



UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO
CIÊNCIA DA COMPUTAÇÃO

CARLOS EDUARDO VAISMAN MUNIZ

EXTRAÇÃO DE MALHAS POLIGONAIS A PARTIR DE MODELOS VOLUMÉTRICOS
CRIADOS POR ARTISTAS

NITERÓI - RJ

2012

CARLOS EDUARDO VAISMAN MUNIZ

EXTRAÇÃO DE MALHAS POLIGONAIS A PARTIR DE MODELOS VOLUMÉTRICOS
CRIADOS POR ARTISTAS

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Mestre em Computação. Área de Concentração: Computação Gráfica.

ORIENTADOR: PROF. DSc. ANSELMO ANTUNES MONTENEGRO

Niterói - RJ
2012

CARLOS EDUARDO VAISMAN MUNIZ

EXTRAÇÃO DE MALHAS POLIGONAIS A PARTIR DE MODELOS VOLUMÉTRICOS
CRIADOS POR ARTISTAS

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do grau de Mestre. Área de Concentração: Computação Gráfica

Aprovada em setembro de 2012

BANCA EXAMINADORA

Prof. DSc. Anselmo Antunes Montenegro - Orientador
UFF

Prof. DSc. Marcos de Oliveira Lage Ferreira
UFF

Prof. DSc. Leandro Augusto Frata Fernandes
UFF

Prof. DSc. Waldemar Celes
PUC-RIO

Niterói - RJ
2012

Ao meu pai, pelo apoio, incentivo e carinho por todo meu percurso no mestrado.

À minha irmã que revisou o texto várias vezes, mostrou várias formas de melhorar a forma de redigir o texto e sempre incentivou e deu muito apoio moral durante a redação desta dissertação.

À minha querida mãe, que apesar de não poder estar comigo hoje, me deu uma grande lição de vida, muito carinho, sempre me apoiou mesmo nos momentos mais difíceis e me ensinou a correr atrás dos meus sonhos.

À minha querida namorada Adriana, por todo seu carinho e apoio moral.

AGRADECIMENTOS

À Universidade Federal Fluminense,

Ao Prof. DSc Anselmo Antunes Montenegro por me orientar neste trabalho e por todo o apoio,

Ao Prof. DSc Marcos de Oliveira Lage Ferreira pela coorientação e os slides que auxiliaram muito na escrita do texto dessa dissertação,

À CAPES pelo apoio financeiro durante todo o mestrado,

A todos os professores com os quais tive o privilégio de ser aluno e todos os colegas de faculdade,

A todos que programaram e contribuíram com o Voxel Section Editor III (VXLSE III),

A todos os visitantes do Project Perfect Mod e usuários do VXLSE III que me apoiaram e motivaram a fazer este trabalho,

À Electronic Arts que permitiu o uso de formatos proprietários da série de jogos Command & Conquer e o uso de seus modelos.

RESUMO

Este trabalho apresenta um método para extração de superfícies poligonais a partir de modelos volumétricos criados por artistas, propondo um modo de utilizar ferramentas de modelagem volumétrica para construção de modelos representados por bordo.

Dados criados em editores de voxels, ou adquiridos através de processos volumétricos para aquisição de geometria 3D, podem conter características topológicas que não condizem com a estrutura de um sólido, isto é, um objeto delimitado por uma fronteira que separa de forma bem definida seu interior de seu exterior. Resolver o problema de extração de superfícies que representam dados com essas características é o principal objetivo deste trabalho.

Para o sucesso da extração de superfícies, foi elaborada uma metodologia para reamostrar qualquer modelo representado de forma volumétrica, de tal forma que seja possível reconstruir, a partir da reamostragem, uma variedade combinatória de dimensão 3, que pode ser poligonizada sem a geração de superfícies com buracos ou problemas topológicos.

A malha criada por esta técnica é suave e seus triângulos têm boas propriedades, satisfazendo alguns dos principais critérios utilizados para medir a qualidade de malhas, como razão de aspecto (*aspect ratio*), suavidade na variação das áreas dos triângulos (*smoothness*) e simetria (*skewness*).

Apesar de ter triângulos com boas características, o modelo obtido a partir do processo de extração pode apresentar um aspecto mais facetado do que o esperado. Isto levou a necessidade de aplicar um método de suavização para obtenção de uma superfície poligonal com geometria mais suave, sem, entretanto, perder detalhes essenciais do modelo original. O método de suavização proposto e utilizado neste trabalho garante que a superfície produzida continue sendo uma 2-variedade, não introduzindo degeneração na estrutura topológica final.

Palavras-chave: Poligonização de dados volumétricos, suavização de malhas, Marching Cubes, modelos volumétricos.

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 or obtained from processes that acquire 3D geometry from volumes may contain topological features does not describe solid structures, because they do not have a frontier that separates the interior and the exterior of the volume in a clear and well defined way. The main objective of this work is to solve the problem of extracting triangle meshes from volumes that contains these topological features.

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 surface with gaps or topological problems.

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

Despite having triangles with good properties, the models extracted by this method may present more faceted features than what was expected. In order to avoid this problem, it is necessary to apply a method to obtain a surface with a smoother geometry that preserves crucial details of the original model. The mesh smooth method proposed and used at this work guarantees that the produced surface is still a 2-manifold by not introducing degenerations at the topological structure of the generated object.

Key words: Polygonization of volumetric models, mesh fairing, Marching Cubes, polygon reconstruction, volumetric models.

SUMÁRIO

RESUMO	6
ABSTRACT	7
SUMÁRIO	1
LISTA DE FIGURAS	3
LISTA DE TABELAS	7
LISTA DE ABREVIATURAS E SIGLAS	8
Capítulo 1 – Introdução	9
Capítulo 2 – Trabalhos relacionados	14
2.1 – Métodos para extração de superfícies a partir de dados volumétricos	14
2.2 – Restauração de vértices e arestas não variedades	19
2.3 – <i>Depixelizing Pixel Art</i>	21
2.4 – Conversão de voxels em polígonos através de filtragem de sinais	23
Capítulo 3 – Fundamentos	26
3.1 – Objetos Volumétricos	26
3.2 – Descrição geométrica de objetos volumétricos	27
3.2.1 – Descrição por bordo	27
3.2.2 – Descrição implícita	28
3.3 - Representação discreta de objetos volumétricos	28
3.3.1 - Representação por decomposição espacial	28
3.3.2 - Representação por bordo	30
3.4 – Noções de topologia combinatória	30
3.4.1 – Complexos celulares e complexos simpliciais	31
3.4.2 – Variedades combinatórias	32
3.5 – Grafo de conectividade	33
3.6 – Poligonização de superfícies implícitas	36
Capítulo 4 – O método proposto	38
4.1 – O problema de extração do bordo de modelos volumétricos não variedade	38
4.1.1 – Definição do problema	39
4.2 - O Método proposto	39
4.2.1 – Classificação dos voxels	43
4.2.2 – Reamostragem da região volumétrica e classificação de vértices	46
4.2.3 – Extração da malha	50
4.3 – Resultados	52
4.3.1 – Classificação dos voxels	52
4.3.2 – Reamostragem da região volumétrica	54
4.3.3 – Poligonização	57
Capítulo 5 – Pós-Processamento	60
5.1 – Suavização de atributos	61
5.2 – Suavização da malha	62
5.3 – Análise de resultados	66
5.3.1 – Suavização de atributos	67
5.3.2 – Suavização da malha	67
Capítulo 6 – Resultados	70
6.1 – Ambiente de experimentação	70
6.2 – Técnicas para comparação de resultados	71

6.3 – Análise de desempenho	72
6.4 – Resultados obtidos	74
Capítulo 7 – Conclusão	77
7.1 – Trabalhos Futuros	78
BIBLIOGRAFIA.....	80
Apêndice A – Lookup Table	84

LISTA DE FIGURAS

Figura 1.1 – Exemplo de arte em voxels extraído de ROTD (2012).....	9
Figura 1.2 – Exemplo de objeto gráfico representado por bordo extraído de MODDB (2012). a) Malha de polígonos. b) Domínio de parâmetros. c) Mapa de textura de cores. d) Mapa de textura de normais.	11
Figura 1.3 – Metodologia do método proposto. a) Dados volumétricos originais. b) Extração do bordo com função de atributos para cada face. c) Suavização de normais. d) Suavização de cores. E) Suavização de malha.....	13
Figura 2.1 – Região cúbica com 8 vértices em vermelho e 12 arestas em verde, cada uma com um identificador (BOURKE, 2012).....	15
Figura 2.2 – Configurações de faces geradas a partir do <i>Marching Cubes</i> (LORENSEN e CLINE, 1987).....	15
Figura 2.3 – Configurações de faces geradas a partir do <i>Marching Squares</i> (WIKIPEDIA, 2012). Vértices em preto estão dentro da área do objeto e os vértices brancos estão fora.	16
Figura 2.4 – (a) - Modelo de voxels binário original. (b) - Exemplo de poligonização por Lewiner <i>et al.</i> (2003) com voxels (cubos) cujos vértices não são variedade.....	17
Figura 2.5 – Usando os pontos de interseção com a região e as normais nesses pontos, estima-se um vértice interno, mostrado a partir das linhas vermelhas (KOBBELT <i>et al.</i> , 2001).	18
Figura 2.6 – Resultado de uma reconstrução da malha a partir de dados de um scanner 3D. Resultado com <i>Marching Cubes</i> à esquerda e EMC à direita, obtendo melhor detalhamento na boca, nos olhos e mais superfícies pontiagudas (KOBBELT <i>et al.</i> , 2001).	18
Figura 2.7 – Processo do <i>Dual Contouring</i> . Divide-se o volume em octrees, detectando quem está dentro e fora. Encontra-se a posição a partir do argumento mínimo da QEF. Depois, faz-se a triangulação (JU <i>et al.</i> , 2002).	19
Figura 2.8 – Soluções de voxels de Bischoff e Kobbelt (2006) requerem subdivisão de voxels em 5 x 5 x 5 partes menores.	20
Figura 2.9 – Solução de Rossignac e Cardoze (1999) para vértices e arestas não variedade. (a) Mapeamento do objeto original. (b) Perturbação do mapeamento para remover problemas topológicos.	21
Figura 2.10 – Imagem do Yoshi aumentada em 8 vezes. (a) Vizinho mais próximo. (b) Depixelizing Pixel Art (DEPIXELIZING_COMP, 2012).	22
Figura 2.11 – Etapas do Depixelizing Pixel Art (KOPF e LISCHINSKI, 2011). (a) Imagem original. (b) Geração do grafo de similaridade. (c) Resolução de conflitos do grafo de similaridade. (d) Reamostragem do grafo com conflitos resolvidos. (e) Transformação das arestas em curvas. (f) Resultado final.....	22
Figura 2.12 – Etapas da geração de malhas poligonais a partir de modelos de voxels (MUNIZ, 2010). (a) Extração da malha de quadrados. (b) Suavização da malha. (c) Suavização das normais. (d) Suavização das cores.....	24
Figura 2.13 – Exemplo de suavização de malha com voxels (cubos) cujos vértices não são variedade (MUNIZ, 2010).....	24
Figura 2.14 – Vizinhança (em vermelho) de um vértice (em azul) da malha de quadrados em (a) e de triângulos em (b).....	25
Figura 3.1 – (a) exemplos de sólidos degenerados; (b) exemplos de sólidos não degenerados. Imagem extraída de (BOTSCH <i>et al.</i> , 2010).	27

Figura 3.2 – Modelo de voxels extraído de Command & Conquer: Red Alert 2 (© Electronic Arts) é um exemplo de modelo volumétrico representado por decomposição espacial. As partes pintadas pelo usuário são consideradas dentro do volume e as não pintadas estão fora.	29
Figura 3.3 – Exemplo de complexo regular 1-conectado (LAGE, 2006).	32
Figura 3.4 – Exemplo de aresta, destacada em azul, que incide em mais de duas faces (LAGE, 2006).	32
Figura 3.5 – Exemplo de vértice, em azul, com estrela não homeomorfa a um disco em \mathfrak{R}^2 (LAGE, 2006).	33
Figura 3.6 – Exemplo de conjunto de triângulos com orientação no sentido anti-horário (LAGE, 2006).	33
Figura 3.7 – (a) Exemplo de um complexo regular Σ' que contém vértices e arestas críticas. (b) Centroide de cada célula que induz um vértice no grafo GCDE. (c) Grafo GCDE formado a partir de Σ' com o mapeamento descrito na seção 3.5.; as arestas cheias correspondem à bijeção $\eta_{fe} : \tau \rightarrow e$, as com tracejado largo à bijeção $\eta_{ee} : \mathcal{E} \rightarrow \mathcal{E}$ e as com pontilhado fino a bijeção $\eta_{ve} : \mathcal{V} \rightarrow \mathcal{E}$	35
Figura 3.8 – Exemplo de aproximação e interpolação de geometria, respectivamente, a partir da posição dos pontos amostrados (BOTSCH <i>et al.</i> , 2010).	37
Figura 4.1 – Modelo extraído de ROTD (2012). (a) Dados de entrada. (b) Resultado da extração do modelo poligonal, com função de atributos para cada face.	38
Figura 4.2 – Fluxograma do método proposto.	40
Figura 4.3 – Etapas do processo de extração de uma representação poligonal de um modelo implícito. (a) Dados de entrada. (b) Classificação das células. Células externas são brancas, células internas são pretas, células de bordo estão em cinza escuro e as zonas de refinamento estão em cinza claro. (c) Operação de refinamento. (d) Reamostragem nos centros das células de bordo (círculos em cinza) e no centro das adjacências entre as células de bordo (losangos) - em 2D a reamostragem ocorre apenas nos vértices e arestas, mas em 3D também em faces. (e) Dilatação das novas posições de vértices determinadas (quadrados). (f) Extração do bordo após ladrilhamento (não desenhado) que gera as novas células afins.	41
Figura 4.4 – (a) Geração das novas amostras para um sólido com uma aresta crítica, com as respectivas classificações (círculos brancos são posições de vértice externos, círculos cinza são internos e o círculo preto corresponde a reamostragem na adjacência entre duas células). Observe que em torno do círculo preto passa a existir um conjunto de amostras internas que, quando estruturada, dará origem a uma célula afim homeomorfa à $B_1^3(0) \in \mathfrak{R}^3$. (b) Resultado da extração do bordo das células afins geradas. A região do bordo, cujas faces não são visíveis pelo observador, foi omitida para tornar o desenho mais claro.	42
Figura 4.5 – (a) Geração das novas amostras para um sólido com um vértice crítico, com as respectivas classificações (círculos brancos são posições de vértice externos, círculos cinza são internos e o círculo preto corresponde a reamostragem na adjacência entre duas células. Observe que em torno do círculo preto passa a existir um conjunto de amostras internas que, quando estruturada, dará origem a uma célula afim homeomórfica à $B_1^3(0) \in \mathfrak{R}^3$. (b) Resultado da extração do bordo das células afins geradas. A região do bordo, cujas faces não são visíveis pelo observador, foi omitida para tornar o desenho mais claro.	42
Figura 4.6 – Método de mapeamento de voxels internos e externos. (a) Dados de entrada em que voxels pretos são preenchidos pelo usuário e os brancos são transparentes. (b) Cópia, com o volume aumentado em um voxel em todas as direções e com as cores invertidas. (c) Resultado da	

