

Polygonal Mesh Extraction From Digital Voxel Art

Carlos Eduardo Vaisman Muniz, Anselmo Montenegro, Marcos Lage, Cristina Nader Vasconcelos
Instituto de Computação
UFF - Universidade Federal Fluminense
Niterói - RJ, Brazil
www.ic.uff.br

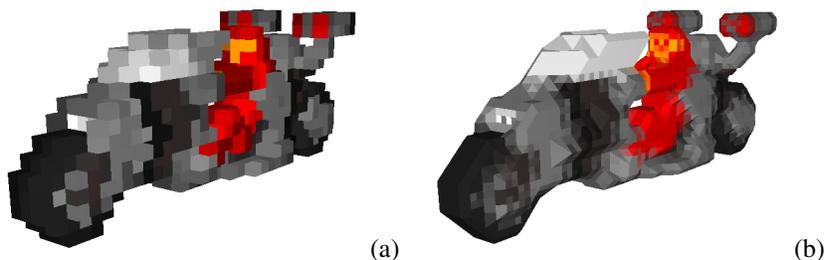


Fig. 1. Example of digital voxel art obtained from [1]. (a) Original volumetric data displayed in a voxel editor. (b) Polygonal mesh from it extracted by our method.

Abstract—This work presents a method to extract polygonal surfaces from volumetric models created by artists, proposing a way of using voxel modeling tools to build B-Rep models.

The volumetric data created by voxel editors usually contain topological features that do not describe solid structures. Hence, the main objective of this work is to solve the problem of extracting triangle meshes from volumes that contain these topological problems.

In order to extract surfaces successfully, a methodology was conceived to resample any volumetric model, in a way that it is possible to reconstruct a tridimensional manifold that can be polygonized without generating a surface with gaps or topological problems.

The meshes generated by this technique have good properties, satisfying some of the main criteria used to measure the quality of meshes, such as aspect ratio, smoothness and skewness.

Keywords—Mesh extraction; Voxel Art; Polygonization of Volumetric Data; Mesh reconstruction;

I. INTRODUCTION

Voxel art is a form of digital art in which the details and the geometry of a digital model are specified by means of voxels and its properties [2]. A model depicted by voxel art is usually described as a tridimensional array of voxels representing a uniform grid. An attribute function defines which voxel in the grid belongs to the geometric support of the volume by means of a binary valued density function which can be understood as a characteristic function. Other properties associated to each voxel can also be defined by means of functions as, for example, a color function that associates a *rgba* color to each voxel that defines the shape of the model. Empty spaces can be associated to a full transparent color. Figure 1a shows an example of voxel art.

Such form of digital art was present in several games in the end of the 90's as an alternative solution for the representation

of 3D objects which were composed of only one material and that could be manipulated using euclidean transformations [3]. The use of such objects made possible to run 3D games at that time in computers with no hardware acceleration. The use of voxel based representation fell into a kind of ostracism as soon as hardware accelerated graphics APIs such as OpenGL and Direct3D became available in the twenties, together with the advent of programmable and parallel GPUs. Further on, it was much easier and efficient to represent and deal with textured polygonal meshes than with volumetric data usually relying on raycasting algorithms for visibility and illumination computation. Thus, nowadays, the use of voxel art has become a form of art used by few enthusiasts although some games have nostalgically revived such aesthetics as, for example, 3D Dot Game Heroes by Silicon Studio of Japan [4].

The problem we definitely deal with here is to extract a textured mesh representation from a voxel art model produced in voxel editors. Such tools enable the user to edit each part of the volume manually, voxel by voxel, or using techniques for voxel carving. In this work the texture is inferred only from the colors of the voxels in the volumetric data which means that no texture atlas is built in the process.

Mesh extraction from volumetric data has been thoroughly investigated in the literature as it will be shown in the related works section. Nevertheless, differently from other graphical objects considered in computer graphics, voxel art has its own idiosyncrasies that have led to the way we tackle the mesh extraction problem in this work. Aliasing and discontinuities which are usually seen as an alien and undesirable artefact in many models produced by geometric modeling and computer graphics are an inherent aspect of voxel digital art. These aspects arise naturally when the artist uses his expression to

describe the shapes and details in the boundary of the volume. Hence, we may expect that aliasing and discontinuities will not be outlier cases in the modeling process associated to voxel art but a natural and frequent phenomena that arises when the artist tries to express his ideas of shape by means of simple volumetric elements, similarly to someone that paints an art picture in any pixel drawing tool.

Mesh extraction in the presence of topological degeneracies is a problem not very explored in the literature. Here we present a new technique to deal with such problem that we have tested and considered very satisfactory, producing quite good results.

This work has two main relevant contributions: (1) it proposes a new method to resample volumetric models in such a way that it is possible to reconstruct 3-manifolds which can be later polygonized without introducing any artifact such as holes in the surface or other topological problems usually seen in conventional techniques such as Marching Cubes [5] and its variations as in [6]. (2) we present an algorithm that generates a triangle mesh from volumetric data which is an orientable combinatorial manifold that is consistent with the geometry of the original voxel model. Besides, the mesh produced by the method is robust considering important mesh quality metrics.

This work is organized as follows: section 2 describes the related works associated to our proposal; in section 3 we present some theoretical fundamentals about combinatorial topology used to describe our method; section 4 describes the method in details and section 5 presents results and analysis; finally, in section 6 we present a conclusion and possible works to be pursued in the future.

