Anisotropic Meshing of Implicit Surfaces

Sergei Azernikov and Anath Fischer
Laboratory for CAD and LCE
Faculty of Mechanical Engineering
Technion - Israel Institute of Technology
Haifa, Israel 32000
mesergei@technion.ac.il

Abstract

This paper proposes a new grid-based method for adaptive anisotropic meshing of implicit surfaces. Grid-based methods are considered the major technique for implicit surface meshing, mainly due to their efficiency and simplicity. However, these methods suffer from a number of inherent drawbacks, resulting from the fact that the imposed Cartesian grid is generally not well adapted to the iso-surface, either in size or in orientation. To overcome the above obstacles, we propose a new implicit surface meshing method. The main idea of this method is first to construct a geometric field which is induced by the shape of the surface. This geometric field represents the natural directions and grid cell size for each point in $\mathbb{R}^3$. Then, the imposed volumetric grid is deformed toward the object's shape according to the produced geometric field. The iso-surface meshes can be extracted from the resulting adaptive grid by any conventional grid-based contouring technique, since the topology of the grid is not modified during the adaptation process. These meshes provide better approximation of the unknown surface and exhibit anisotropy, which is present in this surface.

1. Introduction

In nature, a surface is usually appears implicitly as a boundary between two environments. In synthetic solid modeling, this implicit form is used as well, since CSG operations with implicit models are rather simple. Recently, several researchers have shown that high quality implicit surfaces can be reconstructed from unorganized point clouds [9, 22] and polygonal meshes [28]. For other applications however, i.e. rendering, analysis and manufacturing, an explicit representation may be beneficial. An implicit representation is usually converted into an explicit one by iso-surface extraction or contouring. Therefore, numerous studies on iso-surface extraction have been published during the last two decades [6]. The existing contouring methods can be classified into two main categories: *volume-based* methods and *surface-based* methods.

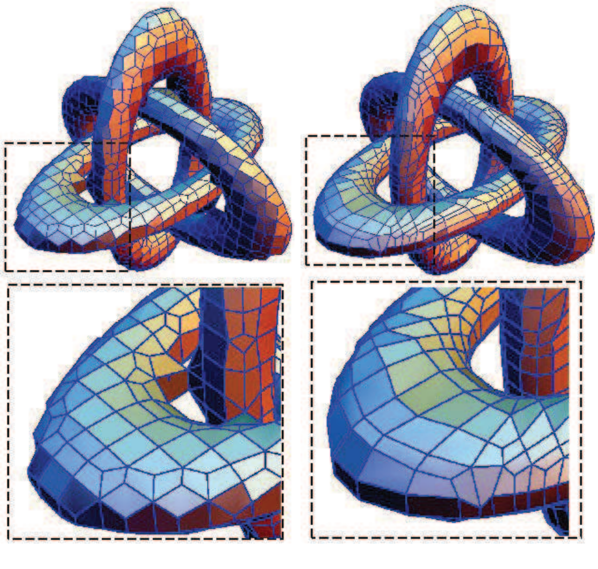


Figure 1. Adaptive anisotropic meshing of a complex implicit solid constructed from three disjoint rings: (a) iso-surface extracted from the uniform Cartesian grid 25x25x25 (1776 faces); (b) iso-surface extracted from the same grid after the proposed geometric adaptation (2374 faces).

Volume-based methods impose a structured grid of voxels on the domain. These voxels can be cubical, as in the original *Marching Cubes* algorithm [20], tetrahedral [5] or octahedral [8], uniform [5] or adaptive [25]. Then, intersection points are found between the surface and edges of the grid cells. These points represent the geometry of the surface. Roughly speaking, this geometry can be interpreted as a projection or mapping of the grid onto the surface. This mapping makes the volume-based methods efficient and robust, since the topology of the surface is dictated *a-priori* by the imposed grid. Moreover, *all* the portions of the iso-surface are detected. There is also an option for parallel computing acceleration, because each voxel can be processed independently.

Unfortunately, though efficient, volume-based methods often produce low quality meshes, since the imposed grid in general is not well adapted to the iso-surface shape. Many attempts have been made to solve this problem. The main research directions are 1) post-processing or *remeshing* of the low quality meshes [2, 23]; 2) surface-based methods [1]; and 3) grid adaptation [4, 21].

Surface-based methods start from a seed point on the surface and propagate the mesh by region growing. These methods produce high quality meshes because they consider the local differential geometry of the surface during the propagation process [1,12,18]. However, since the topology of the surface is recovered on-the-fly, dealing with complex high genus objects might by difficult. Consider, for example, the object shown in Figure 1, which is constructed from several disjoint parts. Extensive investigation of this problem can be found in [26].

The grid adaptation approach for implicit surface contouring was introduced by Moore and Warren [21]. In this work, the imposed grid vertices are perturbed in order to reduce the number of triangles produced by the Marching Cubes algorithm. However, this perturbation is indifferent to the surface geometry. Balmelli et al. [4] proposed to warp the grid according to an *importance map*. This map makes grid vertices attract to regions of interest chosen by the user. As a result, the extracted iso-surface mesh is denser in these regions and sparser elsewhere. The importance map constitutes a *scalar* metric field, which guides the adaptation of the grid. Since the field is scalar, the adaptation is *isotropic*. Recently, Tchon et al. [27] showed that employing an *anisotropic* geometric field for grid adaptation is very natural and effective. These authors proposed an effective and simple method to adapt a uniform volumetric grid to the triangulated surface. The adaptation is guided by the geometric field, which is constructed from the tensor of curvature and thickness of the solid bounded by the surface. The adapted grid tends to follow the shape of the surface and clearly reflects anisotropic properties of this shape. These results suggest that a geometric field is a very natural and general representation of a shape for applications, where anisotropic properties are important. Indeed, these properties must be considered during high quality surface meshing [14]. Therefore, we employ the anisotropic grid adaptation approach for adaptive meshing of implicit surfaces.

