC with 2.4 GHz Pentium IV CPU and 1.0GB memory. The generated results are listed in Fig. 10 to Fig. 15. Additionally, the data on the dense meshes and their CBC are shown in table 1. In this table, the second and third columns list the numbers of vertices of the dense meshes and their CBCs, respectively. The fourth column presents the similarity values between the dense meshes and its CBCs, calculated by the method developed in Ref. [11]. The last column shows the sparse factors of the CBCs (Eq. (1)).

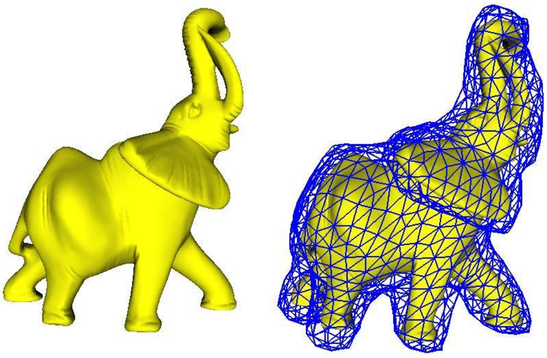

Fig. 11. Elephant (left) and its CBC (right).

Figs. 10 and 11 demonstrate two graphical models and their CBCs. The dense mesh model *Amarlido* in Fig. 10 has 130838 vertices. Its CBC has a very low sparse factor 0.01, and high similarity value 0.75 with the the original mesh. The model *elephant* in Fig. 11 contains 24955 vertices. Its CBC has the sparse factor 0.037 and higher similarity value 0.80. Generally speaking, the automatically generated CBCs have low sparse factor and high similarity with the original meshes, and by increasing the sparse factor, the similarity value will be improved.

The examples above and in section 5 validate the efficiency and effectiveness of the automatic CBC generation algorithm developed in this paper.

5. Applications

In sections 5.1 and 5.2, the automatically generated CBCs are employed in two applications, mesh deformation and subdivision surface fitting.

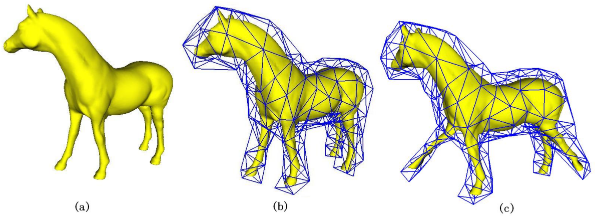

Fig. 12. Horse and its deformation. (a)Horse model; (b) its CBC; (c) deformation result.

5.1 *Mesh Deformation*

Volume-based deformation depends on a coarse cage bounding the mesh model to be deformed. The coarse bounding cage is usually generated interactively, and its generation is very tedious and time-consuming ([1]-[5]).

TABLE 1
DATA ON THE DENSE MESHES AND THEIR CBC.

Model names	Mesh vertices	Cage vertices	Similarity	Resolution	Sparse Factor
Amarlido(Fig. 10)	130838	1359	0.75	$16 \times 16 \times 16$	0.010
Elephant(Fig. 11)	24955	934	0.80	$16 \times 16 \times 16$	0.037
Horse(Fig. 12)	19851	159	0.71	$4 \times 8 \times 8$	0.008

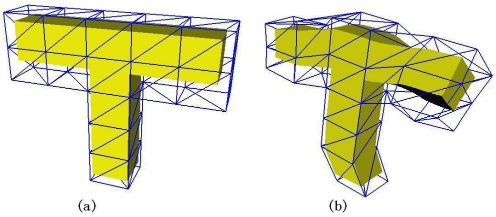

Fig. 13. T-shape and its deformation. (a.) T-shape and its CBC (#mesh vertices = 3762, #CBC vertices = 48), (b.) deformation result.

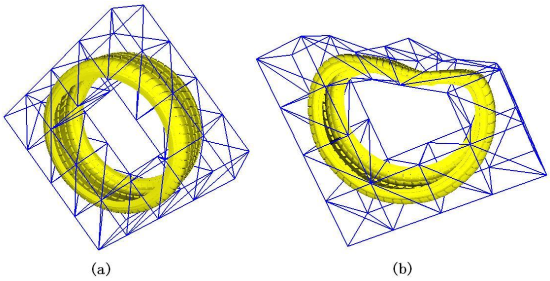

Fig. 14. Tyre and its deformation. (a.) Tyre model and its CBC (#mesh vertices = 12660, #CBC vertices = 48), (b.) the deformation result.

However, by the automatic CBC generation method developed in this paper, the CBC generation is fully automatic, and usually costs several seconds. More importantly, since the automatically generated CBCs can hold the topological structure and major geometric features of the original mesh model, they make the deformation results more desirable (Figs. 12- 14).

The model horse and its deformation result are illustrated in Fig. 12. Fig. 12b demonstrates the automatically generated CBC of the horse, which encloses the model tightly. Fig. 12c is its deformation result by manipulating the CBC.

Fig. 13a shows the model *T* with sharp features and its CBC. The CBC and the deformation result (Fig. 13b) hold the sharp features well.

Fig. 14 is a torus-like model and its deformation result. The model and its CBC are homeomorphic, so the effect of the adjustment of the CBC vertex is locally (Fig. 14b), not disturbing the part far away.

The deformation in this section is carried out by the *Green coordinates* method developed in Ref. [4].