II. RELATED WORKS

Marching Cubes [5] is the most known method to extract a polygonal surface from volumetric data. It scans the whole model creating triangles at cubic regions that intersect the surface that delimits the model. This method uses a lookup table that describes how these triangles are generated according to the configuration of each cubic region. The signal of the implicit function describing the surface at the vertices of each cubic region determines the local triangulation configuration associated to the mesh that will be created. The lookup table of Marching Cubes has ambiguous configurations which may create cracks in the generated mesh. The work in [6] has implemented a lookup table from [7] that prevents that problem from happening, generating polygonal models that are manifold for any input data. Marching Cubes has received numerous extensions such as Extended Marching Cubes [8] and several polygonization methods have appeared in the literature as, for example, Dual Contouring [9], Dual Marching Cubes [10], among others [11]. These methods may not generate appropriate results for degenerate solids. The polygonization of such models will either result in holes or non-manifold components as shown in figure 2.

Topological problems from degenerate voxel art objects can be removed at voxel level or polygonal mesh level. The work in [12] subdivides critical volumetric cells into 125

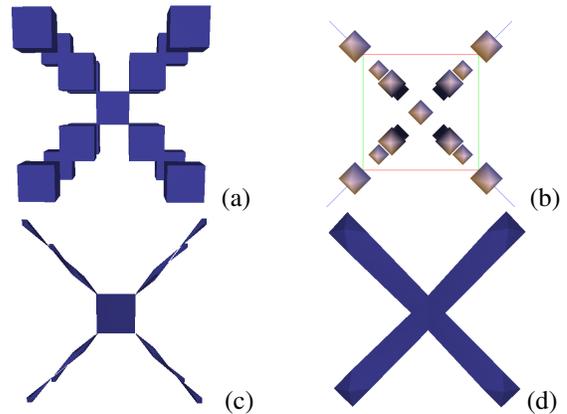


Fig. 2. Example of polygonization of a voxel art: (a) is the original model where each voxel is represented as a cube. (b) is how [6] polygonizes it, which generates gaps between voxels. (c) situation where the polygonization generates non-manifold components. (d) is the desirable result as a manifold polygonal model with no gaps between voxels.

smaller cells fixing degenerate connections between the input voxels, however making the mesh extraction process much more expensive in terms of memory and processor usage. On the other hand, [13] has an extensive methodology to repair a myriad of problems on polygonal meshes, including the presence of non-manifold edges and vertices. Both methods work by replacing non-manifold edges with cylinders and the non-manifold vertices with spheres. The geometry generated by this kind of approach is not coherent with the connectivity between the elements from a digital voxel art. Differently, the method we propose works at voxel level by re-sampling the discrete scalar field that describes the volume of the object, subdividing critical elements in 8 parts and it generates a geometry that is coherent with the topology of the original volumetric model.

The work in [2] solves a problem in 2D that resembles the problem that is being investigated in this work. Their work converts digital images used in video-games from the 80's and early 90's into a set of closed curves with no intersection, allowing them to rescale these images into any resolution. Their method eliminates the topological problems that were derived from the aliasing that exists in digital images and it respects the connectivity between the original elements of the image. Voxel art stores a description of the surface, distinguishing the object from the background, unlike digital images derived from old video games. The author of that work circumvents this limitation by treating different colors as multiples materials, thus, increasing the complexity of the solution. The similarity graph of their method has inspired the connectivity graph of our method, which is an important concept to understand how it resamples the scalar field that describes the volume of the object, before its triangularization. It also guarantees the reconstruction of a 3-manifold whose boundary will yield a two manifold represented by the desired mesh to be extracted.

III. THEORETICAL FRAMEWORK

Here we define some concepts of combinatorial topology [14] and the concept of connectivity graphs which are the fundamentals of our proposal.

A. Basic Combinatorial Topology Concepts

Let an *affine cell* immersed in \mathbb{R}_n be the convex hull of a finite set of points and define a cell of dimension p in \mathbb{R}_n , or simply p -cell, as the homeomorphic image of an affine cell in \mathbb{R}_n . We define a cell γ generated from a subset of points of a cell σ as a *subcell* of γ . We also define a *cell complex* Σ as a finite set of cells where the intersection of two cells $\sigma, \gamma \in \Sigma$ also belongs to Σ . Moreover, we call the complex *d-regular* if every p -cell with $p < d$ is a subcell of at least one cell of dimension p . Two cells σ and γ are *connected* if there is a sequence of cells $(\gamma_i)_{i=0}^n$ connecting them. A subcomplex Σ^* of a cell complex Σ is a *connected component* of Σ if all its cells are connected.

A cell complex Σ is a *combinatorial manifold of dimension d* if it satisfies the following properties:

- 1) Σ is a connected d -regular complex.
- 2) Every cell with dimension $(d-1)$ of Σ must be subcell to a maximum of two cells with dimension d .
- 3) The neighborhood of a vertex in Σ is homeomorphic to \mathbb{R}^d or \mathbb{R}_+^d .

A combinatorial manifold is orientable if it is possible to choose a coherent orientation for all of its cells, in other words, two adjacent k -cells always induce opposite orientations at the $(k-1)$ -cells that are shared by them.

B. Connectivity Graph

Let $V \subset \mathbb{R}^3$ be a volumetric region, described by a uniform cell decomposition, and Ω a combinatorial 3-manifold associated to V . The geometric support of the non-degenerate digital voxel art models is represented by regular complexes $\Sigma \subset \Omega$, while the geometric support from degenerate volumetric models is $\Sigma' \subset \Omega$. If a cell σ is such that $\sigma \in \Sigma$ or $\sigma \in \Sigma'$, then $\sigma \notin \partial\Omega$.