operação do preenchimento do voxel origem (0,0) com cor branca. (d) Resultado obtido a partir da união dos voxels pretos em (a) com os pretos de (c).....	44
Figura 4.7 – Nessa imagem, a região 5 é uma zona de refinamento, pois os voxels 2 e 6, que são de bordo, não contêm arestas em comum, mas estão ligadas por um vértice que incide na região 5. As demais regiões não são zonas de refinamento.	45
Figura 4.8 – Rotulação inicial dos vértices. As posições internas estão desenhadas como círculos pretos e as externas como círculos brancos.	46
Figura 4.9 – Dilatação das posições interiores gerando a rotulação final.	47
Figura 4.10 – Reticulado dual rotacionado de 45. As células têm como centro os vértices criados na etapa de refinamento antes da dilatação. Os demais vértices formam os pontos de interseção do reticulado.	48
Figura 4.11 – Células geradas na região sem vértices críticos	49
Figura 4.12 – Células geradas nas regiões com vértices críticos.	49
Figura 4.13– Os seis casos da <i>Lookup Table</i> da poligonização do método proposto. Vértices em azul fazem parte da superfície, enquanto os em preto estão fora. (a) Tetraedro, com 10 rotações. (b) Pirâmide quadrada, com 24 rotações. (c) Prisma triangular, com 12 rotações. (d) União de duas pirâmides quadradas, com 12 rotações. (e) Dual de tetraedro no cubo, com 8 rotações. (f) Cubo (caso 255 da tabela).	51
Figura 4.14 – Resultado do mapeamento de regiões internas com modelos sem buracos (Muniz, 2010). (a) Kirov Airship (© Electronic Arts). (b) Volume interno pintado pelo artista. (c) O volume interno removido por meio do mapeamento da seção 4.2.1.	53
Figura 4.15 – Região interna de modelos com buracos como o Night Hawk (a) (© Electronic Arts) faz parte da superfície externa (b) do modelo gerado.	53
Figura 4.16 – Exemplo de poligonização que têm células que compartilham vértices. (a) Modelo original representado por cubos. (b) Modelo reamostrado representado por cubos. (c) Modelo original poligonizado por Lewiner et al. (2003). (d) Modelo reamostrado poligonizado por Lewiner et al. (2003) . (e) Modelo original poligonizado por Muniz et al. (2010). (f) Modelo reamostrado poligonizado por Muniz et al. (2010).	55
Figura 4.17 – Mesmo modelo da figura 4.16 poligonizado pela técnica proposta nesta dissertação. (a) Antes de suavizar as normais. (b) Depois de suavizar a malha.	55
Figura 4.18 – Exemplo de poligonização de um modelo que têm células que compartilham arestas. (a) Modelo original representado por cubos. (b) Modelo reamostrado representado por cubos. (c) Modelo original poligonizado por Lewiner et al. (2003). (d) Modelo reamostrado poligonizado por Lewiner et al. (2003) . (e) Modelo original poligonizado por Muniz et al. (2010). (f) Modelo reamostrado poligonizado por Muniz et al. (2010). (g) Modelo extraído pela técnica proposta antes de suavizar a malha. (h) Modelo extraído pela técnica proposta após a suavização de malha.	57
Figura 4.19 – Histograma do Aspect Ratio do modelo GDI Nemesis.	59
Figura 5.1 – Etapas da metodologia proposta. Modelo extraído de ROTD (2012). (a) Modelo obtido após a extração da malha poligonal, com função de atributos para cada face. (b) Atribuição de normais para cada vértice. (c) Atribuição de cores para cada vértice. (d) Suavização de malhas.	60
Figura 5.2 – Fluxograma do método proposto. Este capítulo explica as etapas de pós-processamento.	61
Figura 5.3 – Intuição para detecção de regiões côncavas e convexas na superfície. Seja α e β ângulos de arestas com o vetor normal \vec{n} . (a) Nas regiões côncavas, o módulo de pelo menos um	

dos ângulos entre arestas e normais é maior do que 90° . (b) Nas regiões convexas, o módulo de todos os ângulos entre arestas incidentes ao vértice com a normal é menor do que 90°	63
Figura 5.4 – Região analisada em torno do vértice v . Neste exemplo, v_λ, v'_i , onde $0 \leq i \leq 5$, são pontos que estão no plano paralelo ao plano tangente passando v_λ , que é a base da região amostrada. A escala λ é a distância de v_λ para v	64
Figura 5.5 Estimativa da curvatura em v . Nesta figura n_i , são as normais nas faces f_i e α_i é o ângulo das arestas de f_i que incidem em v_λ	65
A ponderação por α_i tem como objetivo fazer com que a estimativa da curvatura dê menos importância aos triângulos estreitos que incidem em v do que triângulos com maior ângulo no ponto de incidência.....	65
Figura 5.6 – Suavização de atributos no War Miner. (a) Modelo após extração da superfície. (b) Modelo após suavização das normais. (c) Modelo após a suavização de cores.....	67
Figura 5.7 – Suavização de malha no Nod Buggy (© Electronic Arts). (a) Modelo obtido da etapa de extração da superfície de bordo. (b) Suavização da malha com o método proposto na seção 5.2. (c) Suavização da malha de (a) com o método de Muniz <i>et al.</i> (2010) gera problemas na antena traseira do bugre, com várias arestas se colapsando.	68
Figura 6.1 – Imagens dos jogos (© Electronic Arts): (a) Command & Conquer Tiberian Sun. (b) Command & Conquer: Red Alert 2.....	70
Figura 6.2 – Editor de modelos volumétricos Voxel Section Editor III. A versão 1.39.145 foi utilizada nos testes.	71
Figura 6.3 – Resultados obtidos com modelos de alta resolução utilizados nos testes de desempenho. (a) War Miner (WARMINER, 2012). (b) Kirov Airship MKII (KIROV, 2012). (c) GDI Nemesis (NEMESIS, 2012). (d) Mad Dog (MADDOG, 2012).	72
Figura 6.4 – Modelos originais e resultados obtidos de vários modelos. (a) FS TomCat original. (b) FS TomCat após a suavização. (c) Kirov Airship original. (d) Kirov Airship após a suavização. (e) ROTD Buggy original. (f) ROTD Buggy após suavização. (g) Nighthawk original. (h) Nighthawk após suavização. (i) Nod Buggy original. (j) Nod Buggy após a suavização. (k) War Miner original. (l) War Miner após a suavização.....	76
Figura 7.1 – Yoshi da figura 2.10 aumentado em 8 vezes, estendendo o conceito de reamostragem da região volumétrica proposto neste trabalho para imagens.....	79
Figura A.1 – Sistema de coordenadas usado para gerar a Lookup Table e a numeração dos vértices.....	84

LISTA DE TABELAS

Tabela 4.1 – Qualidade e complexidade da malha pelo método proposto.....	58
Tabela 6.1 – Complexidade dos modelos utilizados para verificação do desempenho dos métodos.	73
Tabela 6.2 – Tempo de execução de cada etapa do método proposto (em ms).	73
Tabela 6.3 – Tempo de execução de cada etapa do método usado para comparação (em ms).....	74
Tabela 6.4 – Tempo de execução de cada etapa do método usado para comparação com reamostragem (em ms).	74
Tabela A.1 – Lista de todos os casos da Lookup Table e como cada um é poligonizado.	84

LISTA DE ABREVIATURAS E SIGLAS

3DS MAX	Autodesk 3ds Max – Programa de manipulação de imagens em 3D baseado em bordo
API	Application Programming Interface - Interface de Programação de Aplicativos
B-REP	Boundary Representation – Representação baseada em bordo
CPU	Central Processing Unit – Unidade Central de Processamento
EMC	Extended Marching Cubes – técnica de reconstrução de superfícies poligonais
GPU	Graphics Processing Unit – Unidade de Processamento Gráfico
GNU	GNU's Not Unix – Sistema operacional em código livre
MC	Marching Cubes – técnica de reconstrução de superfícies poligonais
OPENGL	Open Graphics Library
PIXEL	Picture Element – Elemento mínimo da imagem
QEF	Quadric error function – Função quádrlica de erro.
RAM	Random Access Memory – Memória de acesso aleatório
SINC	Sinus cardinalis – Cardinal seno ($\text{sen}(x)/x$)
VOXEL	<i>Volume Pixel</i> – O menor elemento do volume representado por decomposição do espaço.
VXLSE III	Voxel Section Editor III – Editor de imagens volumétricas baseadas em pixels

Capítulo 1 – Introdução

Arte em voxels é uma forma de arte digital na qual os detalhes no volume são representados no nível de voxels (KOPF e LISCHINSKI, 2011).

O suporte geométrico desses volumes é representado por um array tridimensional de voxels em um reticulado uniforme. A função de atributos para cada elemento armazena informações sobre cores e um valor binário de densidade que estabelece se o voxel é sólido ou não. Os modelos que descrevem esses volumes podem conter *aliasing* e descontinuidades como uma forma de representar detalhes ou feições da superfície que o artista tenta expressar. Por este motivo, cada voxel é relevante para a descrição do objeto final. A figura 1.1 ilustra um exemplo de arte com voxels.



Figura 1.1 – Exemplo de arte em voxels extraído de ROTD (2012).

Modelos, como o da figura 1.1, são gerados a partir de editores de voxels que permitem aos usuários editar cada parte do volume manualmente, voxel a voxel, ou usando técnicas para esculpir voxels. Ainda, é possível gerar modelos volumétricos a partir de malhas poligonais. Editores de voxels também podem converter modelos de voxels em malhas poligonais. Esta conversão costuma gerar uma representação que é tipicamente incompatível com o modelo idealizado pelo artista, pois não existem cuidados e tratamentos especiais com descontinuidades e inconsistências topológicas do modelo.

Arte volumétrica foi utilizada em alguns jogos, no final da década de 1990, como uma solução alternativa para representar objetos de três dimensões que eram compostos por um material interno homogêneo e podiam ser manipulados através de transformações euclidianas. Com isso, era possível rodar jogos com gráficos tridimensionais em computadores com placas

gráficas sem aceleração 3D. A difusão do uso de APIs como o OpenGL e Direct3D nos anos 2000, além da concepção de GPUs programáveis e paralelizáveis, fez com que o custo computacional de representar a cena através de voxels fosse maior do que com o uso de polígonos e do mapeamento de atributos em texturas, devido ao cálculo da iluminação. Desde então, a arte em voxels é utilizada principalmente por poucos entusiastas que criam jogos ou filmes.

Atualmente, os editores de gráficos tridimensionais utilizados profissionalmente como 3ds Max (3DSMAX, 2012), Maya (2012), Blender (2012), Cinema 4D Studio (CINEMA4D, 2012), entre outros, representam objetos sólidos homogêneos através de sua superfície de bordo. Esta abordagem é conhecida como *Boundary Representation (B-Rep)*. A superfície de bordo do objeto é descrita através de um poliedro, composto pelas coordenadas de seus vértices no universo e pelo grafo de conectividade que descreve os seus polígonos. A figura 1.2 ilustra um exemplo de modelo B-Rep.

A edição da geometria de malhas poligonais é intuitiva para muitos usuários, pois seu grau de expressividade é coerente com o resultado final visto pelo usuário. No entanto, a edição dos mapas de textura não é uma atividade trivial, já que o resultado final é diferente daquilo que é manipulado pelo usuário, pois um mapa de textura pode ser distorcido ao ser projetado na malha ou ser separado em retalhos que são posicionados em diferentes setores do objeto, como o exemplo ilustrado na figura 1.2b. Alguns programas permitem ao usuário editar as cores da textura diretamente na malha, como o 3D-Coat, (3DCOAT, 2012), mas não fazem isso com outros atributos, como o vetor normal. Geralmente, a edição dos mapas de texturas é feita separadamente do modelo, sendo necessário o uso de múltiplas interfaces ou até mesmo de outros programas para esta finalidade.