Our adaptation method is based on the one proposed by Tchon et al. [27], with several important differences due to a completely distinct application. Tchon et al. proposed their method in the context of hexahedral meshing of *volumes* bounded by a given polygonized surface. In contrast, we are interested in meshing implicit *surfaces*. Therefore, we must deal with a number of specific problems:

- How to construct the geometric tensor field from an implicitly defined surface?
- How to sample the produced tensor field when the region of interest is the surface only?

In the following sections we describe the proposed approach in detail and give answers to the stated questions.

2. The Approach

In the current work we propose to enhance the performance of grid-based implicit surface contouring methods by adapting the imposed volumetric grid with the anisotropic metric field induced by this surface shape. The motivating idea behind this approach is simple. In terms of signal processing, the imposed grid is a *sampling* of $\mathbb{R}^3$. For the homogeneous and isotropic Euclidean space, uniform sampling is optimal. In our case, however, the presence of the implicit surface introduces *curvature* into the space. As a result, the space is no longer Euclidean but rather Riemannian. Indeed, the uniform grid in Euclidean space is non-uniform in the Riemannian space. Consequently, this grid should be *warped* in order to provide uniform sampling of the anisotropic space.

The proposed method consists of four main phases, as shown in Figure 2. The process begins by constructing the *background mesh*. Then, the geometric tensor field is evaluated and propagated on this mesh. As a result, a metric tensor is set for each point in the problem domain. The grid is adapted geometrically by relaxing the vertex position, while edge length is calculated in the produced anisotropic metric. Finally, the iso-surface mesh is extracted using one of the standard grid-based contouring techniques.

This approach has the following important advantages:

- The geometric adaptation is shape-driven and not uniform or axis-aligned, as with octrees. Therefore, complex geometric features can be recovered with a lower number of voxels.
- The geometric adaptation only perturbs the grid's geometry, while maintaining the structured topology. That is, the adaptation can be used as a preprocessor for any existing grid-based contouring method.

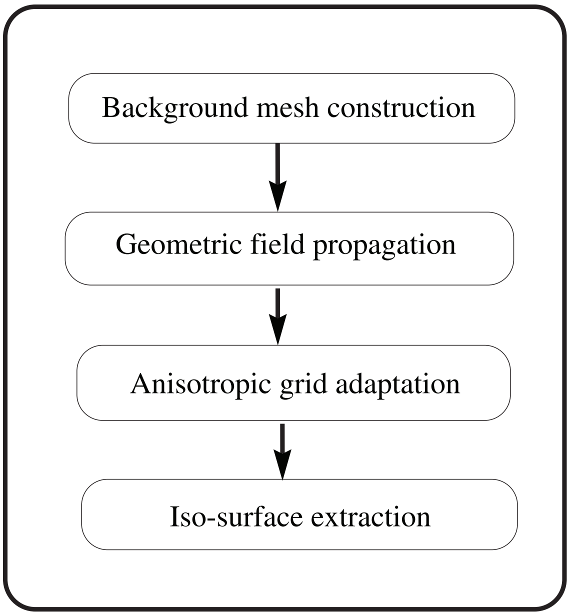

Background mesh construction
Geometric field propagation
Anisotropic grid adaptation
Iso-surface extraction

Figure 2. Flow chart of the proposed implicit surface meshing method.

- The meshes extracted from the adapted grids are of much higher quality than those extracted from the original uniform grids with the same number of voxels. Moreover, dual contouring [3, 17] of the adapted grids produces *all-quad, anisotropic* meshes.
- Recovery of *all* parts of the iso-surface, connected or not, is guaranteed. This is not trivial to achieve using surface-based methods [12].
- In analysis applications, the behavior of some physical phenomena often must be incorporated into the meshing process. This is supported naturally by the proposed approach, by applying metric intersection [14].

Since the geometric tensor field definition requires notation from the differential geometry of implicit surfaces, we start with a brief review of this notation.

3. Differential Geometry of Implicit Surfaces

Although the results of this section can be found in [16], we summarize them here for completeness. Consider $F : \mathbb{R}^3 \mapsto \mathbb{R}$ a scalar function defined over the domain of support Ω. Let S be a surface defined implicitly as $S = \{\forall \mathbf{x}(x, y, z) \in \Omega | F(\mathbf{x}) = 0\}$. Our goal is to determine local differential properties for $\forall \mathbf{x} \in S$. These properties are compactly encoded in the *Weingarten map* [13], or more intuitively in the *shape operator* $\mathcal{L}(\mathbf{v})$. This operator shows how the unit surface normal $\hat{\mathbf{n}}$ changes at the point $\mathbf{x} \in S$ in the tangent direction $\mathbf{v}$. The surface S unit normal $\hat{\mathbf{n}}$ at $\mathbf{x}$ is equal to the normalized gradient of F at $\mathbf{x}$,

$$\hat{\mathbf{n}}(\mathbf{x}) = \frac{\nabla F(\mathbf{x})}{\|\nabla F(\mathbf{x})\|}, \forall \mathbf{x} \in S \qquad (1)$$

Since the gradient field ∇F is defined over the whole domain Ω, it is possible to extend the unit normal field $\hat{\mathbf{N}} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ from S to Ω as follows,

$$\begin{aligned} \hat{\mathbf{N}}(\mathbf{x}) &= \frac{\nabla F(\mathbf{x})}{\|\nabla F(\mathbf{x})\|}, \forall \mathbf{x} \in \Omega, & (2) \\ \hat{\mathbf{N}}(\mathbf{x}) &= \hat{\mathbf{n}}(\mathbf{x}), \forall \mathbf{x} \in S. & (3) \end{aligned}$$

By definition, $\mathcal{L}(\mathbf{v})$ is a *covariant derivative* of the normal field in direction $\mathbf{v}$,

$$\mathcal{L}(\mathbf{v}) = D\hat{\mathbf{N}} \cdot \mathbf{v} = D\left(\frac{\nabla F}{\|\nabla F\|}\right) \cdot \mathbf{v}. \qquad (4)$$