Let $\Sigma \subset \Omega$ be a non-degenerate solid. We define the *dual connectivity graph* associated to Σ , denoted by $DCG = (VE, ED)$, where VE is a set of vertices and ED is the set of edges of the graph. The sets VE and ED are defined by two bijections: $\eta_c : \sigma \rightarrow v$, where $\sigma \in \Sigma$ and $v \in VE$; $\eta_f : \tau \rightarrow e$ that for each 2-subcell $\tau \in \Sigma$, adjacent to two 3-cell σ and σ' , associates an edge $e = (u, v) \in ED$, $u \in VE$ and $v \in VE$, where $\eta_c(\sigma) = u$ and $\eta_c(\sigma') = v$.

Let $\Sigma' \subset \Omega$ be a degenerate solid. In this work, we will denote a *critical 1-face* or *critical edge* ϵ , every edge that is incident to two cells σ and σ' , $\sigma \cap \sigma' = \epsilon$. Analogously, a *critical 0-face* or *critical vertex* v is a vertex that is incident to two cells σ and σ' , $\sigma \cap \sigma' = v$. Observe that the definitions of critical vertices and edges are particular for the combinatorial structure of Ω .

We define the *extended dual connectivity graph* associated to Σ' denoted by $EDCG = (VE', ED')$ where VE' is the set of vertices and ED' is the set of edges of the graph. The set

VE' is defined by the bijection $\eta_c : \sigma \rightarrow v$, where $\sigma \in \Sigma'$ and $v \in VE'$. The set of edges is defined by three different types of bijections that depend on the neighborhood system between the cells of Σ' :

- 1) For each 2-subcell $\tau \in \Sigma'$, adjacent to two cells σ and σ' , $\eta_{fe} : \tau \rightarrow e$ associates an edge $e = (u, v) \in ED'$, where $\eta_c(\sigma) = u$ and $\eta_c(\sigma') = v$;
- 2) For each critical edge $\epsilon \in \Sigma'$, $\eta_{ee} : \epsilon \rightarrow e$ associates an edge $e = (u, v) \in ED'$, where $\eta_c(\sigma) = u$, $\eta_c(\sigma') = v$, $\sigma, \sigma' \in \Sigma'$;
- 3) For each critical vertex $v \in \Sigma', v \notin \epsilon$, where ϵ is a critical edge, $\eta_{ve} : v \rightarrow e$, associates an edge $e = (u, v) \in ED'$, where $\eta_c(\sigma) = u$, $\eta_c(\sigma') = v$, $\sigma, \sigma' \in \Sigma'$.

Case 3 affirms that we only create an edge in the *EDCG* corresponding to two cells incident to a critical vertex when such vertex is not already a member of a critical edge. This avoids by construction that crossed edges arise in the considered graph.

Figure 3 illustrates an example of *EDCG* being constructed from a degenerate regular complex Σ' .

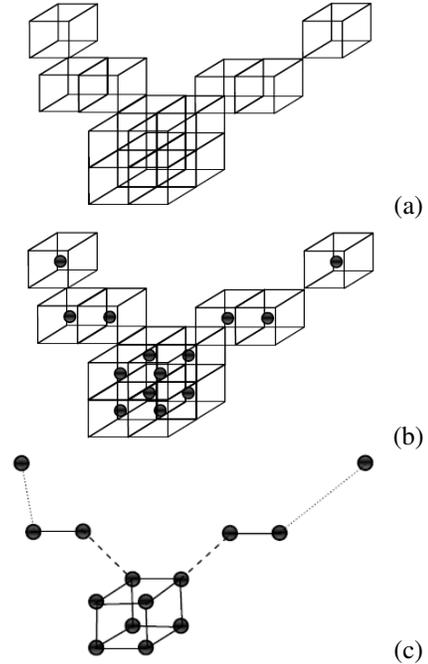


Fig. 3. (a) Example of a regular complex Σ' that contains critical edges and vertices. (b) The centroid of each cell induces a vertex in the graph *EDCG*. (c) The graph *EDCG* is generated from Σ' ; the filled edges correspond to the bijection $\eta_{fe} : \tau \rightarrow e$, the dashed edges correspond to $\eta_{ee} : \epsilon \rightarrow e$ and the ones with a thin dotted line correspond to $\eta_{ve} : v \rightarrow e$.

In this work, the cells from the regular complex Ω will receive four distinct classifications. First, define $\bar{\Omega} = \Omega - \Sigma$. A cell $\sigma \in \Sigma$ is *interior* if every 2-subcell $\tau \subset \sigma$ is adjacent to exactly two distinct cells of Σ . A cell $\omega \in \bar{\Omega}$ is *exterior* if $\omega \cap \sigma = \emptyset$ or $\omega \cap \sigma = \tau$, $\tau \subset \sigma$ is 0,1,2-subcell, for every cell

$\sigma \in \Sigma$ and ω does not contain any critical vertex or edge. A cell $\sigma \in \Sigma$ is a *boundary cell* if $\exists \omega \in \bar{\Omega}, \sigma \cap \omega \neq \emptyset$ and $\sigma \cap \omega$ contains one or more 2-cells. Finally, a cell $\omega \in \bar{\Omega}$ is called a *refinement cell* if $\exists \sigma_1, \sigma_2 \in \Sigma, \sigma_1 \cap \omega \neq \emptyset, \sigma_2 \cap \omega \neq \emptyset$ and $\sigma_1 \cap \sigma_2$ is a 1-subcell or 0-subcell, i.e., ω is incident to a critical vertex or edge. The same definitions are valid if we consider a degenerate solid represented by the regular complex Σ' . This classification will be used later to select the regions that will be part of the boundary of the object that will be generated by the method proposed at this document.

From now on, in this text, the term *voxel* will be used to reference three dimensional cells of the cell complex.