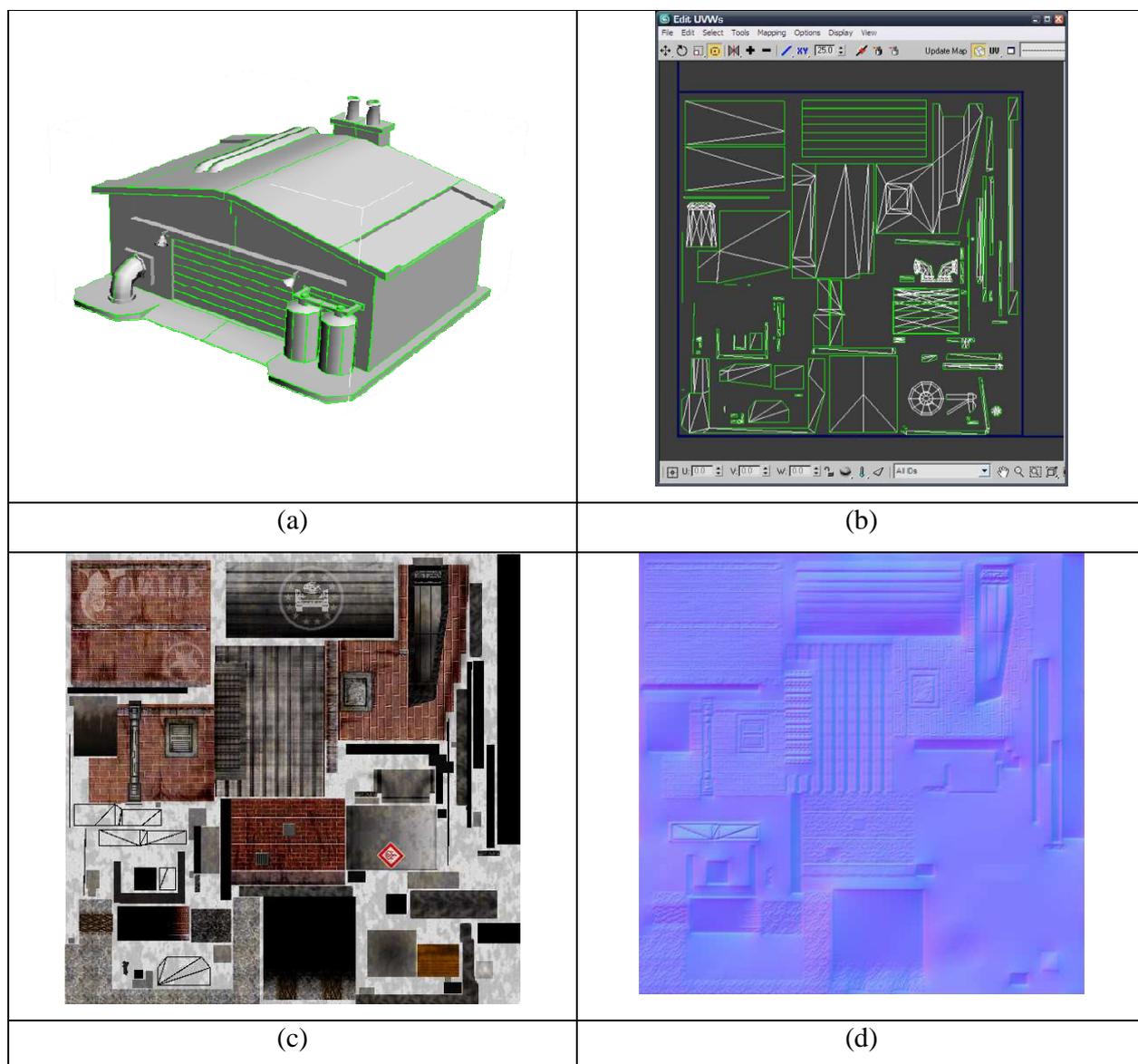


Figura 1.2 – Exemplo de objeto gráfico representado por bordo extraído de MODDB (2012). a) Malha de polígonos. b) Domínio de parâmetros. c) Mapa de textura de cores. d) Mapa de textura de normais.

As dificuldades inerentes da edição de mapas de textura podem ser suprimidas pela abordagem da edição de voxels, que permite ao usuário detalhar partes de objetos de forma mais intuitiva, pois não exige o uso de uma interface diferente para isso, além de permitir que o usuário visualize o resultado de suas ações em tempo real e onde essas mudanças estão ocorrendo. No entanto, a modelagem da geometria do objeto pintando cada voxel é um processo mais demorado do que com o uso de polígonos.

O objeto de estudo desta dissertação é a conversão de dados representados por decomposição espacial através de voxels em malhas poligonais que sejam coerentes com os modelos idealizados pelos artistas. Assim, pode-se contribuir para permitir o uso das técnicas de manipular e esculpir voxels nas ferramentas profissionais de modelagem 3D, permitindo ao usuário fazer uso das duas formas de modelagem ao mesmo tempo. Além disso, pode-se converter dados de jogos que usam modelos volumétricos para serem usados em ferramentas que fazem uso de malhas poligonais.

Este trabalho contribui em três formas relevantes: (1) Propõe uma metodologia para reamostrar modelos volumétricos de tal forma que seja possível reconstruir variedades computacionais de dimensão 3, que podem ser poligonizadas sem a geração de superfícies com buracos ou problemas topológicos comumente observado em técnicas como o *Marching Cubes* de Lewiner *et al.* (2003), entre outras que conseguem extrair bordo de malhas poligonais topologicamente corretas a partir de variedades. (2) Propõe um algoritmo que produz malhas de triângulos a partir de dados volumétricos. A malha gerada é uma variedade combinatória orientável, sendo consistente com a relação de adjacência entre os voxels originais. Além disso, a malha criada por esta técnica é suave e seus triângulos são robustos em relação às métricas de medição de qualidade de malha. (3) Propõe um algoritmo de suavização de malhas que mantém a orientação das concavidades das regiões em torno dos vértices, não permitindo que uma malha que era uma variedade combinatória orientável deixe de ser variedade após a suavização.

A metodologia proposta consiste em um método para extrair o bordo do modelo volumétrico (figura 1.3a), atribuindo inicialmente as funções de atributos para suas faces, seguido de um pós-processamento, onde esses atributos são transferidos para seus vértices (figura 1.3b e 1.3c) e a malha é suavizada (figura 1.3d).

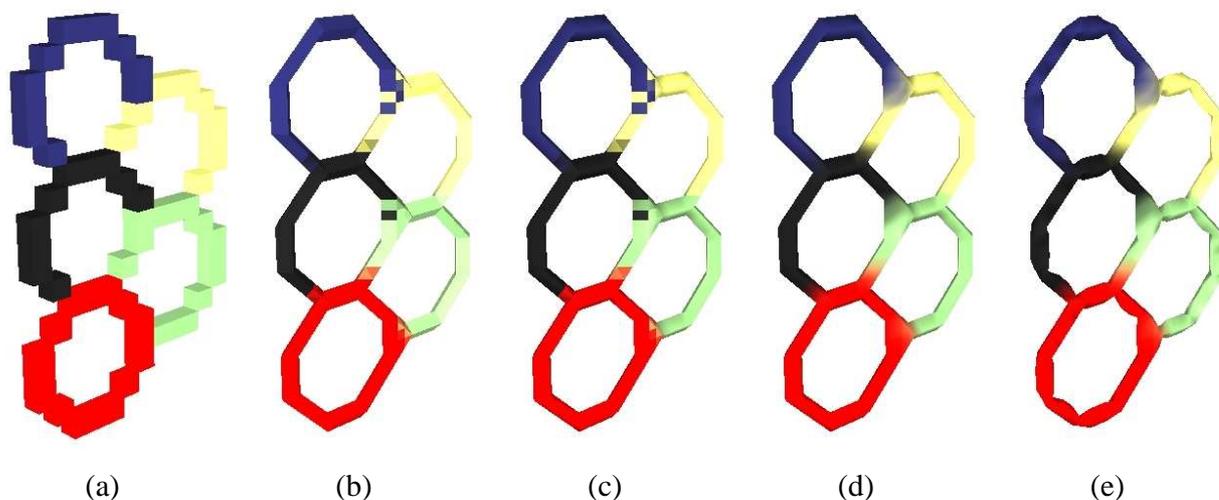


Figura 1.3 – Metodologia do método proposto. a) Dados volumétricos originais. b) Extração do bordo com função de atributos para cada face. c) Suavização de normais. d) Suavização de cores. E) Suavização de malha.

Os testes para a validação do método foram feitos com o Voxel Section Editor III (VXLSE3, 2012), que tem um modelo de dados compatível com o que é utilizado neste trabalho, requerendo poucas adaptações. O código fonte dessa pesquisa encontra-se disponível no SVN do programa (VXLSE3SVN, 2012).

Esta dissertação está organizada da seguinte forma: o capítulo 2 aborda pesquisas anteriores relacionadas à extração de malhas poligonais a partir de dados volumétricos através de filtragem de sinais e detecção de padrões geométricos, além de descrever métodos de restauração de malhas; o capítulo 3 apresenta alguns fundamentos sobre representação de superfícies, topologia combinatória, e descreve o problema da extração de superfícies poligonais a partir de dados volumétricos; o capítulo 4 apresenta e detalha o método de extração de bordo a partir de dados volumétricos proposto neste trabalho; o capítulo 5 explica as etapas de pós-processamento, onde se suaviza cores, normais e a malha poligonal; o capítulo 6 descreve a metodologia e o ambiente utilizado para testes e os resultados do desempenho do algoritmo e, finalmente, o capítulo 7 conclui o trabalho e apresenta sugestões para continuação desta linha de pesquisa.

Capítulo 2 – Trabalhos relacionados

A conversão de modelos volumétricos criados por artistas, para representação por malhas poligonais, é um problema que não foi muito explorado no meio científico. De todos os trabalhos pesquisados, aquele que, em termos de objetivos, mais se aproxima ao que se propõe nesta dissertação é o trabalho de Kopf e Lischinski (2011), que lida com reamostragem e reconstrução de regiões bidimensionais vetoriais a partir de arte em pixels. Até o momento, o único trabalho que aborda o problema dessa natureza, para dados volumétricos, é o trabalho de autoria do próprio autor deste documento (MUNIZ *et al.*, 2010). No entanto, diversas etapas da solução foram estudadas por outros pesquisadores para outras finalidades, etapas que influenciaram na elaboração do método proposto, que será apresentado no capítulo 4.

Este capítulo descreve diversos trabalhos que influenciaram na elaboração da técnica proposta nesta dissertação. O capítulo apresenta, na seção 2.1, uma revisão dos trabalhos em poligonização de dados volumétricos implícitos e busca justificar brevemente porque eles não podem ser aplicados diretamente ao problema aqui abordado, em decorrência de problemas topológicos no dado de origem. Em seguida, na seção 2.2, são descritos alguns trabalhos que tentam corrigir problemas topológicos originários do modelo que podem atrapalhar o processo de poligonização. Na seção 2.3 é apresentada a técnica de reamostragem de imagens concebida por Kopf e Lischinski (2011), que usa o conceito de grafo de similaridade, que influenciou no procedimento de reamostragem do volume da metodologia proposta, para poligonizar o campo escalar correspondente ao modelo volumétrico. Finalmente, na seção 2.4, é apresentado o trabalho de (MUNIZ, 2010) que deu origem a linha de investigação abordada nesta dissertação.

2.1 – Métodos para extração de superfícies a partir de dados volumétricos

O *Marching Cubes* (MC) (LORENSEN e CLINE, 1987) é o algoritmo mais conhecido para extração de uma superfície poligonal a partir de modelos volumétricos. Ele percorre todo o modelo, criando triângulos em regiões cúbicas que intersectam a superfície que delimita o modelo. A cada região cúbica é associado um número de 8 *bits*, onde cada *bit* indica a posição de um dos vértices do cubo em relação ao interior do volume do objeto de interesse, isto é, se ele é interno ou externo ao volume. O algoritmo utiliza esse número para consultar uma tabela, que mostrará a disposição dos vértices e das faces geradas para cada configuração. A figura 2.1 ilustra qual bit identifica cada vértice e aresta.

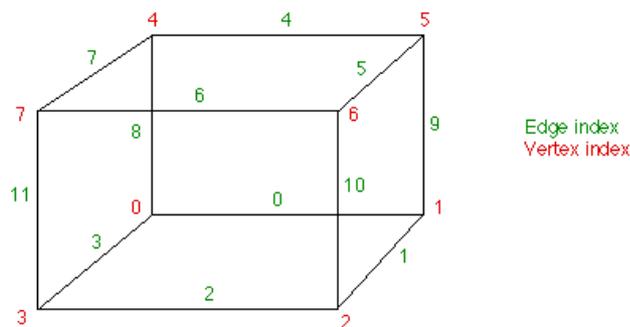


Figura 2.1 – Região cúbica com 8 vértices em vermelho e 12 arestas em verde, cada uma com um identificador (BOURKE, 2012).

As 256 possíveis combinações geradas pelos vértices do cubo permitem a construção de uma tabela de configurações, chamada na literatura de *Lookup Table*, para criar triângulos para cada uma delas. O uso dessa tabela diminui consideravelmente a complexidade do processo de triangularização. As configurações da tabela se resumem a 15 variações mostradas na figura 2.2, porém com rotações diferentes:

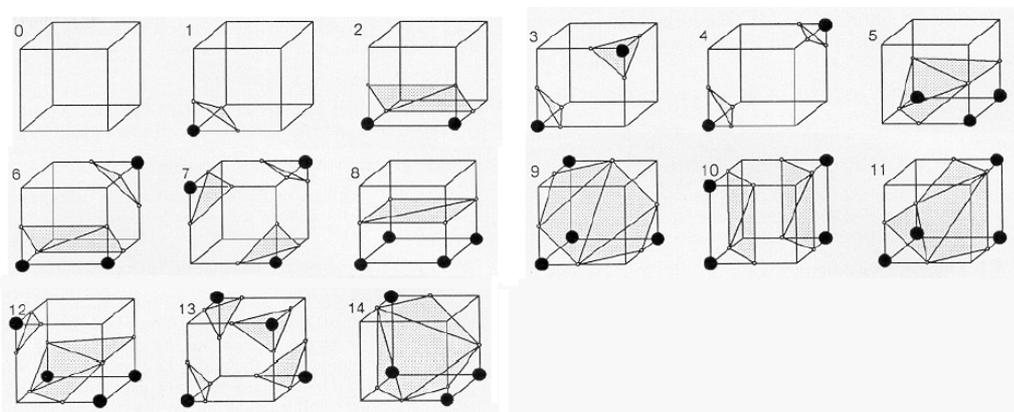


Figura 2.2 – Configurações de faces geradas a partir do *Marching Cubes* (LORENSEN e CLINE, 1987).