The *Hessian* matrix of F is defined as,

$$\mathcal{H}F \triangleq \begin{pmatrix} \frac{\partial^2 F}{\partial x \partial x} & \frac{\partial^2 F}{\partial x \partial y} & \frac{\partial^2 F}{\partial x \partial z} \\ \frac{\partial^2 F}{\partial y \partial x} & \frac{\partial^2 F}{\partial y \partial y} & \frac{\partial^2 F}{\partial y \partial z} \\ \frac{\partial^2 F}{\partial z \partial x} & \frac{\partial^2 F}{\partial z \partial y} & \frac{\partial^2 F}{\partial z \partial z} \end{pmatrix} \qquad (5)$$

It is easy to show that,

$$\mathcal{L}(\mathbf{v}) = \frac{1}{\|\nabla F\|} \mathcal{H}F \cdot \mathbf{v}. \qquad (6)$$

Thus, the matrix form of the shape operator $\mathcal{L}(\mathbf{v})$ at point $\mathbf{x}$ is given by,

$$l(\mathbf{x}) = \frac{1}{\|\nabla F(\mathbf{x})\|} \begin{pmatrix} \hat{\mathbf{b}}_1^T \mathcal{H}F(\mathbf{x}) \hat{\mathbf{b}}_1 & \hat{\mathbf{b}}_1^T \mathcal{H}F(\mathbf{x}) \hat{\mathbf{b}}_2 \\ \hat{\mathbf{b}}_2^T \mathcal{H}F(\mathbf{x}) \hat{\mathbf{b}}_1 & \hat{\mathbf{b}}_2^T \mathcal{H}F(\mathbf{x}) \hat{\mathbf{b}}_2 \end{pmatrix} \qquad (7)$$

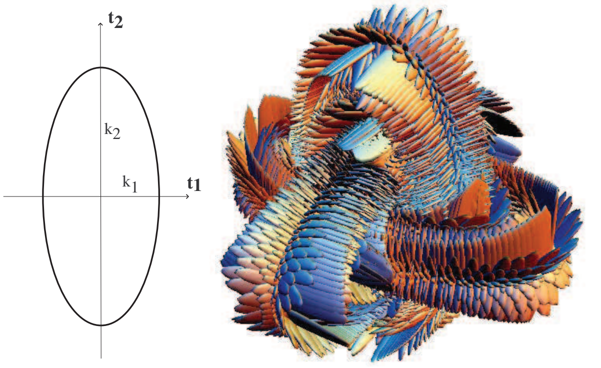
where $(\mathbf{b}_1, \mathbf{b}_2)$ is an arbitrary basis of the tangent space of F at $\mathbf{x}$. Having $l(\mathbf{x})$ in hand we can calculate all the required differential properties at $\mathbf{x}$, as summarized in Table 1.

Table 1. Differential properties of F at $\mathbf{x}$ that can be extracted from the shape operator matrix form $l(\mathbf{x})$.

Property	Expression
principal curvatures $\kappa_{1,2}$	eigenvalues of $l(\mathbf{x})$
principal directions $\hat{\mathbf{t}}_{1,2}$	eigenvectors of $l(\mathbf{x})$
Gaussian curvature K	$det[l(\mathbf{x})]$
mean curvature H	$\frac{1}{2}trace[l(\mathbf{x})]$

These properties set a *tensor of curvature* $\mathcal{C}(x)$,

$$\mathcal{C}(\mathbf{x}) = \begin{pmatrix} \hat{\mathbf{t}}_1 & \hat{\mathbf{t}}_2 \end{pmatrix}^T \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{t}}_1 & \hat{\mathbf{t}}_2 \end{pmatrix} \qquad (8)$$



(a) (b)

Figure 3. Tensors of curvature can be efficiently visualized as ellipsoids: (a) tensor of curvature represented as ellipse; (b) tensors of curvature evaluated for the rings model.

This tensor is a core of the geometric tensor defined in Section 4. Figure 3 presents evaluated tensors of curvature on torus.

4. Geometric Tensor Field

Using the results of the previous section, we are able to formally define the geometric field as a *Riemannian metric* induced by the shape of the surface S on the domain Ω. For each point $\mathbf{x} \in \Omega$ this metric defines a tensor $\mathcal{M}_{3\times3}$ as follows:

$$\mathcal{M} = \mathcal{R}\Lambda\mathcal{R}^{-1} \tag{9}$$

where $\mathcal{R}$ is an orthonormal rotation matrix and Λ is a diagonal scaling matrix. This tensor defines the mapping of the desired hexahedral voxel to a unit axis-aligned cube for $\forall\mathbf{x} \in \Omega$ [14].

As in [27], we associate the lower 2×2 minor of $\mathcal{M}$ with the tensor of curvature $\mathcal{C}$ of the surface. But for the normal direction we give a different interpretation, which is more appropriate in our context. Tchon et al. [27] deduce the target size of an element at $\mathbf{x}$ in the normal direction from the local *thickness* of the solid enclosed by S. In terms of the normal field $\hat{\mathbf{N}}$ produced by S, this is twice the distance from $\mathbf{x}$ to the closest singular point of $\hat{\mathbf{N}}$. This thickness is approximated from the digital skeleton of the solid bounded by S. In our case, however, S does *not* necessarily bound a solid, and in any case we are interested only in a *narrow band* surrounding S and not in the bounded volume. In our implementation, the size of this band is set to twice the surface-voxel intersection evaluation precision ϵ. The geometric tensor components can be now rewritten as follows:

$$\mathcal{R} = \begin{pmatrix} \hat{\mathbf{n}} & \hat{\mathbf{t}}_1 & \hat{\mathbf{t}}_2 \end{pmatrix}^T \tag{10}$$

$$\Lambda = \begin{pmatrix} 1/4\epsilon^{-2} & 0 & 0 \\ 0 & \alpha\kappa_1^2 & 0 \\ 0 & 0 & \alpha\kappa_2^2 \end{pmatrix} \tag{11}$$

where $\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2$ are the principal directions which span the tangent space of the surface, $\hat{\mathbf{n}}$ is the normal vector and α is a coefficient which allows control of the introduced anisotropy level.