IV. MANIFOLD MESH EXTRACTION FROM VOXELS

The aim of this work is to extract meshes that represent the boundary of volumetric data built by artists described by uniform spatial decomposition. Such data are the analogous in 3D to the pixel art in 2D. The extraction of meshes from volumetric data is a problem that has been thoroughly investigated in the literature. On the other hand, the problem we tackle here involves volumetric data which do not correspond in the majority of the cases to 3D-combinatorial manifolds. In other words, such data correspond to degenerate solids. Extracting meshes representing boundary surfaces of such solids while preserving the original expression of the artist is a non trivial problem with challenges that have led to the development of this work.

A. Problem definition

Let V' be a volumetric model represented by a cell complex $\Sigma' \in \Omega$. Extract a 2D combinatorial manifold mesh representing the boundary $\partial \bar{\Sigma}$ of a 3-manifold $\bar{\Sigma}$ reconstruction of V' . The 3-manifold reconstruction of V' does not need to preserve the original combinatorial structure of V' .

B. The proposed method

The solution for the mesh extraction problem we tackle here is based on a constructive approach that relies on concepts of combinatorial manifolds and discrete data resampling. The method proposed subsumes two macro steps: the first step consists in obtaining a 3-manifold reconstruction $\bar{\Sigma}$ from the original volumetric data V' , not necessarily with the same combinatorial structure; the second step is the one that finally extracts a combinatorial 2-manifold $\partial \bar{\Sigma}$ which is the boundary of the reconstructed volume.

The 3-manifold reconstruction of V' is obtained by constructing the primal connectivity graph from the extended dual connectivity graph of the regular complex Σ' associated to V' . The primal connectivity graph guarantees by construction that, in the new cellular complex, each face is adjacent to exactly two cells if it is interior to the volume or to one cell if it is part of the boundary. Hence, such cellular complex is bound to be a 3-manifold. The final mesh is then extracted from the boundary of the 3-manifold reconstructed cellular complex. The extracted mesh is guaranteed to produce a manifold as a consequence of the result of combinatorial topology that states

that the boundary of a 3-manifold is a 2-manifold without boundaries.

In the proposed method, the connectivity graph is not constructed explicitly as it will be shown in more details in the sequel. We use a resampling process followed by the assembly of affine cells, which are locally defined by the convex hull of the new computed samples. The resampling process enables the correction of the topology in the degenerate neighborhoods of the solid so that in final stage each region will be locally homeomorphic to the sphere. The resampling process is defined in such a way that the topology is modified and corrected only near the critical vertices and edges preserving the original topology of the non-degenerate regions.

The overall method can be organized and detailed in three steps as shown in the fluxogram in figure 4.

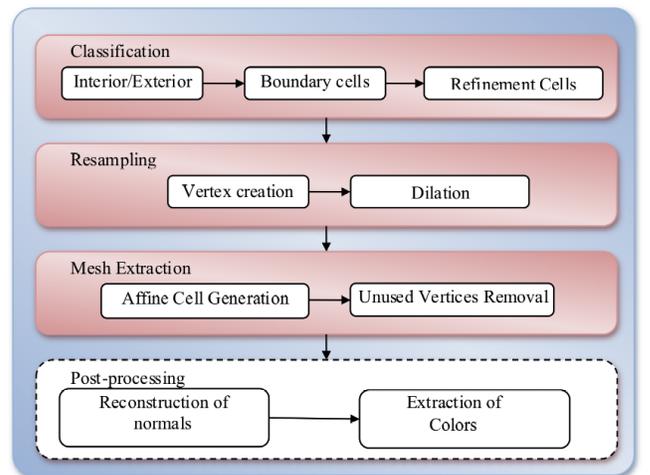


Fig. 4. Fluxogram of the proposed method.

The first step consists in classifying the cells in the cell complex Ω embedded in the Euclidean R^3 space in *interior*, *exterior*, *boundary* and *refinement* cells, considering the volumetric model represented by Σ' . A set of polygons will be produced for the cells classified as boundary and for the refinement cells which are in the frontier between the interior and exterior of the considered volumetric object. Conversely, no polygons will be associated to cells classified as interior or exterior. There are two reasons for considering the refinement cells which are located out of the object in the polygonization substep: the first one is to locally modify the topology of Σ' in order to enable the extraction of a 2-manifold which will bring forth the desired mesh; the second reason is to enhance the geometric quality of the polygonized region. Here we built a intermediary complex $\bar{\Sigma}$ from which $\partial \bar{\Sigma}$ will be extracted.

The second step aims at the resampling of the scalar field given by the characteristic function of the volume defined on the cellular complex Σ' associated to V' . The resampling is the step that yields a new set of vertices. A new intermediary regular complex $\bar{\Sigma}$ will be constructed from the new set of vertices obtained. This step consists of two substeps:

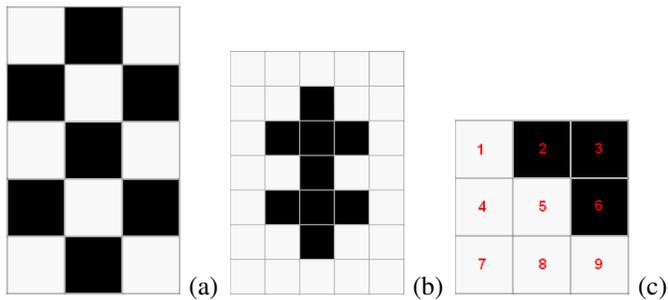


Fig. 5. (a) Example of data provided by the user with black cells being the ones the user has painted, while the white ones are not painted. (b) Classification of the cells in interior (black) and exterior cells (white). (c) Zoom in at the lower left part of second image. Cell 5 is a refinement cell because the boundary cells 2 and 6 are neighbors of cell 5; cells 2 and 6 have no edges (2-subcells) in common but they are connected by a vertex (1-subcell) that is incident to the cell 5. The other cells are not refinement cells.