O *Marching Squares* é o método análogo ao *Marching Cubes* em duas dimensões. Por ser bidimensional, ele usa uma *Lookup Table* menor do que o *Marching Cubes*, resumida na figura 2.3.

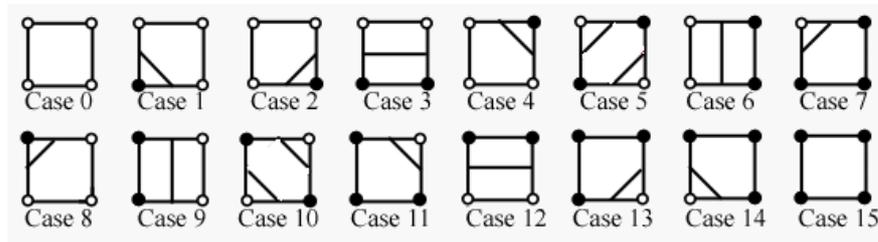


Figura 2.3 – Configurações de faces geradas a partir do Marching Squares (WIKIPEDIA, 2012). Vértices em preto estão dentro da área do objeto e os vértices brancos estão fora.

Ambos o *Marching Squares* e o *Marching Cubes* têm o problema de gerar resultados ambíguos para determinadas configurações. Os casos 5 e 10 do *Marching Squares* são situações em que a topologia local não faz uma distinção clara entre parte interna e externa do objeto. Portanto, ambos os casos podem ser resolvidos de mais de uma forma diferente. O caso 5 também pode ser resolvido com a solução do caso 10 e vice-versa. Analogamente, todos os casos do *Marching Cubes* em que o caso 5 e 10 do *Marching Squares* aparecem na sua topologia também têm soluções ambíguas. Isso significa que os casos 3, 4, 6, 7, 10, 12 e 13 do *Marching Cubes* também precisam de uma amostragem mais precisa para que a superfície correta seja poligonizada.

Para resolver o problema de casos com soluções ambíguas, Chernyaev (1995) elaborou uma nova *Lookup Table* que estende os casos 3, 4, 6, 7, 10, 12 e 13. A tabela de Chernyaev foi implementada por Lewiner *et al.* (2003), que conseguiu gerar variedades combinatórias orientáveis a partir de qualquer modelo volumétrico que não tenha problemas topológicos.

Todas as técnicas baseadas em *Marching Cubes* têm resultados inesperados quando a topologia da superfície do objeto de entrada não separa a parte interna da externa do volume. Na maioria dos métodos mencionados, obtêm-se modelos que não são variedade. O *Marching Cubes* que utiliza a *Lookup Table* de Chernyaev obtém os melhores resultados, sendo seus problemas limitados à presença de buracos indesejados. A figura 2.4(b) mostra o modelo visto na figura 2.4(a) sendo poligonizado pelo *Marching Cubes* de Lewiner *et al.* (2003).

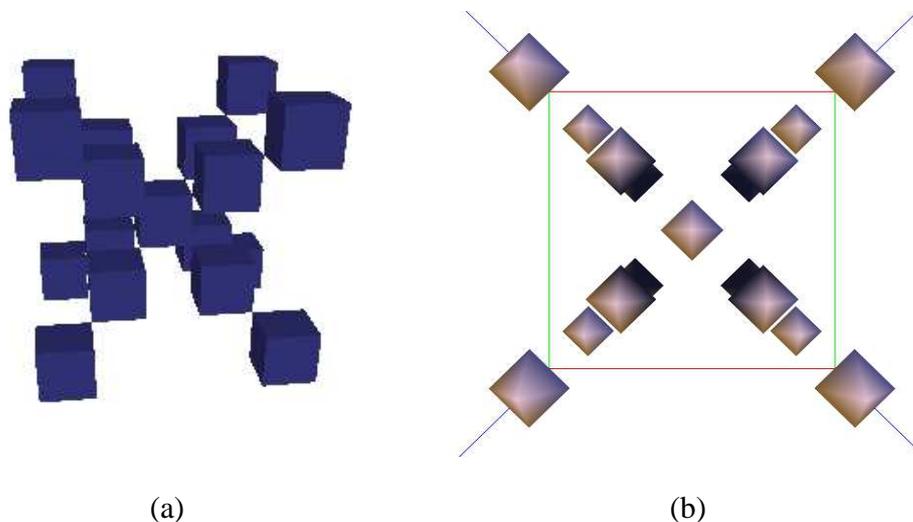


Figura 2.4 – (a) - Modelo de voxels binário original. (b) - Exemplo de poligonização por Lewiner *et al.* (2003) com voxels (cubos) cujos vértices não são variedade.

Os problemas do *Marching Cubes* original não se restringem às questões topológicas. O algoritmo não captura detalhes pontiagudos na superfície, não detecta curvas muito suaves, gera malhas muito densas, entre outras limitações. Por esses motivos, existem várias extensões do *Marching Cubes* na literatura (NEWMAN e YI, 2006).

Uma dessas variações é o *Extended Marching Cubes* (EMC) (KOBBELT *et al.*, 2001) que obtém superfícies que preservam mais detalhes que no método original. Ele usa uma função quádrlica de erro (conhecida como *quadric error function* ou QEF) a partir da posição das amostras originais e suas respectivas normais, para encontrar um novo vértice interno à região. As faces são geradas a partir de leques de triângulos (*triangle fans*) usando esse novo vértice como centro do leque. Com isso, consegue-se o efeito pontiagudo. A figura 2.5 mostra como o vértice central é encontrado em duas dimensões e, portanto, usando segmentos em vez de triângulos. Já a figura 2.6 mostra uma comparação de um modelo gerado a partir do *Marching Cubes* e outro usando o *Extended Marching Cubes*:

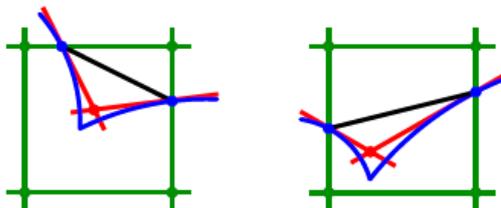


Figura 2.5 – Usando os pontos de interseção com a região e as normais nesses pontos, estima-se um vértice interno, mostrado a partir das linhas vermelhas (KOBBELT *et al.*, 2001).

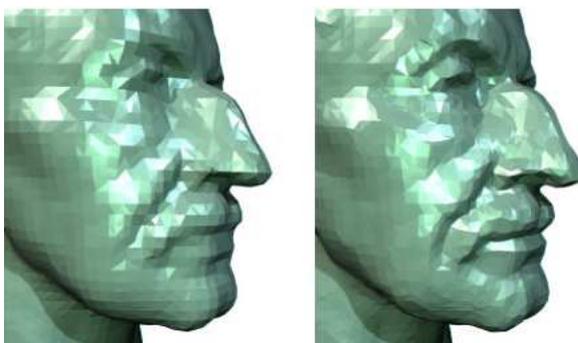


Figura 2.6 – Resultado de uma reconstrução da malha a partir de dados de um scanner 3D. Resultado com *Marching Cubes* à esquerda e EMC à direita, obtendo melhor detalhamento na boca, nos olhos e mais superfícies pontiagudas (KOBBELT *et al.*, 2001).

O *Dual Contouring* (JU *et al.*, 2002) é um algoritmo que extrai as superfícies poligonais a partir de octrees. Ele adapta o cálculo da QEF visto no EMC para encontrar a posição dos vértices, independentemente se há detalhes pontiagudos ou não. A figura 2.7, a seguir, ilustra todas as etapas do algoritmo do *Dual Contouring*.

Os resultados obtidos com *Dual Contouring* têm menos vértices e faces do que os que são obtidos com EMC, devido à estrutura de *octree*. Apesar de o cálculo da QEF ser mais simples e a estrutura de dados mais leve do que nos métodos baseados no *Marching Cubes*, o cálculo da triangulação é feito através de um algoritmo recursivo que não usa uma *Lookup Table*, o que faz com que esse método tenha um custo computacional mais alto do que as outras técnicas.

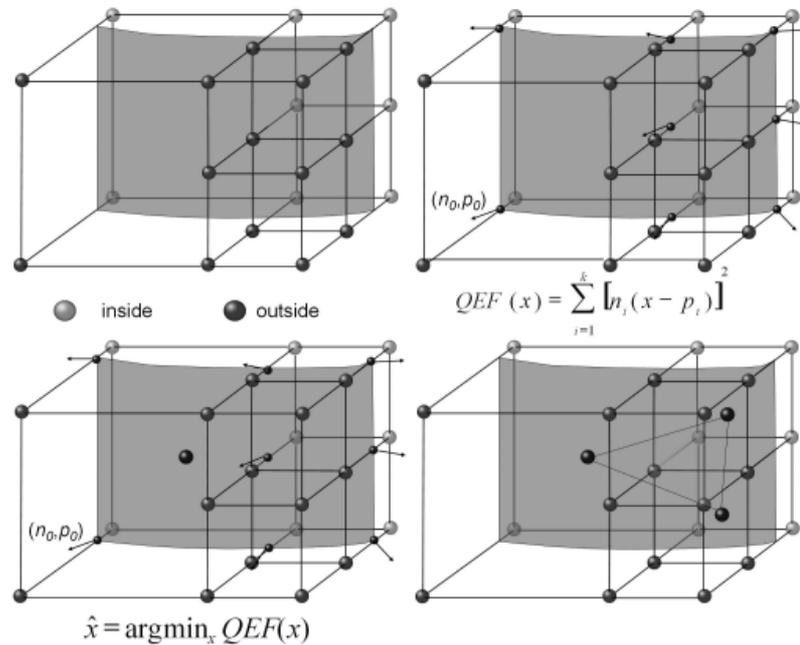


Figura 2.7 – Processo do *Dual Contouring*. Divide-se o volume em octrees, detectando quem está dentro e fora. Encontra-se a posição a partir do argumento mínimo da QEF. Depois, faz-se a triangulação (JU *et al.*, 2002).

O *Dual Marching Cubes* (SCHAEFER e WARREN, 2004) também estende o *Marching Cubes* aplicando-o no dual de octrees, obtendo malhas não uniformes que são topologicamente corretas para entradas que sejam variedades combinatórias.

2.2 – Restauração de vértices e arestas não variedades.

Durante o desenvolvimento deste trabalho, foram feitos estudos de formas de resolver o problema topológico em dados de entrada que não são variedades. Além de se pesquisar formas de resolver esse problema durante a poligonização, também foram investigados, na literatura, meios de resolvê-lo, antes e após a poligonização.

Bischoff *et al.* (2005) propôs uma metodologia para restaurar diversos problemas de malhas poligonais e, entre eles, a presença de vértices e arestas cujas vizinhanças não são homeomórficas uma esfera aberta pertencente ao \mathfrak{R}^3 , denominados vértices e arestas críticas. Na solução proposta, os vértices críticos são substituídos por esferas e as arestas críticas são trocadas por cilindros. Essa solução, apesar de gerar uma superfície de bordo do volume não degenerada, que de fato dividirá a parte externa e a interna do volume, não é coerente geometricamente com o que se espera de um objeto proveniente de um modelo de voxels.

Outra técnica proposta por Bischoff e Kobbelt (2006) tenta resolver esse problema antes da poligonização, ainda no nível de voxel. A figura 2.8 ilustra como se resolve uma aresta não variedade por essa técnica.

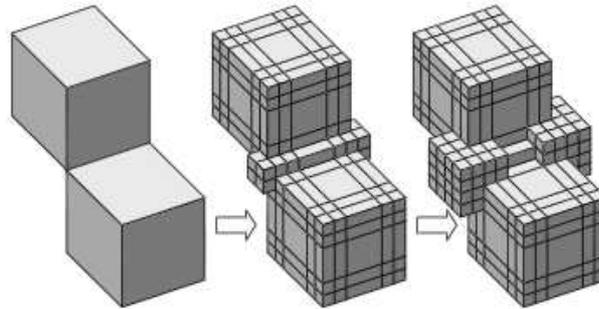


Figura 2.8 – Soluções de voxels de Bischoff e Kobbelt (2006) requerem subdivisão de voxels em 5 x 5 x 5 partes menores.

Enquanto o problema é detectável com mais facilidade no conjunto de voxels, a subdivisão em 125 pedaços encarece consideravelmente a solução em termos de memória e processamento. Além disso, é necessária uma etapa de remalhamento antes da etapa de suavização para obtenção da malha final. No método proposto neste trabalho, a etapa de remalhamento não é necessária, pois a malha obtida no processo de poligonização tem boas características e é coerente com a forma original do modelo de voxels.

Rossignac e Cardoze (1999) apresentaram uma abordagem diferente para resolver o mesmo problema. Eles mapeavam o objeto para fazer pequenas perturbações na superfície, de forma com que ela deixe de ter problemas topológicos. A figura 2.9 ilustra a forma como essas perturbações são feitas.