5. Background Mesh Construction

For the geometric field to be represented faithfully, it should be appropriately sampled on the background mesh. Sampling it on a dense uniform grid is possible, but such an approach is extremely memory-intensive. A better solution is to sample this field *adaptively*. Octrees have been used for adaptive sampling of scalar distance fields by Frisken et al. [15], and geometric tensor fields by Tchon et.al [27]. Octrees are very suitable for adaptive field representation since they provide local refinement and efficient point localization. In our implementation, the octree is constructed by the *top-down* approach. With this approach, the process starts from the root voxel, which occupies the whole domain of support Ω. This voxel v is then subdivided recursively until the minimal allowed size is not reached and one of the following conditions hold:

1. the voxel v contains more than one disconnected component of S,
2. the size of v is bigger than the target size at v.

As can be seen in Figure 4, these conditions occur in regions where the gradient field *divergence* $div(\mathbf{N})$ is high.

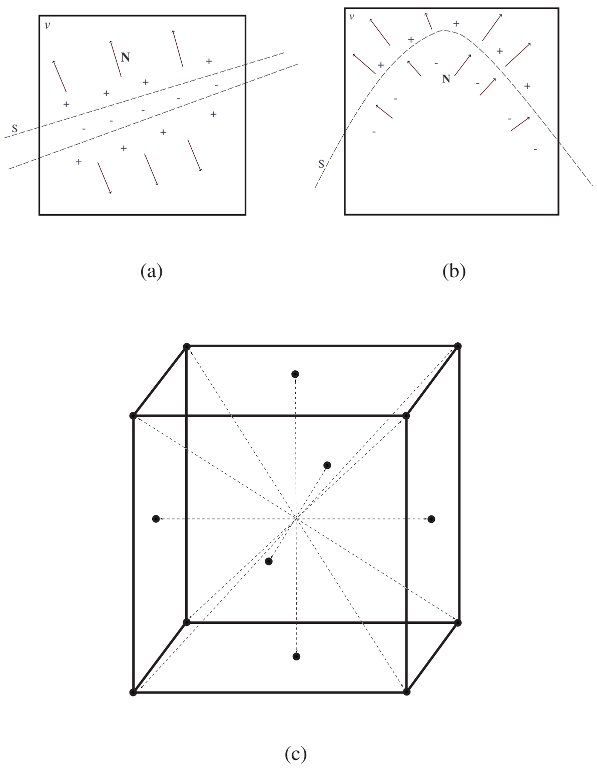
In order to evaluate the behavior of S in v, we sample the implicit function F and its gradient field $\mathbf{N}$ at 10 pairs of points $P = \{p_i, p_j\} \in v$, as shown in Figure 4(c). Then, for each pair, the change of sign of the function and the gradient vector rotation are calculated as follows,

$$sign_{ij} = F(p_i) \cdot F(p_j), \tag{12}$$

$$\alpha_{ij} = \langle \mathbf{N}(p_i), \mathbf{N}(p_j) \rangle. \tag{13}$$

If these values are positive, it is assumed that v does not contain any portion of S. A negative α_{ij} means that there is a *singular point* of the gradient field $\mathbf{N}$ inside v, and thus v should be subdivided. Otherwise, if only $sign_{ij}$ is negative, the target size inside v is evaluated. The target size is the minimal radius of curvature $\rho = 1/|\kappa_{1,2}|$ of $S \in v$. The procedure described above is summarized in Algorithm 1.

Figure 5 presents the octree created for the rings model. After the octree is created, the geometric tensors are constructed in the leaf voxels which contain a portion of the



(a) (b)

(c)

Figure 4. Surface-voxel intersection configurations which cause refinement: (a) v contains more than one disconnected component of S; (b) size of v is bigger than the target size at v; (c) sampling point pairs in v.

iso-surface S. The tensor construction is done following the procedure described in Section 4. However, there are a number of special cases that should be treated differently:

- In the isotropic regions, where principal directions are undetermined, an arbitrary orthonormal tangent basis is chosen.
- In the flat regions where the curvature vanishes, the voxel target size in tangent directions is set to the radius of the domain Ω circumsphere R.
- In the voxels where the gradient field is discontinuous, the geometric tensors are not evaluated and set later during the metric smoothing.

In the remaining voxels, geometric tensors are initialized to a diagonal matrix,

$$\mathcal{M} = \begin{pmatrix} 1/R^2 & 0 & 0 \\ 0 & 1/R^2 & 0 \\ 0 & 0 & 1/R^2 \end{pmatrix} \qquad (14)$$

Algorithm 1 The background octree construction.

```
CONSTRUCTBACKGROUNDOCTREE($v$) {$v$ is the root voxel.}
if SIZE($v$) = MINSIZE then
  {MINSIZE is set by the user.}
  RETURN()
end if
if GRADIENTFIELDCHANGESDIRECTION($v$) then
  SUBDIVIDE($v$)
else if DISTANCEFIELDCHANGESSIGN($v$) then
  ESTIMATETARGETSIZE($v$){Perform curvature analysis.}
  if SIZE($v$) > TARGETSIZE($v$) then
    SUBDIVIDE($v$)
  end if
end if
for $i \leftarrow 0 : 7$ do
  CONSTRUCTBACKGROUNDOCTREE(SON($v, i$))
end for
```

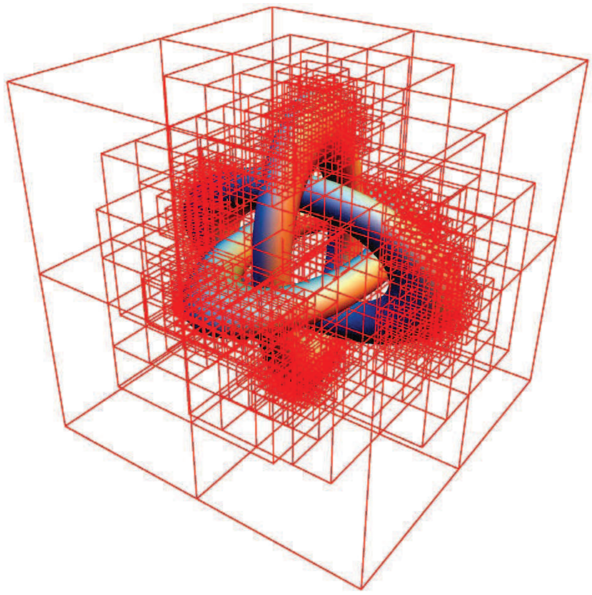


Figure 5. The constructed octree for the rings model contains 3032 nodes, with maximal depth 6.