- 1) A vertex is generated for each centroid of the boundary cells from Σ' . Vertices are also generated for each centroid at all d -faces, $d = 0, \dots, 2$, incident to two boundary cells.
- 2) A dilation operator is applied at all new vertices created in order to yield additional six new vertices for each new vertex created.

1) *Voxel classification:* Voxel classification starts with the detection of interior and exterior voxels. We must be aware that the user may leave interior voxels unpainted (hollow regions) so it is necessary a pre-processing step to correct this problem. This can be simply solved by using a region growing based method.

The next classification step detects border voxels and voxels in the refinement regions which are exactly what we defined as refinement cells (section III.B). An example of refinement voxel can be seen in figure 5.

2) *Volume resampling:* After all voxels are classified, those who were classified as border voxels or refinement voxels are subdivided in eight (8) new voxels, as shown in the 2D example in figure 6a. Then, each new voxel, after the subdivision is considered as a cubic region containing 8 vertices which are initially labeled as internal or external according to the following criterion:

- Internal: new elements created in the center of the subdivision of a original border cell and in the central positions in the adjacent elements(vertices, edges or faces) between two border cells
- External: all other created elements

In the application of the dilation operator, vertices at an unitary distance from each original vertex initially labeled as internal are relabeled as internal, independently of their original labeling. This results in a propagation of internal labels in the four main directions (six directions in the 3D case). This can be seen in figure 6b. This is the step that defines the resampling operation that corrects the topology of the critical vertices and edges.

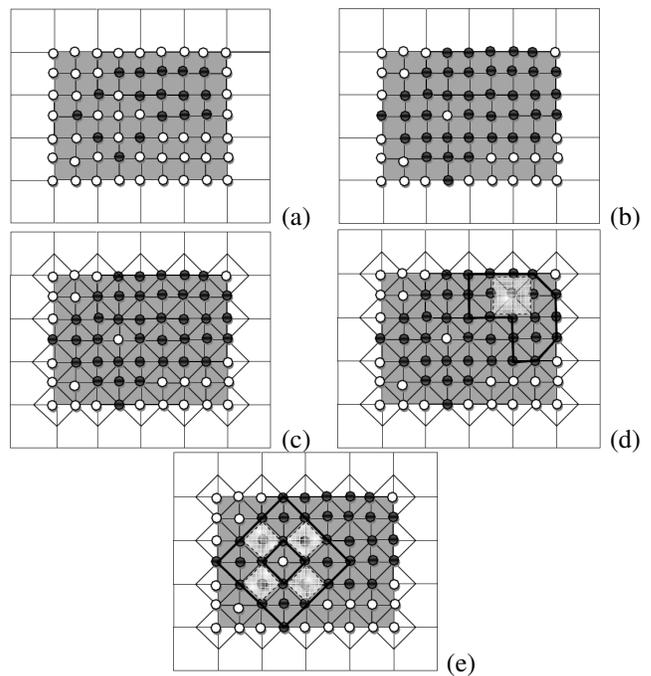


Fig. 6. Example of a slice of a scalar field where dark cells are internal/boundary and white cells are external/refinement. (a) is the scalar field before resampling. (b) is the scalar field after the dilation operation. (c) Dual lattice rotated in 45 degrees. The center of the cells of this lattice are the vertices created before the dilation operation. The remaining vertices intersects with multiples cells from the reticulate. (d) illustrates how a cell is generated in regions with no critical vertices. (e) illustrates how a cell is generated in regions with critical vertices.

It is possible to present a geometric and topological argument that explains how this resampling process removes all critical vertices and edges without changing the combinatorial topology of the topologically correct regions of the cell complex. This argument considers a 45 degree rotated dual version of the original grid and shows that the dilation operation at critical vertices and edges locally produce triangulated 2-manifolds.

We pose the conjecture that a similar argument can be presented for the tridimensional case, considering a dual grid formed by octaedric cells subdivided in tetrahedra. Similarly, one tetrahedra will be considered internal if its vertices are all labeled as internal and otherwise it will be considered external.

3) *Mesh extraction:* The mesh extraction is done in two stages: first, we generate a set of affine cells (polyhedra) that define a new cell complex associated to the intermediary volume $\bar{\Sigma}$; second, we determine among all affine cells created, those that contain faces that are on the surface that bounds the volume represented by a mesh associated to the boundary of $\partial\bar{\Sigma}$.

Affine cell generation - Each new voxel produced in the subdivision step is considered as a volumetric region that will generate an affine cell, that is, a polyhedron (see figure 7). Each cubic region associated to a given voxel will be associated to a 8 bit binary number, where each bit is equal to

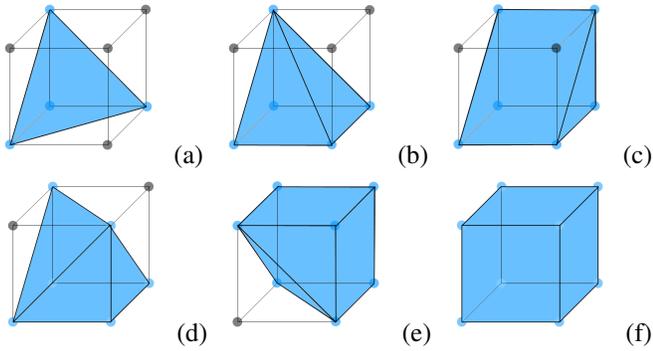


Fig. 7. The six cases of the Lookup Table of our polygonization step. Blue vertices are inside the volume of the mesh, while black ones are in the exterior. (a) Tetrahedron, with 10 rotations. (b) Squared pyramid, with 24 rotations. (c) Triangular prism, with 12 rotations. (d) Union of two squared prisms, with 12 rotations. (e) The geometric dual of a tetrahedron inside a cubic region, with 8 rotations. (f) Cube (case 255 of the Lookup Table) used only to remove internal faces.