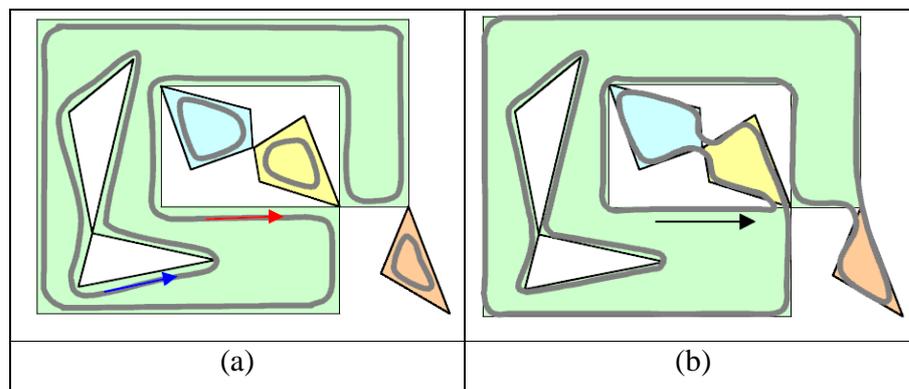


Figura 2.9 – Solução de Rossignac e Cardoze (1999) para vértices e arestas não variedade. (a) Mapeamento do objeto original. (b) Perturbação do mapeamento para remover problemas topológicos.

Essa abordagem não é ideal para dados provenientes de voxels, pois os problemas topológicos desses dados se originam da existência de *aliasing*, que, nesse caso, é a incapacidade de manter a conexão visual entre vizinhos diagonais. A abordagem de Rossignac e Cardoze (1999) resolve os problemas topológicos, mas produz um resultado cuja geometria é incompatível com o que o artista idealiza nas regiões com *aliasing* de seus modelos de voxels.

2.3 – *Depixelizing Pixel Art*

A pesquisa mais próxima do problema proposto por esta dissertação foi publicada por Kopf e Lischinski (2011). Seu objetivo é transformar imagens de baixa resolução, capturadas de jogos criados nos anos 80 e começo dos anos 90, em um conjunto de regiões delimitadas por curvas fechadas sem auto interseção, que permitem a reconstrução dessas imagens em qualquer resolução. Enquanto a técnica de conversão de modelos volumétricos em malhas poligonais proposta nesta dissertação e em Muniz *et al.* (2010) trata a geometria com apenas dois materiais, isto é, sólido ou vazio, a técnica de Kopf e Lischinski (2011) trabalha com múltiplos materiais. Cada material é formado por pixels de cores semelhantes e receberá um conjunto de vetores próprios. Para lidar com a presença de múltiplos materiais no modelo de dados, deve-se determinar qual material se sobressai em áreas em que há conflitos, ou seja, em que dois ou mais materiais diferentes disputam o mesmo espaço.

O resultado do uso dessa técnica para aumentar a amostragem do personagem Yoshi, de direitos autorais da Nintendo (2012), na figura 2.10(a) em 8 vezes é mostrado na figura 2.10(b). Suas etapas estão ilustradas de forma sucinta na figura 2.11.

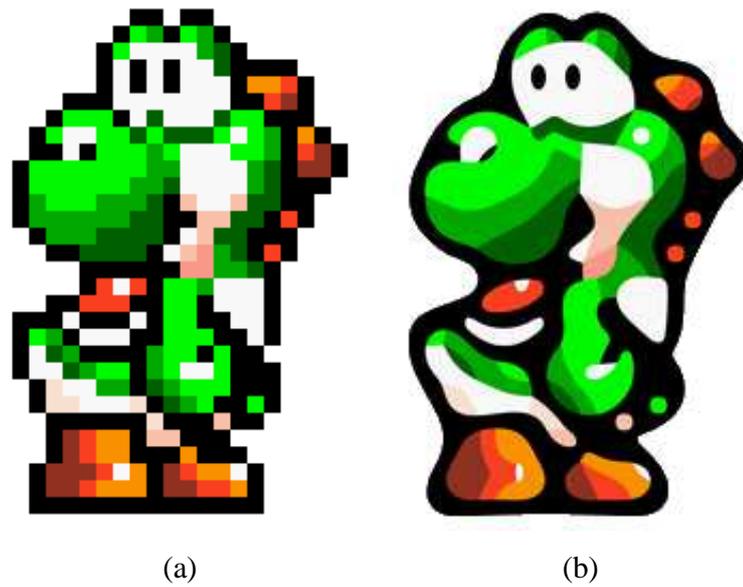


Figura 2.10 – Imagem do Yoshi aumentada em 8 vezes. (a) Vizinho mais próximo. (b) Depixelizing Pixel Art (DEPIXELIZING_COMP, 2012).

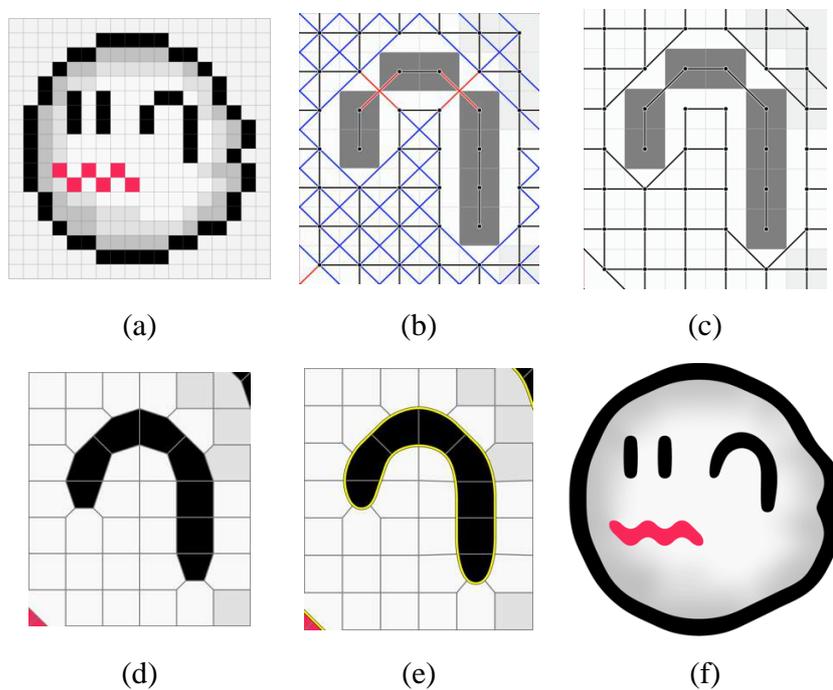


Figura 2.11 – Etapas do Depixelizing Pixel Art (KOPF e LISCHINSKI, 2011). (a) Imagem original. (b) Geração do grafo de similaridade. (c) Resolução de conflitos do grafo de similaridade. (d) Reamostragem do grafo com conflitos resolvidos. (e) Transformação das arestas em curvas. (f) Resultado final.

2.4 – Conversão de voxels em polígonos através de filtragem de sinais.

A técnica concebida por Muniz (2010) gera malhas de triângulos extraíndo a superfície que delimita o modelo de voxels original. Inicialmente, os voxels do modelo são mapeados em internos, externos e superfícies. Depois, cada voxel da superfície é extraído como um conjunto de quadrados formando a parte externa de um cubo, cujo centroide fica localizado na posição do voxel original. Portanto, a obtenção das posições do conjunto de vértices do modelo é feita a partir de uma reamostragem por meio de interpolação de vizinho mais próximo das posições do centroide de cada voxel (figura 2.12a), sendo que vértices que não fazem parte de nenhum polígono no modelo final são eliminados.

Em seguida, utiliza-se um operador Laplaciano uniforme composto com um núcleo de Lanczos adaptado para reconstruir uma malha (figura 2.12b) visualmente aproximada daquilo que é idealizado pelo artista, fazendo uma suavização da malha original. Após esse procedimento, cada quadrado é transformado em dois triângulos através da geração de uma nova aresta que liga dois vértices opostos.

Neste trabalho também é feita uma reconstrução de atributos como normais (figura 2.12c) e cores (figura 2.12d) da malha que finalmente é simplificada. A figura 2.12 (FSTOMCAT, 2012) ilustra as etapas desse processo.

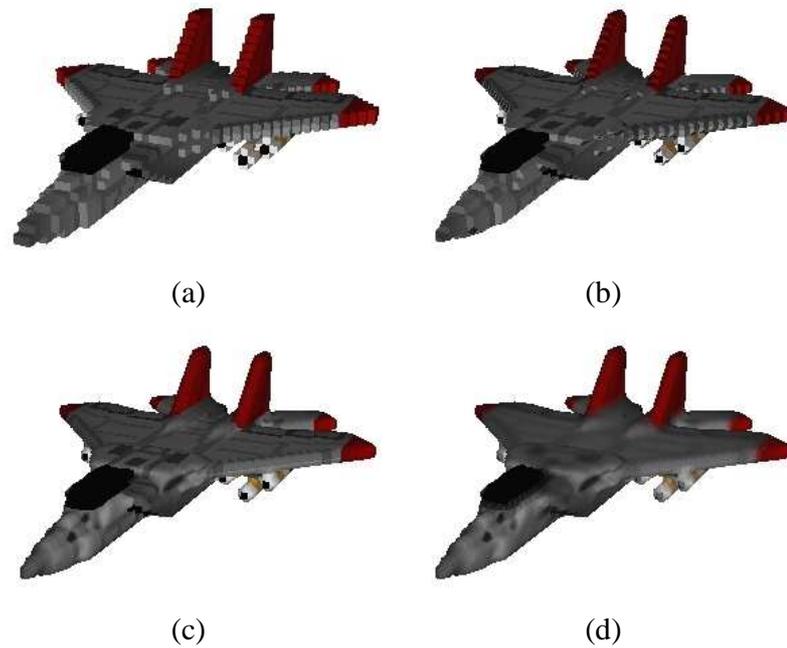


Figura 2.12 – Etapas da geração de malhas poligonais a partir de modelos de voxels (MUNIZ, 2010). (a) Extração da malha de quadrados. (b) Suavização da malha. (c) Suavização das normais. (d) Suavização das cores.

A técnica apresentada gera resultados inesperados com modelos volumétricos cujo campo escalar que, descreve o volume, contenha configurações derivadas dos casos 5 e 10 do Marching Squares. Nesses casos a operação de suavização e qualquer outra operação de processamento da malha serão comprometidas nas regiões com problemas topológicos. A figura 2.13 ilustra o resultado dessa técnica em um modelo em que todos os voxels têm esse problema.

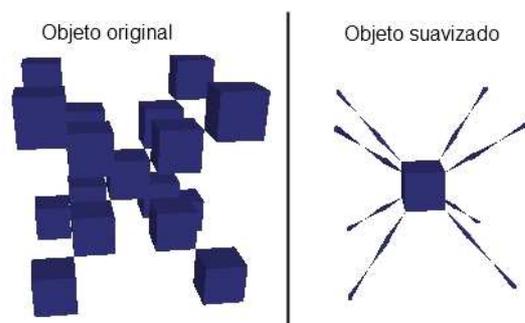


Figura 2.13 – Exemplo de suavização de malha com voxels (cubos) cujos vértices não são variedade (MUNIZ, 2010).

A transformação de quadrados em triângulos faz com que informações importantes de vizinhança sejam perdidas, como mostrado na figura 2.14. Isto afeta negativamente o resultado de todas as operações de suavização de atributos. Por isso essa operação é feita depois da suavização de malhas.

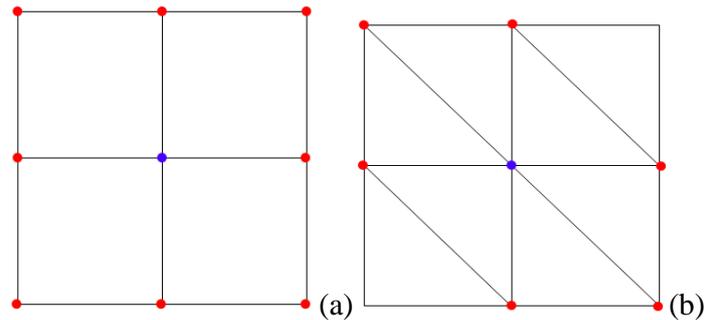


Figura 2.14 – Vizinhança (em vermelho) de um vértice (em azul) da malha de quadrados em (a) e de triângulos em (b).

O trabalho de Muniz *et al.* (2010) estende Muniz (2010) melhorando a forma com que as normais são calculadas em cada vértice, fazendo alterações no filtro de Lanczos, entre outras modificações, melhorando sensivelmente o resultado obtido. No entanto, as malhas poligonais geradas por esta extensão continuam não sendo variedades combinatórias.

O fato do método de Muniz (2010) e Muniz *et al.* (2010) não corrigir o problema dos modelos volumétricos, que não são variedades combinatórias, foi um dos grandes motivadores para o desenvolvimento desta dissertação.

Capítulo 3 – Fundamentos

Este capítulo descreve os principais conceitos sobre objetos volumétricos utilizados neste trabalho, incluindo formas de descrição matemática, representação computacional e noções de topologia combinatória. Seu objetivo é estabelecer notações e enunciar os resultados conceituais necessários para a formulação do problema de extração de malhas poligonais a partir de modelos volumétricos criados por artistas.

3.1 – Objetos Volumétricos

Definição: Um *objeto volumétrico* ou *sólido* é um subconjunto limitado $V \subset \mathfrak{R}^3$, tal que para todo ponto $p \in V$, existe uma vizinhança aberta $B_\varepsilon^3(p) \subset \mathfrak{R}^3$ tal que $B_\varepsilon^3(p) \cap V$ é homeomorfa a esfera unitária $B_1^3(0) \in \mathfrak{R}^3$,

$$B_1^3(0) = \{(x, y, z) \in \mathfrak{R}^3; x^2 + y^2 + z^2 < 1\} \quad (3.1)$$

ou à esfera unitária $\tilde{B}_1^3(0) \in \mathfrak{R}_+^3$

$$\tilde{B}_1^3(0) = \{(x, y, z) \in \mathfrak{R}^3; x^2 + y^2 + z^2 < 1, z \geq 0\}. \quad (3.2)$$

A definição de objeto volumétrico, exibida acima, não indica como caracterizar matematicamente a geometria de um sólido, nem como representá-lo de forma discreta, necessária em uma implementação computacional. Esses assuntos são tratados nas seções 3.2 e 3.3.