Then, the geometric field is propagated from the iso-surface to the neighboring voxels by iterative Laplacian smoothing, as in [27],

$$\mathcal{M}_i^{n+1} = \mathcal{M}_i^n + \frac{\sum(\mathcal{M}_i^n - \mathcal{M}_j^n)/l_{ij}}{\sum 1/l_{ij}} \qquad (15)$$

where summation is performed on the face neighboring voxels j, and l_{ij} is the distance between the centers of voxels i and j. Since we are interested in propagating the field into a *narrow band* surrounding S, we apply only two iterations of the above operator.

6. Anisotropic Grid Adaptation

The grid adaptation is performed by relaxation or iterative Laplacian smoothing in the metric field, described in Section 4. Grid boundaries must be constrained in order to prevent shrinkage. In our implementation, these boundaries are projected onto a *circumsphere* of the domain Ω. This technique provides maximum flexibility, since the grid boundary vertices can travel freely on the sphere and are not constrained to the specific cube sides, corners or edges, as in [27] and [4] . Then, for each grid vertex $\mathbf{v}_i$ the optimal location is calculated by the standard iterative scheme:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + w\frac{\sum l_{ij}^{\mathcal{M}}/l_{ij}(\mathbf{v}_j^n - \mathbf{v}_i^n)}{\sum l_{ij}^{\mathcal{M}}/l_{ij}} \tag{16}$$

where summation is applied on the neighboring vertices $\mathbf{v}_j$, w is a relaxation factor, which is usually between 0 and 1 (0.5 in our implementation), l_{ij} is a Euclidean distance between the vertices $\mathbf{v}_i$ and $\mathbf{v}_j$ and $l_{ij}^{\mathcal{M}}$ is the *metric* length of the edge $\{\mathbf{v}_i, \mathbf{v}_j\}$ between these vertices. This length is calculated by integrating the metric dot product along the edge,

$$l_{ij}^{\mathcal{M}} = \int_0^1 \sqrt{(\mathbf{v}_j - \mathbf{v}_i)^T\mathcal{M}(\mathbf{x}(t))(\mathbf{v}_j - \mathbf{v}_i)}dt \tag{17}$$

where $\mathbf{x}(t) = \mathbf{v}_i + (\mathbf{v}_j - \mathbf{v}_i)t$ and $\mathcal{M}(\mathbf{x}(t))$ is the metric tensor at $\mathbf{x}$. This tensor is set to the tensor associated with the background octree voxel which contains $\mathbf{x}$.

Figure 6 demonstrates a grid adaptation for the rings model shown in Figure 1.

7. Iso-surface Mesh Extraction

In this phase, the iso-surface mesh is extracted from the adaptive grid produced in the previous phase. Since the grid is adapted with an anisotropic metric, the extracted mesh should have anisotropic properties. Traditional contouring techniques [5, 20] produce triangular or mixed meshes as shown in Figure 7(b). However, for exploiting the anisotropy of a surface, employing quads is more appropriate [2]. Recently, *dual contouring* techniques have been proposed [3, 17]. In the current work, a modified dual contouring method is used. This method starts by calculating zero-crossing points and their normals on grid edges, as in [17]. Then, each iso-surface voxel is classified as smooth or sharp according to the scheme proposed by Kobbelt et al. [19]. In the smooth regions, the estimated dual vertex is located in the centroid of the voxel intersection points. Afterwards, this vertex is projected by the Newton-Raphson method onto the surface [23]. For the sharp voxels, the dual vertex is located so as to minimize quadric error [17].

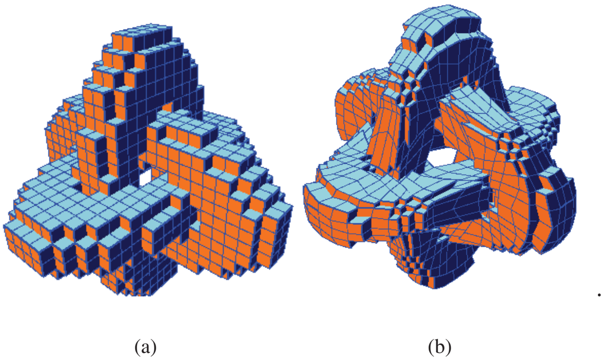


Figure 6. Grid adaptation for the rings model: (a) initial grid 25x25x25 (shown 1752 voxels which contain a portion of the iso-surface), (b) after 100 relaxation iterations (2450 voxels)

Applying the contouring method described above on structured grids produces *all-quad* meshes, as can be seen in Figure 7(c). Unfortunately, these meshes often contain topologically redundant faces, which can be collapsed in order to improve the overall mesh quality and to reduce the size of the mesh (see Figure 7(d)). The candidates for collapsing are those faces that have a pair of *opposite* vertices with valence 3. We create a virtual edge between these vertices and push the resulting edges into a priority queue. The priority is given by the length of the edge. Afterwards, the edges are popped sequentially from the queue and collapsed if both vertices still have valence 3. For dual contouring, the vertices are classified into three categories [19] : (a) smooth, (b) sharp edge and (c) sharp corner. If both vertices of the collapsed virtual edge are corners, the edge cannot be collapsed. If one of these vertices is sharp, the new vertex is located at the sharp vertex position. Otherwise, the new vertex is located at the middle of the collapsed edge.

8. Results and Analysis

The proposed method was implemented in C++ using OpenGL for rendering. All the examples presented in this paper were produced on a mobile Intel Centrino 1.5GHz, 512MB RAM, running WindowsXP.