1 if the associated vertex is interior to the volume or zero in the case the vertex is external. The binary numbers are used to query a Lookup Table which defines the final configuration of the affine cell inside the cubic volumetric region.

The Lookup Table, shown in figure 7 presents only 67 valid configurations which can be seen as rotations of 6 fundamental different cases. It must be made clear that the lookup table used here is not a sub-case of the Marching Cubes lookup table containing a completely different set of configurations. All fundamental cases, shown in figure 7 are: tetrahedron, squared pyramid, triangular prism, union of two squared prisms, the geometric dual of a tetrahedron inside a cubic region and the cube (for more details see [15]).

Mesh construction - The set of all faces belonging to the generated affine cells determines a set of candidate faces to define the final mesh that bounds the desired volume.

There are two conditions that establish when a candidate face will be part of the resulting polyhedral surface;

- The candidate face must not be part of a face that bounds a internal cubic region of the volume;
- Each face in the mesh must be created only once. If the candidate face already exists, then it must be removed, as it is part of the interior of the volume

The attributes of a face of the mesh must be computed as soon as it is generated. The normal vector of the face is trivially calculated by the cross product of the vectors associated to two edges of the face in the counterclockwise orientation. The color of each face is defined by a weighted mean of the color of the voxels that are adjacent to the region that contains the considered face. We associate different weights that depend on the type of the voxel adjacency to the voxel that contains the considered face. Here we used the following weights: 8 for the color of the voxel that contains the considered face, 4 to the voxels that share a face with the voxel in which the face is embedded and 1 for voxels that share on vertex with the considered region.

After the generation of all faces of the mesh, we remove all vertices that are not incident to any faces that were not removed from the generated object.

V. RESULTS

This section presents the results obtained in each step of the proposed method to validate it. For this reason, strategic steps of the process will be analyzed separately, describing some of its limitations and how to avoid them.

A. Voxel Classification

The solution adopted to detect the internal regions of the volumetric model may not work correctly when the artist creates an object where the boundary is a surface without border. In this case, the model may have tunnels and there will be mesh generated inside these tunnels regardless if they will be visible or not, which may be an undesirable result in several situations. The recognition of internal regions in this case is an unsolved problem [16] and it is beyond the scope of this work. These cases may be avoided if the artist closes the holes using a transparent color or black.

The proposed method do work correctly with models that do not have genus zero as seen in figure 8.

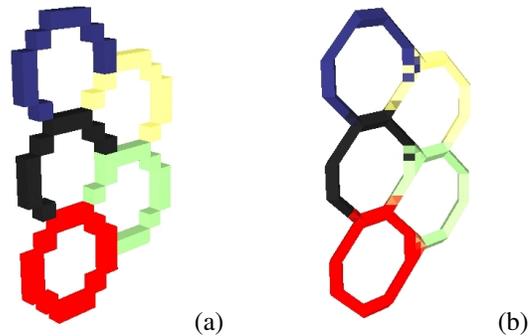


Fig. 8. Model that does not have genus zero. (a) Original volumetric data. Each voxel is a cube. (b) Mesh generated from it, with colors and normals per face.

B. Volume Resampling

Here we present some results of the resampling process and its effect at the mesh extracted by our method and other methods of the literature.

Figure 9 illustrates a volumetric data that has critical vertices while figure 10 shows a volumetric model that has a critical edge. In both cases, it shows the model resampled by our technique and polygonized using our technique and the method from [6].

Our resampling method guarantees that most of the ambiguous cases from Marching Cubes are eliminated, except one that is treated by our Lookup Table as the union of two prisms, which shows up at refinement regions near critical edges. This is the reason why Lewiner's implementation of Marching Cubes did not generate a mesh with a geometry coherent with the one from the volumetric model, while it

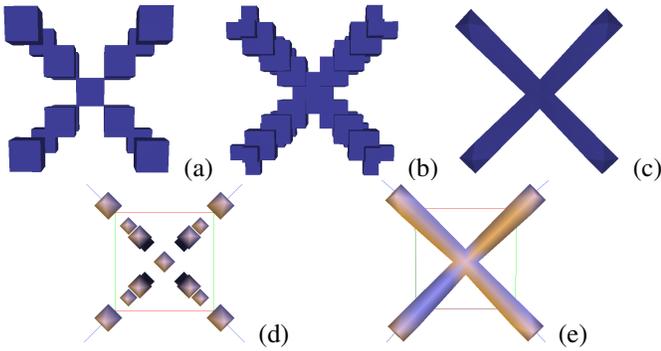


Fig. 9. Example of volumetric data with critical vertices. (a) Original volumetric data. (b) Volumetric data resampled by our technique. (c) Result of our mesh extraction process with the original volumetric data. (d) Result of [6]’s mesh extraction process with the original volumetric data. (e) Result of [6]’s mesh extraction process with the volumetric data resampled by our method.

works correctly with resampled objects that have no critical edges. On the other hand, the results obtained by our mesh extraction method are appropriate on both topological and geometrical points of view.