É importante observar que o *bordo* M de um sólido $V \subset \mathfrak{R}^3$ construído de acordo com a definição acima é uma *2-variedade orientável sem bordo* imersa no \mathfrak{R}^3 , ou simplesmente, *superfície sem bordo*. A definição de uma 2-variedade é análoga à de um sólido:

Definição: Uma 2-variedade é um conjunto de pontos $M \subset \mathfrak{R}^3$, tal que para todo ponto $p \in M$ existe uma vizinhança aberta $B_\varepsilon^2(p) \subset \mathfrak{R}^3$ tal que $B_\varepsilon^2(p) \cap V$ é homeomorfa ao disco unitário $B_1^2(0) \in \mathfrak{R}^3$,

$$B_1^2(0) = \{(x, y, z) \in \mathfrak{R}^3; x^2 + y^2 < 1, z = 0\} \quad (3.3)$$

ou ao disco unitário $\tilde{B}_1^2(0) \in \mathfrak{R}_+^3$

$$\tilde{B}_1^2(0) = \{(x, y, z) \in \mathfrak{R}^3; x^2 + y^2 < 1, y \geq 0, z = 0\}. \quad (3.4)$$

Quando uma superfície não tem bordo, apenas o primeiro tipo de homeomorfismo, descrito pela equação (3.3), ocorre.

3.2 – Descrição geométrica de objetos volumétricos

É possível caracterizar geometricamente um objeto volumétrico V de duas formas: utilizando uma *descrição por bordo*, ou através de uma *descrição por funções implícitas*. A seguir, as duas abordagens serão descritas e relacionadas.

3.2.1 – Descrição por bordo

Na descrição por bordo, o sólido V é caracterizado por sua superfície de bordo M . Reciprocamente, pode-se pensar que se M é uma superfície sem bordo então M separa apropriadamente o interior do exterior do sólido não degenerado V , não existindo em V nenhuma estrutura infinitesimalmente fina, formada por pontos p que não satisfaçam a definição de sólido apresentada na seção 3.1. A figura 3.1(a) ilustra exemplos de sólidos degenerados; a estrutura degenerada da figura de cima é o vértice comum aos dois cubos, enquanto na figura de baixo é a aresta compartilhada pelos cubos. A figura 3.1(b) ilustra exemplos de sólidos onde as regiões degeneradas foram corrigidas.

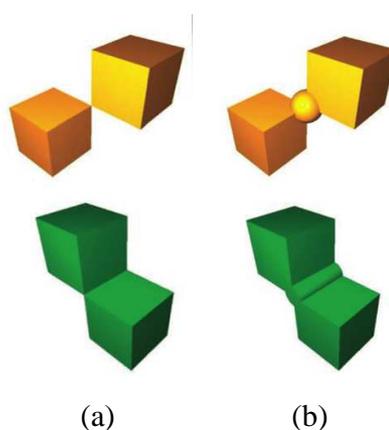


Figura 3.1 – (a) exemplos de sólidos degenerados; (b) exemplos de sólidos não degenerados. Imagem extraída de (BOTSCH *et al.*, 2010).

3.2.2 – Descrição implícita.

Um sólido tridimensional V pode ser caracterizado implicitamente através de uma função $f: \Omega \subseteq \mathfrak{R}^3 \rightarrow \mathfrak{R}$, como a função característica que assume valores negativos no interior e positivos no exterior de V .

Neste contexto, assumindo que a função f é diferenciável, dizemos que $p \in V$ é um ponto singular se $\nabla f(p) = 0$ e chamamos $f(p)$ de valor singular. Se um valor $a \in \mathfrak{R}$ não é um valor singular de f , o chamamos de valor regular. Pode-se então definir superfícies através do Teorema da Função Implícita, enunciado a seguir:

Teorema da Função Implícita: Seja $f: \Omega \subseteq \mathfrak{R}^3 \rightarrow \mathfrak{R}$ uma função de classe C^k , se $a \in \mathfrak{R}$ é um valor regular de f , então $S = f^{-1}(a)$ é uma superfície em \mathfrak{R}^3 de classe C^k .

Tomando $a=0$, sem perda de generalidade, podemos definir o interior, o exterior e a fronteira de V como o conjunto dos pontos $p \in \mathfrak{R}^3$ tais que $f(p) < 0$, $f(p) > 0$ e $f(p) = 0$, respectivamente. A função f também pode ser interpretada fisicamente como a função que descreve a densidade nos pontos do espaço.

3.3 - Representação discreta de objetos volumétricos

Em Computação Gráfica, objetos volumétricos são representados de forma discreta para que possam ser manipulados no computador. Esta representação pode ser feita seguindo duas estratégias: através de uma *decomposição espacial* do espaço ambiente ou pela *representação poligonal* do bordo. Nas próximas duas subseções as abordagens serão descritas e comparadas.

3.3.1 - Representação por decomposição espacial

A representação por *decomposição espacial* de um objeto volumétrico V , consiste em subdividir o espaço em um conjunto de células volumétricas $C = \{v_0, v_1, \dots, v_n\}$ e fazer uma enumeração das células que intersectam o objeto, isto é, obter o conjunto $RP(V) = \{v_i \mid v_i \cap V \neq \emptyset\}$, e reamostrar os atributos do objeto em cada uma das células.

Quando a decomposição é feita através de um reticulado uniforme, a representação é denominada *decomposição uniforme*, e dá origem a uma *representação matricial*. Essa representação pode ser obtida através do produto cartesiano de partições uniformes dos eixos coordenados, formando um reticulado. Uma representação matricial de um objeto volumétrico é denominada *representação volumétrica*. Cada célula do reticulado é denominada *voxel*, que define um paralelepípedo do espaço.

Neste trabalho, o sólido é descrito através de uma representação volumétrica editada por artistas. Para cada voxel, atribui-se um valor binário indicando quando este foi “pintado” pelo artista (voxel interno) ou não (voxel externo). Além disso, o artista também define os atributos de cor dos voxels, armazenando uma cor para cada posição amostrada do volume. A figura 3.2 ilustra um exemplo de modelo volumétrico considerado neste trabalho.



Figura 3.2 – Modelo de voxels extraído de Command & Conquer: Red Alert 2 (© Electronic Arts) é um exemplo de modelo volumétrico representado por decomposição espacial. As partes pintadas pelo usuário são consideradas dentro do volume e as não pintadas estão fora.

Uma das vantagens de uma representação volumétrica é que é possível utilizar operações análogas às que se aplicam em Processamento Imagens, uma vez que a representação volumétrica pode ser vista como uma imagem tridimensional; como desvantagem cita-se a necessidade de enumeração de um grande número de elementos para definir uma forma com muitos detalhes ou manipular a geometria do sólido.

3.3.2 - Representação por bordo

A *representação por bordo* de um objeto volumétrico V , consiste em representar a superfície de bordo do sólido por um conjunto de polígonos $C=\{t_0,t_1,\dots,t_n\}$ que formam uma aproximação poliédrica do sólido original.

A representação por bordo é adequada quando os atributos do sólido no seu interior não variam como, por exemplo, em uma peça mecânica, não sendo tão apropriada quando os atributos variam praticamente em toda a região como, por exemplo, em dados médicos.

Uma das hipóteses deste trabalho é que a representação volumétrica é conveniente quando o dado foi adquirido diretamente na forma volumétrica, em particular, durante a edição de modelos através da *pintura de voxels*. Por outro lado, sabe-se que a representação volumétrica é inapropriada para armazenamento, visualização e processamento geométrico, sendo mais conveniente nesses casos a representação por bordo. Por estes motivos, pode-se argumentar que o problema de conversão entre representações é de extrema relevância para o sucesso da criação de modelos sólidos através de pintura de voxels.

Para especificar de forma precisa o problema de conversão entre a representação volumétrica e uma representação por bordo de um sólido, a próxima seção introduzirá algumas noções de topologia combinatória, com base em Lewiner *et al.* (2010) e Lage (2006).

3.4 – Noções de topologia combinatória

Diferentemente de outros trabalhos, aqui consideram-se modelos de voxels produzidos por artistas, que não satisfazem necessariamente a definição de sólidos. Isto torna o processo de extração de malhas poligonais a partir de modelos volumétricos mais complexo, pois não é possível garantir que a malha gerada aproxima uma 2-variedade orientável sem bordo, essencial para todo o tipo de processamento gráfico consistente como, por exemplo, renderização.

Esta seção apresenta algumas noções de conceitos de topologia combinatória, Estes conceitos serão importantes para a compreensão de como representar uma variedade através dos objetos de interesse deste trabalho: modelos de voxels e malhas poligonais.

3.4.1 – Complexos celulares e complexos simpliciais

Seja n a dimensão do espaço ambiente. Uma *célula afim* imersa em \mathfrak{R}^n é o *fecho convexo* de um conjunto finito de pontos. Uma *célula de dimensão p* em \mathfrak{R}^n é uma imagem homeomorfa de uma célula afim em \mathfrak{R}^p .

Um caso particular de célula afim são os *simplexos*. Um *simplexo* σ de dimensão k em um espaço com n dimensões, é uma célula afim formada por $k+1$ pontos $\{v_0, v_1, \dots, v_k\}, v_i \in \mathfrak{R}^n$, em posição geral. Por convenção, um simplexo de dimensão k é chamado de *k -simplexo*. Ainda, um 0 -simplexo é chamado de ponto, um 1 -simplexo de segmento, um 2 -simplexo de triângulo.

Uma p -célula γ gerada a partir de um subconjunto dos pontos da k -célula σ é chamada de uma *p -face* de σ . Quando não há ambiguidade, pode-se omitir a dimensão da célula γ , denotando-a apenas como face de σ . Uma p -célula γ será uma *face própria* da k -célula σ quando p for menor do que k . Uma 0 -face é o conjunto vazio e é face própria de todas as células. O *bordo* de uma célula σ , denotado por $\partial\sigma$, é o conjunto de todas as faces próprias de σ .

Duas células σ e γ de dimensão k são *p -adjacentes* se a interseção entre elas for uma p -face. Se $k = p + 1$, então σ e γ são adjacentes, omitindo sua dimensão p . Caso não haja interseção entre σ e γ , elas são independentes.

Um *complexo celular* Σ é um conjunto finito de células tal que todas as faces de uma célula σ em Σ também pertencem à Σ e a interseção entre duas células de Σ será uma face própria delas. A dimensão d de um d -complexo celular é a maior dimensão de uma célula pertencente a ele. Se todas as células de Σ forem simplexos, então Σ é chamado de *complexo simplicial*. Um *subcomplexo celular* Σ^* de um complexo celular Σ é um complexo celular formado por qualquer subconjunto de células de Σ .

Seja Σ um d -complexo celular imerso em \mathfrak{R}^n , sendo $0 < d \leq n$. Um *poliedro* de Σ , denotado por $|\Sigma|$, é o subconjunto de \mathfrak{R}^n definido pela união de todas as células de Σ .

A *estrela* de uma célula σ que pertence a um complexo celular Σ , denotada por $Star(\sigma, \Sigma)$, é a união de todas as células $\gamma \in \Sigma$ que tem σ como face. O *elo* de uma célula σ que pertence a um complexo celular Σ , denotado por $Link(\sigma, \Sigma)$, é a união de todas as células $\gamma \in \Sigma$ tal que γ é face de uma célula que pertença à $Star(\sigma, \Sigma)$, mas γ não faz parte dessa estrela.

Uma célula σ que pertence a um complexo celular Σ será considerado uma *célula de topo* se $Star(\sigma, \Sigma) = \sigma$. Um *complexo regular* é um d -complexo celular em que todas as suas células de topo são d -células.

Duas células σ e γ são *h-conectadas* se existe uma sequência de células $(\gamma_i)_{i=0}^n$ tal que σ e γ são faces de γ_0 e γ_n , respectivamente, e células consecutivas da sequência são *h-adjacentes*. Um subcomplexo celular Σ^* de um complexo celular Σ será uma *h-componente conexa* de Σ se todas as suas células forem *h-conectadas*.

3.4.2 – Variedades combinatórias

Um d -complexo celular Σ é uma *d-variedade combinatória* se satisfaz três propriedades:

1) É um complexo regular $(d-1)$ -conectado. A figura 3.3 ilustra um exemplo de um complexo simplicial 1-conectado.

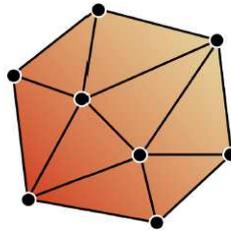


Figura 3.3 – Exemplo de complexo regular 1-conectado (LAGE, 2006).

2) Toda $(d-1)$ -célula do sólido deverá incidir em, no máximo, duas d -células. A figura 3.4 ilustra uma aresta não variedade incidente a mais de dois triângulos.

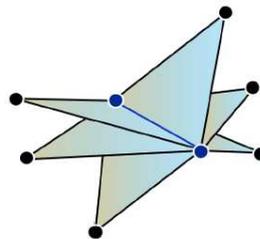


Figura 3.4 – Exemplo de aresta, destacada em azul, que incide em mais de duas faces (LAGE, 2006).

3) A estrela de qualquer vértice do complexo celular Σ é homeomorfa a \mathfrak{R}^d ou \mathfrak{R}_+^d . A figura 3.5 ilustra um complexo simplicial que viola esta propriedade.