Figure 1 presents a model constructed from three disjoint rings. This example demonstrates how the proposed grid adaptation method helps resolve topological problems. The adaptation process is applied on a uniform Cartesian

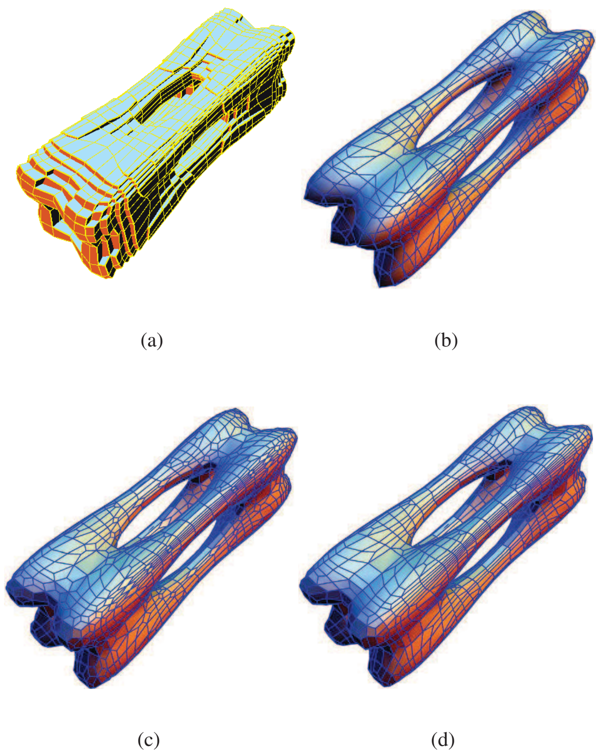


Figure 7. Tangle cube surface : (a) adaptive grid (shown 2240 voxels); (b) iso-surface extracted from the adaptive grid with marching cubes (2240 faces); (c) iso-surface extracted from the adaptive grid with dual contouring (2248 faces); (d) dual contour after decimation (1950 faces).

grid 25x25x25, and performed by 100 relaxation iterations. The time required for the geometric field construction was 3 seconds, and the adaptation took 120 seconds. These times are typical for all presented examples.

Figure 7 shows the tangle cube surface [7], defined by the equation,

$$x^4 - 5x^2 + y^4 - 5y^2 + z^4 - 5z^2 + 11.8 = 0.$$

This surface is stretched in order to introduce anisotropy. We apply two grid-based contouring techniques in order to extract this surface: marching cubes and dual contouring. The best result is achieved after decimation of the dual contour.

Figure 8 presents the chair surface [7] described by the equation,

$$(x^2+y^2+z^2-ak^2)^2-b(((z-k)^2-2x^2)((z+k)^2-2y^2) = 0.$$

Using this surface we demonstrate a correspondence between the principal directions and the edges of the mesh. In addition, the convergence of the adaptive mesh normals to the implicit surface gradients is tested by iteratively applying Catmull-Clark scheme [10] on the produced mesh. The new vertices are projected onto the surface, and normal directions are recalculated for each vertex. These directions converged to the implicit surface gradients (under 0.1 degree error) after three iterations of this subdivision-projection.

Another interesting surface [7] shown in Figure 9, is described by the next equation,

$$\begin{aligned}&-3x^8 - 3y^8 - 2z^8 + 5x^4y^2z^2 + 3x^2y^4z^2\\&-4(x^3+y^3+z^3+1) + (x+y+z+1)^4 + 1 = 0.\end{aligned}$$

This surface contains thin non-axis aligned tubes, which would require a very fine Cartesian grid. However, these tubes can be correctly reconstructed from the adapted grid (Figure 9(d)).

Figures 10 and 11 present applicability of the proposed method on different types of implicit surfaces. Figure 10 shows an R-function [24] model of an airplane. This example demonstrates how the grid is adapted to the fine features of the implicit surface.

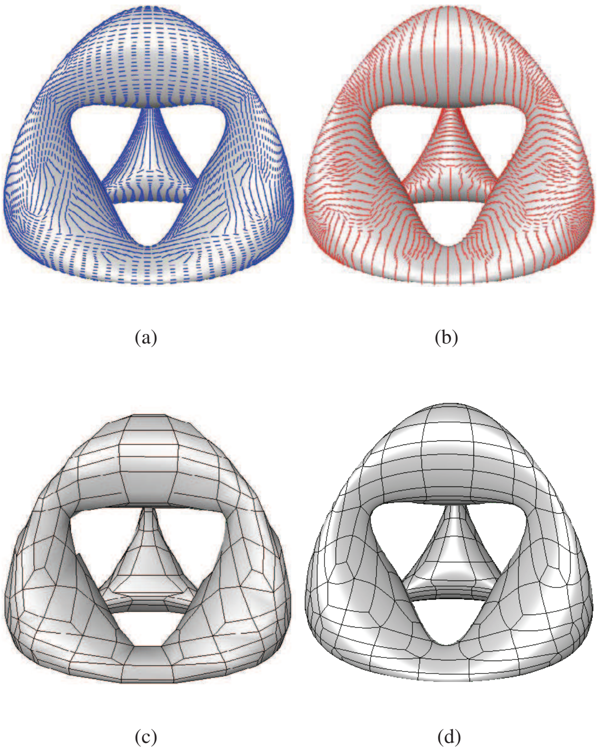
Adaptive reconstruction of a multi-level partition of unity (MPU) [22] implicit surface is shown in Figure 11. This model is reconstructed from cloud of points, scanned from a human ball joint bone [11].

9. Conclusions and Future Work

In the current paper, a new grid-based method for implicit surface meshing has been proposed. The method is based on the grid adaptation concept and can be used as a preprocessor for any grid-based iso-surface extraction algorithm. The meshes extracted from the adaptive grids by dual contouring are all-quad and exhibit anisotropy, induced by the iso-surface shape. This makes the proposed method suitable for numerical simulation applications, where high quality anisotropic meshes are required.