C. Mesh Extraction

This section analyzes the performance of the proposed technique and the quality of the meshes extracted by our method. The test environment consisted of an AMD Phenon x3 (3 cores) 8750 (2.4ghz) and 8gb RAM, although the code uses only one core and it did not use the GPU. Figure 11 illustrates three of the high resolution models that were tested with our method while table I shows the dimension and counts the amount of user painted (solid) voxels and the total of voxels in the input data; it displays the amount of vertices and triangles of the output data; it displays the time elapsed to extract the mesh and it uses some metrics to analyze the quality of the mesh.

TABLE I
Statistics of the models illustrated at figure 11.

Model Name	Mad Dog	Kirov MKII	War Miner
Size	100x97x71	42x67x140	29x26x65
Solid Voxels	24 570	29 558	6 409
Total Voxels	688 700	393 960	49 010
Vertices	93 983	75 838	28 004
Triangles	205 676	169 430	64 858
Aspect Ratio	1.478	1.492	1.482
Skewness	0.222	0.229	0.234
Smoothness	0.007	0.007	0.007
Execution Time	3 464 ms	2 356 ms	403 ms

The meshes extracted by this technique are more dense and uniform than the ones from Marching Cubes, due to the resampling process. However, they are well behaved according to the metrics considered in this work. The first metric, the *aspect ratio*, is the ratio between the longest and shortest edges of each triangle of the mesh. Values near 1 happens on meshes where equilateral triangles are predominant. *Skewness*

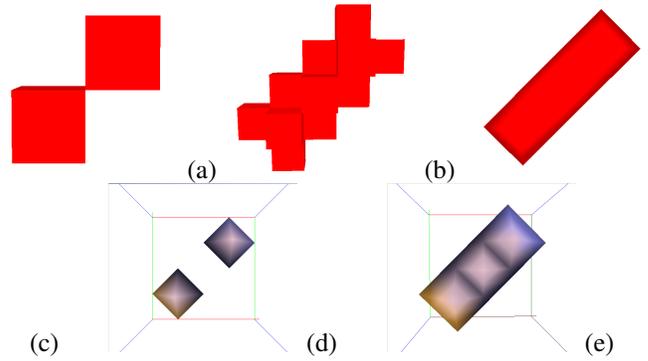


Fig. 10. Example of volumetric data with critical edges. (a) Original volumetric data. (b) Volumetric data resampled by our technique. (c) Result of our mesh extraction process with the original volumetric data. (d) Result of [6]’s mesh extraction process with the original volumetric data. (e) Result of [6]’s mesh extraction process with the volumetric data resampled by our method.

measures asymmetry of the distribution of the triangles in the mesh. Let A_T be the area of a triangle of the mesh, while A_{CT} is the area of an equilateral triangle T inscribed in a circle that circumscribes T . *Skewness* is the ratio between the $A_{CT} - A_T$ and A_{CT} . Values near zero correspond to more symmetric meshes. The third metric is *smoothness* which measures the difference between the area of a triangle and its immediate neighbors. Smoother meshes have values near 0.

In our technique, the triangles are generated with vertices at the border of cubic cells. This ensures that each triangle will either have an aspect ratio of 1, $\sqrt{2}$ or $\sqrt{3}$ and the angles between every edge are multiples of 45° . Most of the times that will make the aspect ratio approach $\sqrt{2}$, while skewness and smoothness will approach zero due to the similarity between the triangles of the mesh.

VI. CONCLUSION

The objective of this work consists of extracting polygonal meshes from volumetric models generating orientable manifolds that are consistent with the connectivity between the original voxels. The solution for this problem must take into account the challenges of dealing with degenerate or low resolution volumetric objects.

In order to do this, our technique classifies the voxels to detect which regions will receive triangles. These regions are refined, detecting the vertices that will be used to generate volumes that will represent combinatorial 3-manifolds, without discontinuities at the adjacencies of the original voxels.

The voxel classification step works correctly as long as the boundary of the volumetric model is defined by a surface without borders. The solution of this problem in the general case is beyond the scope of this work, being up to the artist to fill the empty spaces in the surface of the object.

The volume resampling step generates a discrete scalar field that removes most configurations that causes ambiguities in the original Marching Cubes, except for the one specified in the case (d) of our Lookup Table at figure 7 in refinement cells

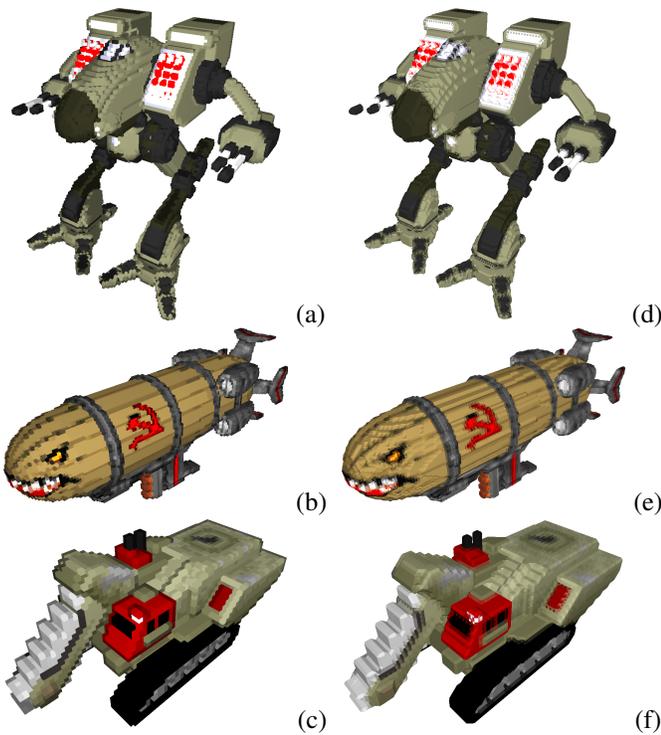


Fig. 11. Examples of high resolution volumetric data before and after the mesh extraction. In order to notice the differences between them, use a 1200% zoom in at each image. (a) Original volumetric data for Mad Dog [17]. (b) Original volumetric data for Kirov MKII [18]. (c) Original volumetric data for War Miner [19]. (d) Polygonal mesh obtained for Mad Dog. (e) Polygonal mesh obtained for Kirov MKII. (f) Polygonal mesh obtained for War Miner.

near critical edges. However, the resampled data is enough for our mesh extraction step to obtain orientable combinatorial manifolds for any input data.