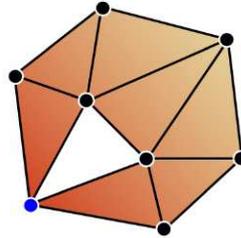


Figura 3.5 – Exemplo de vértice, em azul, com estrela não homeomorfa a um disco em \mathfrak{R}^2 (LAGE, 2006).

Uma d -variedade combinatória é *orientável* se for possível escolher uma orientação coerente para todas as suas células, ou seja, duas k -células adjacentes sempre induzem orientações opostas à $(k-1)$ -célula que compartilham. A figura 3.6 ilustra um complexo simplicial que é uma 2-variedade combinatória orientável.

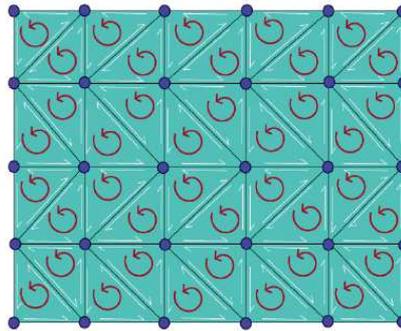


Figura 3.6 – Exemplo de conjunto de triângulos com orientação no sentido anti-horário (LAGE, 2006).

3.5 – Grafo de conectividade

Seja $V \subset \mathfrak{R}^3$ uma região volumétrica, descrita através de uma decomposição celular uniforme, e Ω a 3-variedade combinatória associada a V . Os modelos criados pelos artistas e objeto de estudo deste trabalho são complexos regulares $\Sigma \subset \Omega$ tais que se uma célula $\sigma \in \Sigma$, então $\sigma \cap \Omega \neq \partial\Omega$. Deste modo garante-se que os objetos são envolvidos em uma 3-variedade combinatória Ω e suas células não tem interseção com o seu bordo $\partial\Omega$.

Supondo que $\Sigma \subset \Omega$ é um sólido não degenerado, é possível definir o *grafo de conectividade dual* associado a Σ , denotado por $GCD = (VE, ED)$ onde VE é o conjunto de vértices e ED o conjunto de arestas do grafo. Os conjuntos VE e ED são definidos através de

duas bijeções: $\eta_c : \sigma \rightarrow v$, onde $\sigma \in \Sigma$ é uma célula de Σ e $v \in VE$ é um vértice de VE ; $\eta_f : \tau \rightarrow e$ que para cada 2-face $\tau \in \Sigma$, adjacente a duas 3-células σ e σ' , associa uma aresta $e = (u, v) \in ED$, $u \in VE$ e $v \in VE$, tal que $\eta_c(\sigma) = u$ e $\eta_c(\sigma') = v$.

Seja $\Sigma' \subset \Omega$ é um sólido degenerado. Neste trabalho, denomina-se de *1-face crítica* ou *aresta crítica* ε , toda aresta incidente a duas células σ e σ' , $\sigma \cap \sigma' = \varepsilon$. Analogamente, uma *0-face crítica* ou *vértice crítico* v é um vértice incidente a duas células σ e σ' , $\sigma \cap \sigma' = v$. Observe que as definições de vértice e aresta críticos são particulares para a estrutura combinatória de Ω .

É possível definir o *grafo de conectividade dual estendido* associado a Σ' , no caso degenerado, denotado por $GCDE = (VE', ED')$ onde VE' é o conjunto de vértices e ED' o conjunto de arestas do grafo. Os conjuntos VE' e ED' são definidos através de uma bijeção $\eta_c : \sigma \rightarrow v$, onde $\sigma \in \Sigma'$ é uma célula de Σ' e $v \in VE'$ é um vértice de VE' . O conjunto de arestas é definido por três tipos diferentes de bijeções, descritas na equação 3.3, que dependem do sistema de vizinhanças entre as células de Σ' :

$$\begin{cases} \eta_{fe} : \tau \rightarrow e \\ \eta_{ee} : \varepsilon \rightarrow e \\ \eta_{ve} : v \rightarrow e \end{cases} \quad (3.3)$$

As bijeções fazem o seguinte mapeamento entre as d-faces de Σ' e o grafo $GCDE$:

- Para cada 2-face $\tau \in \Sigma'$, adjacente a duas células σ e σ' , $\eta_{fe} : \tau \rightarrow e$ associa uma aresta $e = (u, v) \in ED'$, tal que $\eta_c(\sigma) = u$ e $\eta_c(\sigma') = v$;
- Para cada *aresta crítica* $\varepsilon \in \Sigma'$, $\eta_{ee} : \varepsilon \rightarrow e$ associa uma aresta $e = (u, v) \in ED'$, tal que $\eta_c(\sigma) = u$, $\eta_c(\sigma') = v$, $\sigma, \sigma' \in \Sigma'$;
- Para cada *vértice crítico* $v \in \Sigma'$, $v \notin \varepsilon$, onde ε é uma aresta crítica, $\eta_{ve} : v \rightarrow e$, associa uma aresta $e = (u, v) \in ED'$, tal que $\eta_c(\sigma) = u$, $\eta_c(\sigma') = v$, $\sigma, \sigma' \in \Sigma'$.

A figura 3.7 ilustra um exemplo de $GCDE$ sendo construído a partir de um complexo regular degenerado Σ' .

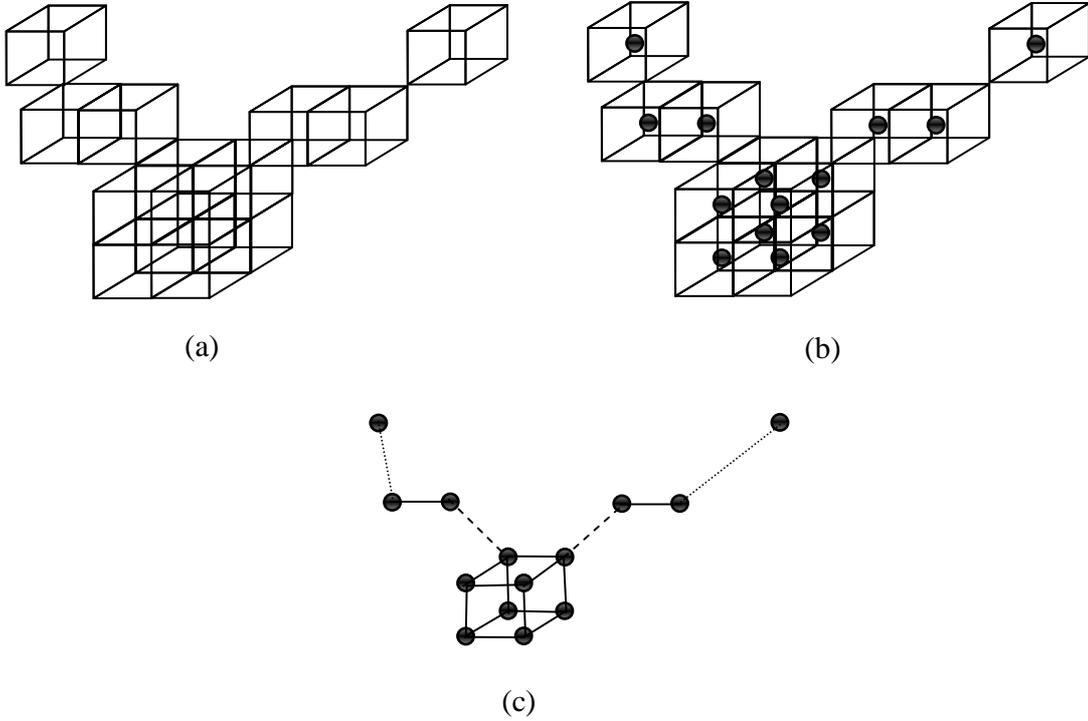


Figura 3.7 – (a) Exemplo de um complexo regular Σ' que contém vértices e arestas críticas. (b) Centroide de cada célula que induz um vértice no grafo GCDE. (c) Grafo GCDE formado a partir de Σ' com o mapeamento descrito na seção 3.5.; as arestas cheias correspondem à bijeção $\eta_{fe} : \tau \rightarrow e$, as com tracejado largo à bijeção $\eta_{ee} : \mathcal{E} \rightarrow \mathcal{E}$ e as com pontilhado fino a bijeção $\eta_{ve} : \mathcal{V} \rightarrow \mathcal{E}$.

Neste trabalho, as células do complexo regular Σ receberão quatro classificações distintas. Uma célula $\sigma \in \Sigma$ ($\sigma \in \Sigma'$) é interior se toda 2-face $\tau \subset \sigma$ é adjacente a somente duas células distintas de Σ (Σ'). Já uma célula $\omega \in \Omega$ é exterior se $\omega \cap \sigma = \emptyset$ ou $\omega \cap \sigma = \tau$, $\tau \subset \sigma$ é 2-face, para toda célula $\sigma \in \Sigma$ ($\sigma \in \Sigma'$). Além disso, uma célula $\sigma \in \Sigma$ ($\sigma \in \Sigma'$) é de bordo se $\exists \omega, \sigma \cap \omega \neq \emptyset$ e $\sigma \cap \omega$ é uma ou mais 2-face e uma célula $\omega \in \Omega$ é denominada célula de refinamento se $\exists \sigma_1, \sigma_2$, $\sigma_1 \cap \omega \neq \emptyset$, $\sigma_2 \cap \omega \neq \emptyset$ e $\sigma_1 \cap \sigma_2$ é uma 1-face ou 0-face. Essa classificação será utilizada posteriormente para selecionar as regiões que farão parte do bordo do objeto que será gerado pelo método proposto nesta dissertação.

A partir deste momento, no texto, o termo voxel será utilizado para referenciar as células de três dimensões do complexo celular

3.6 – Poligonização de superfícies implícitas

Seja $V = \{(x, y, z) \in \mathfrak{R}^3; F(x, y, z) \leq 0\}$ um objeto volumétrico representado de forma implícita pela função F . Pelo teorema da função implícita, enunciado na seção 3.2.2, o bordo de V é descrito pela superfície $S = F^{-1}(0)$. O problema de poligonização de superfícies implícitas consiste em encontrar uma triangulação que aproxime a superfície S .

Muitos métodos de poligonização de superfícies implícitas seguem uma metodologia que envolve três etapas:

- a) Obtenção de uma subdivisão do espaço.
- b) Solução do problema local em cada subdivisão.
- c) Reestruturação.

O processo de subdivisão pode ser feito através de uma decomposição do espaço em um complexo celular formado por um conjunto de células σ . Tipicamente se considera dois tipos de complexos celulares: um complexo simplicial onde cada célula σ é um tetraedro, ou um complexo celular onde cada célula σ é um cubo, como no método *Marching Cubes*.

A solução do problema de poligonização local consiste em aproximar a função implícita $F(x, y, z)$ localmente por uma função afim $\tilde{F} : \sigma \rightarrow R$, no interior de cada célula e calcular $S = \tilde{F}^{-1}(0)$.

Neste trabalho, a superfície implícita que descreve o bordo do modelo é dada por amostras discretas. A construção de superfícies paramétricas como, por exemplo, uma malha de triângulos, a partir de amostras discretas de uma superfície implícita, requer o estabelecimento de uma continuidade entre essas amostras, isto é, uma relação de vizinhança consistente entre as amostras, que se manifesta na existência de uma variedade da qual as amostras são obtidas.

Uma estimativa da superfície da qual provêm as amostras do objeto pode ser feita a partir de aproximação ou interpolação dos dados amostrados em torno de sua vizinhança, através de funções que ponderam esses valores por meio de pesos, utilizando critérios pré-determinados. A diferença entre a aproximação e a interpolação é que, na segunda, os atributos amostrados originalmente não são alterados. No entanto, a aproximação permite que ruídos sejam ignorados

gerando resultados suaves, porém com possível perda de detalhes relevantes. A figura 3.8 ilustra a estratégia de aproximar dados à esquerda e de interpolar dados à direita.



Figura 3.8 – Exemplo de aproximação e interpolação de geometria, respectivamente, a partir da posição dos pontos amostrados (BOTSCH *et al.*, 2010).

O tipo de dado de entrada considerado neste trabalho corresponde a uma amostragem discreta de um volume implícito que não necessariamente corresponde a uma variedade tridimensional. Para resolver este problema, é proposta uma abordagem que faz uma reamostragem do campo escalar dos dados implícitos, de forma que os vértices da malha gerada correspondam a amostras de uma variedade bidimensional coerente com os dados originais. Se não houver esse processo de reamostragem, não haverá garantias de que a topologia do objeto final seja homeomorfa à do objeto original.

Existem várias situações em que isso pode ocorrer. Um exemplo é a situação na qual os dados de entrada possuem voxels isolados. Nesse caso, será impossível formar triângulos que compõem uma vizinhança ao seu redor usando apenas um vértice no centroide dessa região.

Portanto, a resolução do problema proposto deverá obrigatoriamente ter um procedimento de aumento da amostragem do campo escalar que define o volume original, e será necessário extrair um novo conjunto de vértices a partir das posições obtidas com essa reamostragem. Depois disso, os vértices serão conectados por arestas e faces, seguindo algum critério pré-definido, formando o objeto final.

Capítulo 4 – O método proposto

Neste capítulo, é formulado matematicamente o problema de extração de malhas poligonais a partir de representações volumétricas de objetos criados por artistas gráficos (ver Figura 4.1) e o método proposto para resolvê-lo. A seção 4.1 descreve o problema a ser resolvido. A seção 4.2 apresenta uma visão geral do método proposto e depois explica cada etapa, e finalmente, na seção 4.3, são apresentados os resultados obtidos e é feita comparações com propostas anteriores.



Figura 4.1 – Modelo extraído de ROTD (2012). (a) Dados de entrada. (b) Resultado da extração do modelo poligonal, com função de atributos para cada face.