Implicit surface tilers usually face difficulties when two distinct parts of the surface come close. In order to deal with these difficulties, with the proposed approach, voxels near the surface are squeezed in the surface normal direction. This is insufficient to guarantee topological correctness; however, it works well in practice as demonstrated in the paper (see Figure 1).

The main limitation of the proposed method is its high computational cost. For numerical simulation, several minutes is a reasonable amount of time for the mesh generation phase. For the interactive visualization, however, the running time of the proposed method can be significant. But, since the optimal location for a vertex of the grid on each



(a)

(b)

(c)

(d)

Figure 8. Principal directions on the chair surface: (a) minimal curvature directions; (b) maximal curvature directions; (c) anisotropic mesh (556 faces); (d) Catmull-Clark subdivision surface [10].

relaxation iteration can be evaluated independently, accelerating the proposed method with parallelism shows great potential.

Currently, the proposed method was tested on modeled algebraic surfaces and multi-level partition of unity implicits (MPU) [22]. In the future, we plan to apply the proposed method on discrete volumetric data. We would also like to explore applications of the grid adaptation method to the direct hexahedral meshing of this data.

References

[1] S. Akkouche and E. Galin. Adaptive implicit surface polygonization using marching triangles. In *Computer Graphics Forum*, volume 20(2), pages 67–80. Blackwell Publishing, 2001.

[2] P. Alliez, D. Cohen-Steiner, D. Devillers, B. Lévy, and M. Desbrun. Anisotropic polygonal remeshing. In *Proceedings of ACM SIGGRAPH 2003*, volume 22(3) of *ACM Transactions on Graphics*, pages 485–493. ACM Press, 2003.

[3] S. Azernikov, A. Miropolsky, and A. Fischer. Surface reconstruction of freeform objects based on multiresolution volumetric method. In *8th ACM Symposium on Solid Modeling and Applications*, pages 115–126, Seattle, WA, USA, June 2003.

[4] L. Balmelli, C. J. Morris, G. Taubin, and F. Bernardini. Volume warping for adaptive isosurface extraction. In *Proceedings of the 13th IEEE Visualization 2002 Conference (VIS-02)*, pages 467–474, Piscataway, NJ, 27– 2002. IEEE Computer Society.

[5] J. Bloomenthal. An implicit surface polygonizer. In *Graphics Gems IV*, pages 324–349. Academic Press, Boston, 1994.

[6] J. Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufman Publishers, 1997.

[7] P. Bourke. `http://astronomy.swin.edu.au/~pbourke/surfaces/`.

[8] H. Carr, T. Theußl, and T. Möller. Isosurfaces on optimal regular samples. In *Proceedings of the symposium on Data visualisation 2003*, pages 39–48. Eurographics Association, 2003.

[9] J. C. Carr, R. K. Beatson, C. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *SIGGRAPH 2001, Computer Graphics Proceedings*, Annual Conference Series, pages 67–76. ACM Press / ACM SIGGRAPH, 2001.

[10] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10:350–355, spline 1978.

[11] Cyberware. `http://www.cyberware.com`.

[12] B. R. de Araujo and J. A. P. Jorge. Curvature dependent polygonization of implicit surfaces. In *SIBGRAPI*, 2004.

[13] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Inc., 1976.

[14] P. J. Frey and P.-L. George. *Mesh Generation Application to Finite Elements*. HERMES Science Publishing, 2000.

[15] S. F. Frisken, R. N. Perry, A. P.Rockwood, and T. R. Jones. Adaptively sampled distance fields: A general representation of shape for computer graphics. In *Siggraph 2000, Computer Graphics Proceedings*, Annual Conference Series, pages 249–254. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[16] J. F. Hughes. Differential geometry of implicit surfaces in 3-space a primer. Technical Report CS-03-05, Brown University, 2003.

[17] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. *ACM Transactions on Graphics*, 21:339–346, 3 2002.

[18] T. Karkanis and A. J. Stewart. Curvature-dependent triangulation of implicit surfaces. *IEEE Computer Graphics and Applications*, 22(2):60–69, March 2001.

[19] L. P. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel. Feature-sensitive surface extraction from volume data. In *SIGGRAPH 2001 Conference Proceedings*, pages 57–66. ACM SIGGRAPH, 2001.

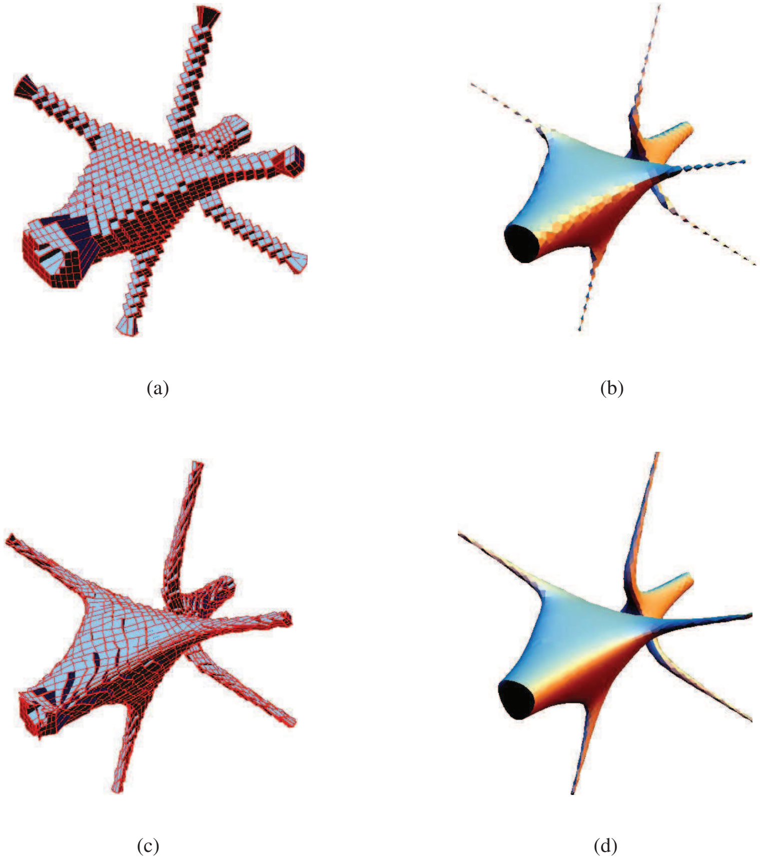


Figure 9. Tubey surface [7]: (a) initial grid 25x25x25 (shown 1491 voxels); (b) dual contour extracted from the initial grid (1449 faces); (c) adaptive grid (1964 voxels); (d) dual contour extracted from the adaptive grid and decimated (1704 faces).