5.2 *Subdivision Surface Fitting*

In subdivision surface fitting, the shape of the initial control mesh is important for good fitting [6]. A desirable control mesh can save the subdivision time greatly. In this section, we take the automatically generated CBC (Fig. 15b) as the initial control mesh to fit the dense mesh model in Fig. 15a by the method developed in Ref. [6]. Since the CBC captures the major geometric features of the original mesh model, just with one subdivision, we obtain the subdivision fitting surface with max-norm error 7.764×10^{-3}, where the original mesh model is normalized to $[-1, 1] \times [-1, 1] \times [-1, 1]$. Fig. 15c is the control mesh of the subdivision surface after one subdivision, and Fig. 15d shows the final fitting surface.

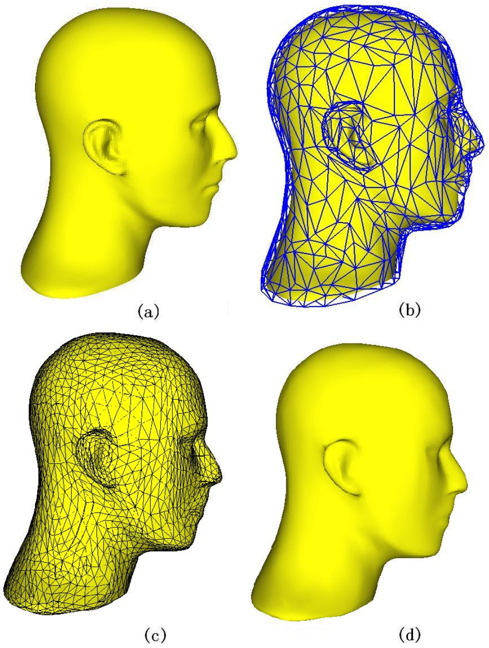

Fig. 15. Subdivision surface fitting. (a) Original mesh model (#vertices = 69279); (b) the automatically generated CBC is taken as the initial control mesh (#vert = 693); (c) the control mesh of the subdivision surface after one subdivision (#vert = 2766); (d) the subdivision fitting surface.

6. CONCLUSIONS

In this paper, we propose an automatic method to generate the coarse bounding box (CBC) for a given original dense mesh M. The original mesh is first voxelized, and the feature voxels intersecting with the mesh surface are identified. In the following, the outer and inner voxels are determined by the seed filling algorithm, generating a tri-value distance field defined on the voxels. Next, the outer faces of the feature voxels are extracted by checking the distance values of voxels, constituting an initial CBC. Finally, an improved mean curvature flow smoothing method is developed to smooth the initial CBC. The mean curvature flow method is so improved that it can keep the CBC out of the original mesh while smoothing.

By the automatic CBC generation algorithm, a CBC is usually generated in several seconds, depending on the amount of data of the original mesh. Thus, it alleviates the users' burden greatly. More importantly, as illustrated in our experimental results, the automatically generated CBCs can hold the topological structure and major geometric features of the original mesh model. Therefore, they work well in lots of applications, such as mesh deformation and subdivision surface fitting.

Acknowledgement

This paper is supported by 973 program of China (No. 2004CB719400), and NSF of China (No. 60736019).

References

[1] Ju T., Schaefer S., AND Warren J. Mean Value Coordinates for Closed Triangular Meshes. Procedings of SIGGRAPH 2005; 24: 561-566.

[2] Ju T., Schaefer S., AND Warren J. Mean Value Coordinates for Closed Triangular Meshes. Procedings of SIGGRAPH 2005; 24 : 561-566.

[3] Joshi P., Meyer M. AND Derose T. Harmonic Coordinates for Character Articulation. ACM Transaction on Graphics. Vol. 26, 3. 2007.

[4] Lipman Y., Levin D., AND Cohen-Or D. Green Coordinates. Procedings of SIGGRAPH 2008.

[5] Huang J., Chen L., Liu X.G., AND Bao H.J. Efficient Mesh Deformation Using Tetrandron Control Mesh. Procedings of ACM Solid and Physical Modeling 2008, 241-247.

[6] Suzuki H., Kanai T., AND Kimura F. Subdivision surface fitting to a Range of Points. Procedings of the Seventh Pacific Graphics International Conference 1999, 158-167.

[7] Ma W., Ma X., Tso S.K., AND Pan Z.G. A direct Approuch for Subdivision Surface Fitting a Dense Triangle Mesh. Computer Aided Design 2004 32 : 525-536.

[8] Gottschalk S., Lin M.C., AND Manocha D. OBBTree: A Hierarchical Structure for Rapid Intereference Detection. Procedings of SIGGRAPH 1996.

[9] Juan J.J. AND Segura R.J. Collision Dection between Complex Polyhedra. Computer & Graphics 2008; 32 : 402-411.

[10] Masuda H. Feature-preserving Deformation for Assembly Model. Computer-Aided Design & Applications 2007.

[11] Elad M., Tal A., AND Ar S. Content Based of VRML Objects - An Iterative and Interactive Approach. The 6th Eurograpics Workshop in Multimedia 2001.

[12] David L. Linear Algebra and Its Applications. Addison-Wesley, New York, 2000.

[13] Varadhan G., Krishnan S., Kim Y. J., Diggavi S., AND Manocha D. Efficient Max-Norm Computation and Reliable Voxelization. Procedings of Eurographics Symposium on Geometry Processing 2003

[14] Foley J., Dam A.V., Feiner S., Hughes J., Phillips R. Computer Graphics: Principles and Practice. 2nd Edition, Addison-Wesley, 1990.

[15] Desbrun M., Meyer M., Schröder P., AND Barr A. H. Implicit fairing of irregular meshes using diffusion and curvature flow. Procedings of ACM SIGGRAPH 1999: 317-324.