During the construction of the mesh, the method successively creates a set of affine cells defined by the sign of the subdivided voxels and the configurations stored in a lookup table that specifies all kinds of affine cells necessary to assemble the desired combinatorial cell complex. Each new affine cell is merged with the previously created affine cells. Finally, the boundary of the 3-manifold cell complex is extracted by removing internal and duplicated faces yielding the final mesh.

The adjacency relationship between the original voxels are preserved, independently of the genus of the input data. The method is executed in interactive time and it generates a dense mesh which also has a good quality considering the metrics used to analyze it, due to the predictable size and angles of the edges, discarding the need of a remeshing process.

The uniform distribution of the vertices allows them to be mapped onto a 3D lattice that allows us to use a mix of implicit data and boundary representation. We shall use this information in our future works to obtain information from models, optimize the mesh, etc.

ACKNOWLEDGMENT

The authors are grateful to Faperj and CAPES for the funds granted to support this work.

The authors are also grateful to the users of Project Perfect Mod for helping with tests, providing some of the models used in this work and supporting this research.

REFERENCES

- [1] Return of the Dawn, *Volumetric model created by the user Reaperr for the modification Return of the Dawn for the game Command & Conquer: Tiberian Sun*. <http://www.ppsite.com/forum/index.php?f=116>. Retrieved April 2013.
- [2] J. Kopf, and D. Lischinski, *Depixelizing Pixel Art*. ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2011, Jul., 2011, vol. 30, No. 4, ACM SIGGRAPH, Article No. 99.
- [3] Voxel - Wikipedia, *List of games that uses voxel data*. <http://en.wikipedia.org/wiki/Voxel>. Retrieved July 2013.
- [4] 3D Dot Game Heroes, *Commercial game for Playstation 3 developed by Silicon Studio*. <http://www.3ddotgameheroes.com/>. Retrieved April 2013.
- [5] W. E. Lorensen and H. E. Cline. *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*. Computer Graphics, Siggraph '87 Conference Proceedings, Jul. 27-31, 1987, vol. 21, No. 4, ACM Siggraph, pp. 163-169.
- [6] T. Lewiner, H. Lopes, A. W. Vieira and G. Tavares. *Efficient implementation of marching cubes cases with topological guarantees*. Journal of Graphics Tools, A.K.Peters, Dec. 2003; vol. 8, No. 2, pp 1-15.
- [7] E. V. Chernyaev. *Marching Cubes 33: Construction of Topologically Correct Isosurfaces*. Technical Report CERN CN 95-17, CERN, 1995.
- [8] L. P. Kobbelt, M. Botsch, U. Schwaner and H. P. Seidel. *Feature sensitive surface extraction from volume data*. Computer Graphics; Siggraph 2001 Conference Proceedings, Ago. 2001, ACM Siggraph; pp. 57-66.
- [9] T. Ju, F. Losasso, S. Schaefer and J. Warren. *Dual contouring of hermite data*. ACM Transactions on Graphics, 2002, vol. 21, No. 3, pp. 339-346.
- [10] S. Schaefer and J. Warren. *Dual Marching Cubes: Primal Contouring of Dual Grids*. Proceeding PG '04 Proceedings of the Computer Graphics and Applications, 12th Pacific Conference; IEEE Computer Society Washington, DC, USA 2004; pp 70-76.
- [11] T. Newman and H. Yi. *A survey of the marching cubes algorithm*. Computers & Graphics, Vol. 30, No, Elsevier, Oct 2006, pp. 854-879.
- [12] S. Bischoff and L. Kobbelt. *Extracting Consistent and Manifold Interfaces from Multi-valued Volume Data Sets*. Bildverarbeitung für die Medizin 2006, Springer Berlin Heidelberg, pp. 281-285.
- [13] S. Bischoff, D. Pavic, L. Kobbelt. *Automatic Restoration of Polygon Models*. ACM Transactions on Graphics, Oct. 2005, vol. 24, No. 4, pp. 1332-1352.
- [14] M. Henle. *A combinatorial introduction to topology*. WH Freeman San Francisco 1979.
- [15] Carlos E. V. Muniz. *Extração de Malhas Poligonais a Partir de Modelos Volumétricos Criados por Artistas*. Dissertação de Mestrado. Universidade Federal Fluminense - 2012.
- [16] J. Digne, D. Cohen-Steiner, P. Alliez, M. Desbrun and F. de Goes. *Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets*. INRIA Research Report 7991, Jun 2012.
- [17] Yuri's Revenge Argentina, *Mad Dog Mech model created by the user WeeRaby2k*. <http://yrarg.cncguild.net/old/index.php?page=voxels/vxl9>. Retrieved April 2013.
- [18] Project Perfect Mod Forums, *Kirov Airship Mark II model created by the user Arikado*. <http://www.ppsite.com/forum/viewtopic.php?t=18995>. Retrieved April 2013.
- [19] Project Perfect Mod Forums, *WarMiner model created by the user Gangster*. <http://www.ppsite.com/forum/viewtopic.php?t=7158>. Retrieved April 2013.