[20] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algoritm. *Computer graphics*, 21(4):163–168, 1987.

[21] D. Moore and J. Warren. Mesh displacement: An improved contouring methods for trivariate data. Technical Report COMP TR91-166, Department of Computer Science, Rice University, P.Ø. Box 1892, Houston, TX 77251-1892, September 1991.

[22] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *Proceedings of ACM SIGGRAPH 2003*, volume 22(3) of *ACM Transactions on Graphics*, pages 463–470. ACM Press, 2003.

[23] Y. Ohtake and A. G.Belyayev. Dual/primal mesh optimization for polygonized implicit surfaces. In *Solid Modeling and Applications*, Saarbrücken, Germany, June 2002.

[24] A. Pasko, V. Savchenko, V. Adziev, and A. Sourin. Function representation in geometric modelling: concept, implementation and applications. *The Visual Computer*, 11(8):429–446, 1995.

[25] R. Shekhar, E. Fayyad, R. Yagel, and J. F. Cornhill. Octree-based decimation of marching cubes surfaces. In *Proceedings of the Conference on Visualization*, pages 335–344, Los Alamitos, 27– 1996. IEEE.

[26] B. T. Stander and J. C. Hart. Guaranteeing the topology of an implicit surface polygonization for interactive modeling. *Computer Graphics*, 31(Annual Conference Series):279–286, 1997.

[27] K.-F. Tchon, M. Khachan, F. Guibault, and R. Camarero. Constructing anisotropic geometric metrics using octrees and skeletons. In *12th International Meshing Rountable*, pages 293–304, September 2003.

[28] G. Yngve and G. Turk. Robust creation of implicit surfaces from polygonal meshes. *IEEE Transactions on Visualization and Computer Graphics*, 8(4):346–359, 2002.

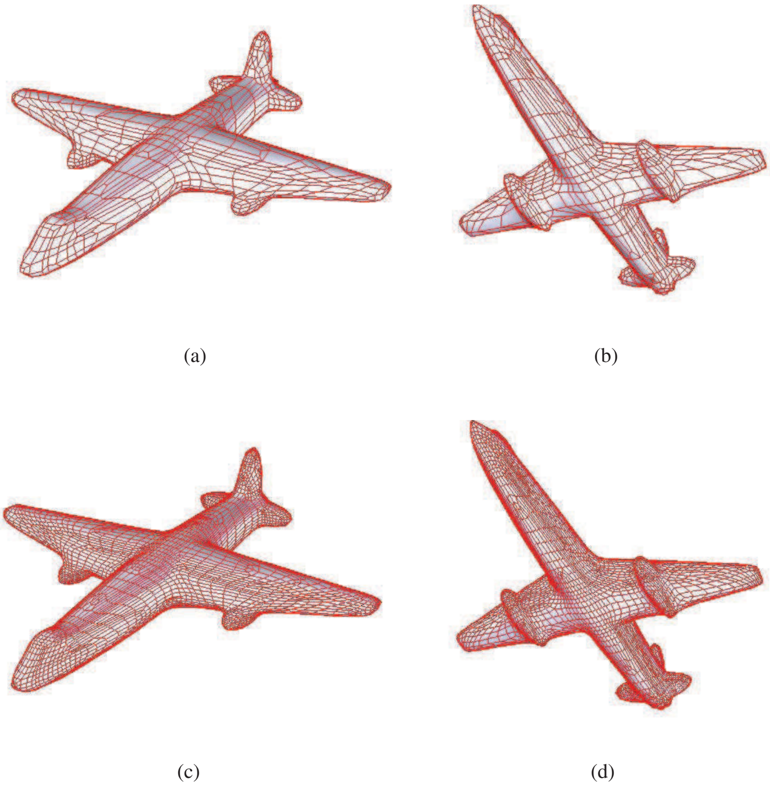


Figure 10. Airplane R-function model: (a),(b) anisotropic mesh extracted from the adaptive grid (2900 faces); (c),(d) anisotropic mesh subdivided with Catmull-Clark scheme [10].

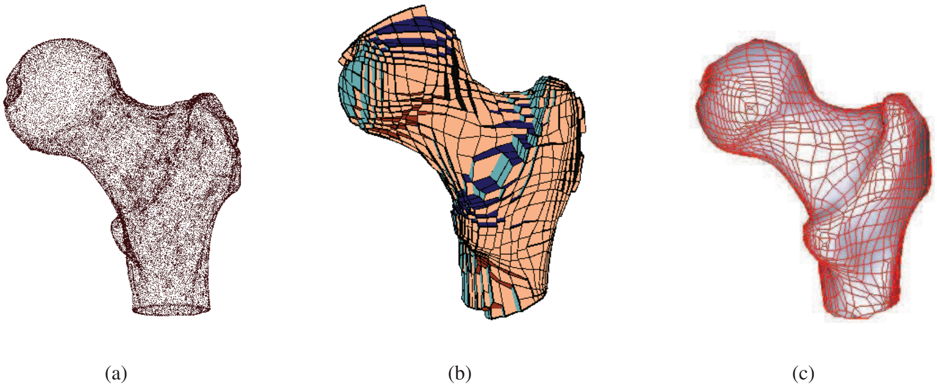


Figure 11. Anisotropic meshing of MPU [22]: (a) cloud of points scanned from ball joint (36588 points); (b) adaptive grid (2321 voxels); (c) dual contour extracted from the adaptive grid (2023 faces)