4.1 – O problema de extração do bordo de modelos volumétricos não variedade

O objetivo deste trabalho é extrair malhas que represente o bordo de dados volumétricos descritos por decomposição espacial uniforme. Uma das características que distingue o problema aqui abordado é o de que os dados volumétricos são modelos gerados por artistas gráficos, que frequentemente correspondem a sólidos degenerados. O problema então consiste em extrair uma representação de uma variedade bidimensional a partir de um modelo volumétrico, que não necessariamente corresponde a uma variedade combinatória tridimensional, produzindo, entretanto, uma geometria coerente com a idealizada pelo modelador.

4.1.1 – Definição do problema

Problema de extração de malhas de representações volumétricas. Seja V' um modelo volumétrico representado por um complexo celular $\Sigma' \subset \Omega$, que pode não corresponder a uma variedade 3D, deseja-se obter uma 3-variedade $\bar{\Sigma}$ que não tem obrigatoriamente a mesma estrutura combinatoria Ω mas que é o *grafo primal* do grafo de conectividade *GCDE* de V' . A malha procurada deve ser o bordo $\partial\bar{\Sigma}$ e corresponder a uma 2-variedade combinatoria.

4.2 - O Método proposto

A solução aqui apresentada, para o problema definido na seção 4.1.1, é uma solução construtiva. A ideia é partir do grafo de conectividades dual estendido do complexo regular não variedade Σ' , construir um novo complexo celular que seja o primal do grafo anterior e posteriormente extrair seu bordo. Com isto, há a garantia de que, por construção, cada face no novo complexo seja adjacente a somente duas células, se a face for interior ao volume, e somente a uma célula, no caso de uma face que faz parte do bordo do volume.

No método proposto, o grafo de conectividades não é construído explicitamente. Um processo de reamostragem e geração de células afins, que localmente são dadas pelo fecho convexo das novas amostras obtidas, é o que garante a correção da topologia nas vizinhanças não homeomorfas à esfera unitária. A reamostragem é escolhida de tal modo que a topologia seja modificada somente nas arestas e vértices críticos, mantendo a topologia original nas regiões não degeneradas.

O método proposto neste trabalho, para resolver o problema definido na seção 4.1.1, é composto de quatro etapas, ilustradas no fluxograma apresentado na figura 4.2.

A primeira etapa consiste em classificar as células do complexo celular regular Ω associado ao espaço ambiente, em *exterior*, *interior*, *bordo* e *refinamento*, em relação ao modelo volumétrico representado por Σ' . As células do tipo *interior* e *exterior* não receberão polígonos, ao contrário daquelas cujo tipo é *bordo* e *refinamento*, que estão na fronteira entre o interior e o exterior do objeto. As células de *refinamento*, que estão na parte externa dessa fronteira, farão parte do processo de poligonização para duas finalidades: para modificar localmente a topologia de Σ' , permitindo assim a obtenção de uma variedade bidimensional que será a malha extraída ou para melhorar o aspecto geométrico da região poligonizada. No método aqui apresentado,

constrói-se um complexo intermediário $\bar{\Sigma}$, por um processo de poliedralização, a partir do qual $\partial\bar{\Sigma}$ será obtido. A etapa 1 está ilustrada na figura 4.3b e será explicada em detalhes na seção 4.2.1.

A segunda etapa tem como objetivo realizar uma reamostragem do campo escalar (figura 4.3c), dado pela própria função característica do volume, definido sobre o complexo celular Σ' associado V' , produzindo como resultado um novo conjunto de vértices, a partir do qual um novo complexo regular intermediário $\bar{\Sigma}$ será construído. Essa etapa, descrita na seção 4.2.2, é feita por meio de dois passos: no primeiro obtém-se um vértice, que é o centroide de cada célula de bordo de Σ' , juntamente com a criação de um novo vértice no centroide das d -faces, $d=0,\dots,2$, incidentes a duas células de bordo de Σ' (figura 4.3d); no segundo, utiliza-se um operador de dilatação, para gerar seis novos vértices no interior da célula original (figura 4.3e).

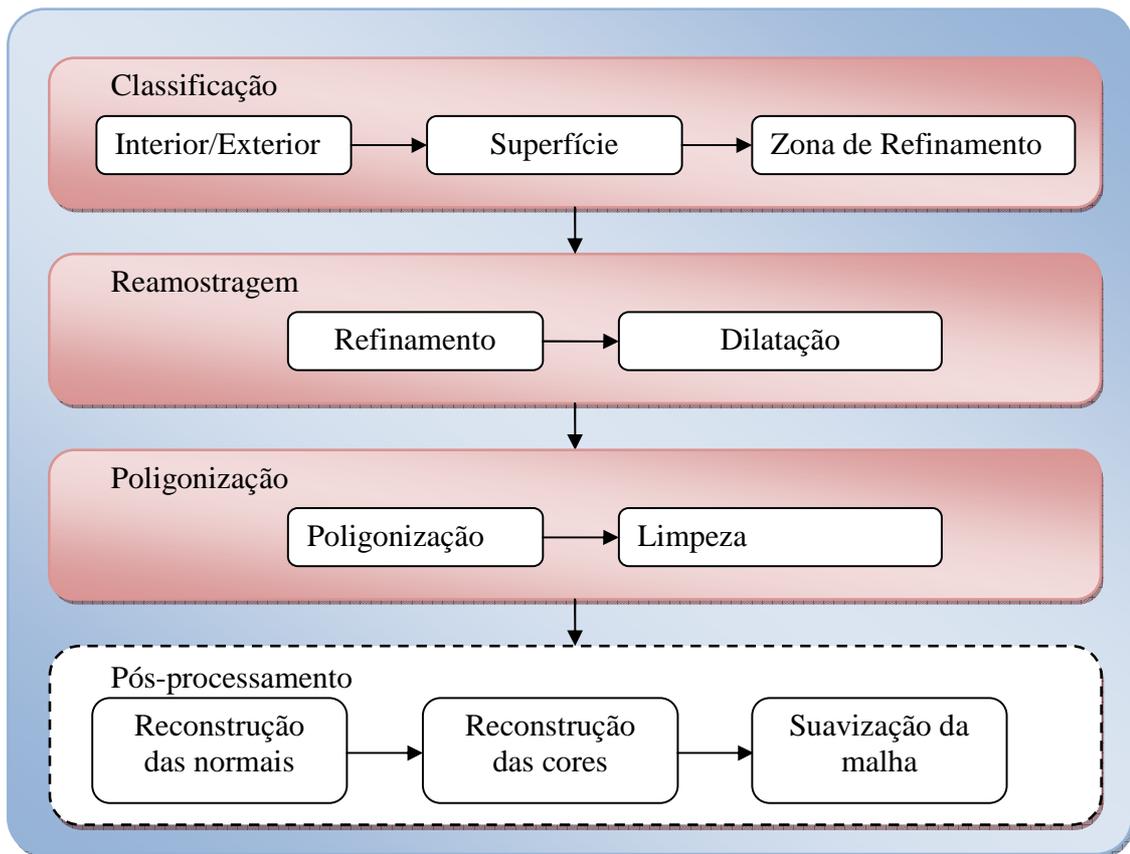


Figura 4.2 – Fluxograma do método proposto.

Na terceira etapa, mostrada na seção 4.2.3, gera-se um conjunto de poliedros para os novos vértices gerados para cada célula original usando uma *Lookup Table*. O bordo $\partial\bar{\Sigma}$ será

obtido de $\bar{\Sigma}$ eliminando-se as faces internas dos poliedros criados e unindo todas as faces que são partes do bordo do volume (figura 4.3f). Finalmente, efetua-se a limpeza de vértices desnecessários.

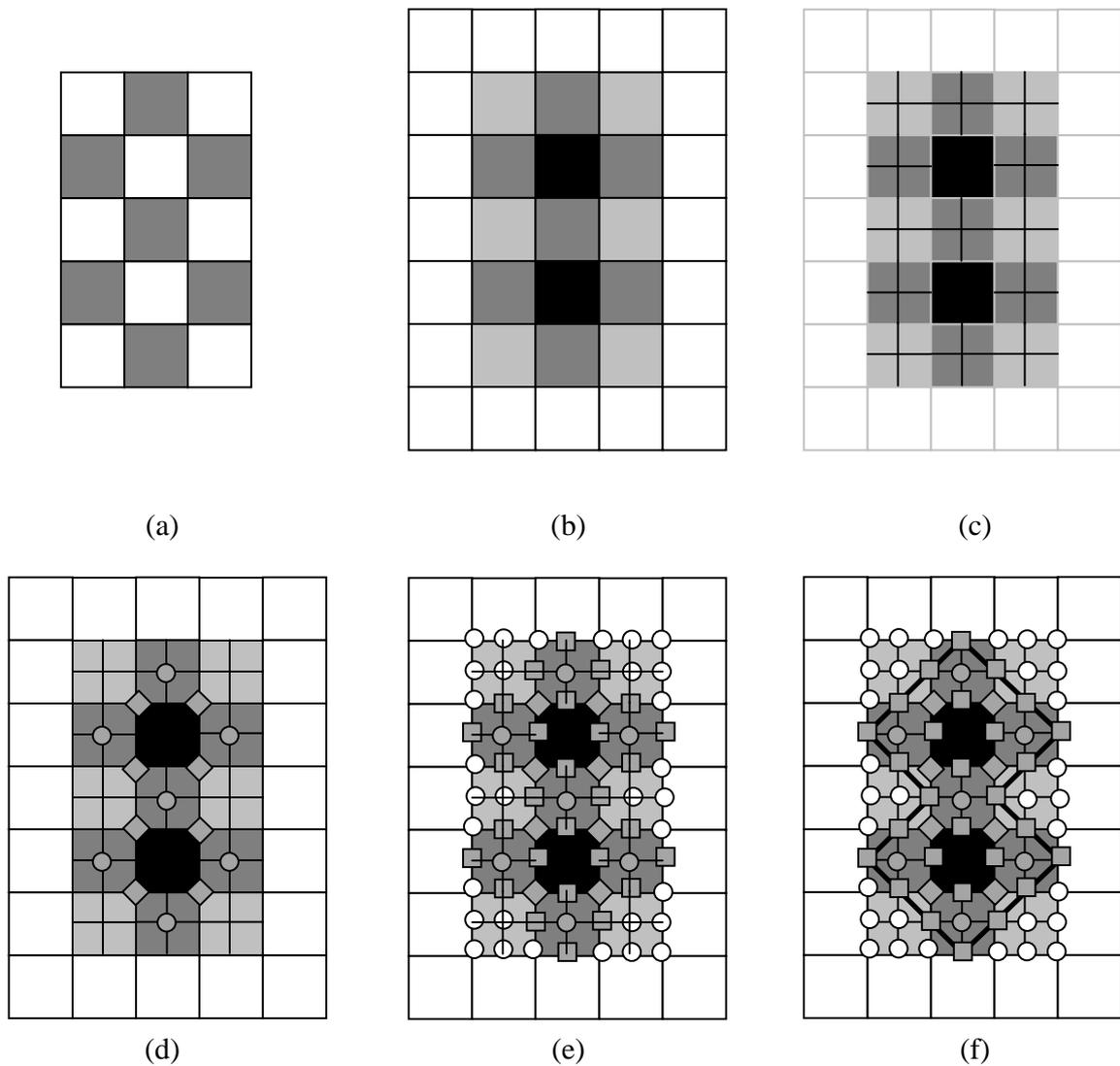


Figura 4.3 – Etapas do processo de extração de uma representação poligonal de um modelo implícito. (a) Dados de entrada. (b) Classificação das células. Células externas são brancas, células internas são pretas, células de bordo estão em cinza escuro e as zonas de refinamento estão em cinza claro. (c) Operação de refinamento. (d) Reamostragem nos centros das células de bordo (círculos em cinza) e no centro das adjacências entre as células de bordo (losangos) - em 2D a reamostragem ocorre apenas nos vértices e arestas, mas em 3D também em faces. (e) Dilatação das novas posições de vértices determinadas (quadrados). (f) Extração do bordo após ladrilhamento (não desenhado) que gera as novas células afins.

As figuras 4.4 e 4.5 mostram a geração da reamostragem e extração de bordo sendo feitos com sólidos que tenham arestas e vértices críticos respectivamente. Pode-se observar que nessas figuras que a região em torno dos vértices que serão poligonizados são homeomorfas à uma esfera em \mathfrak{R}^3 .

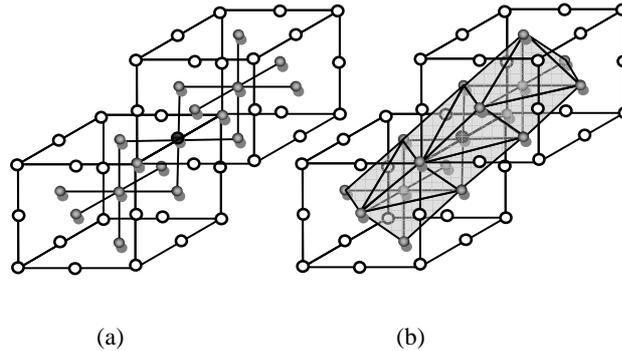


Figura 4.4 – (a) Geração das novas amostras para um sólido com uma aresta crítica, com as respectivas classificações (círculos brancos são posições de vértice externos, círculos cinza são internos e o círculo preto corresponde a reamostragem na adjacência entre duas células). Observe que em torno do círculo preto passa a existir um conjunto de amostras internas que, quando estruturada, dará origem a uma célula afim homeomorfa à $B_1^3(0) \in \mathfrak{R}^3$. (b) Resultado da extração do bordo das células afins geradas. A região do bordo, cujas faces não são visíveis pelo observador, foi omitida para tornar o desenho mais claro.

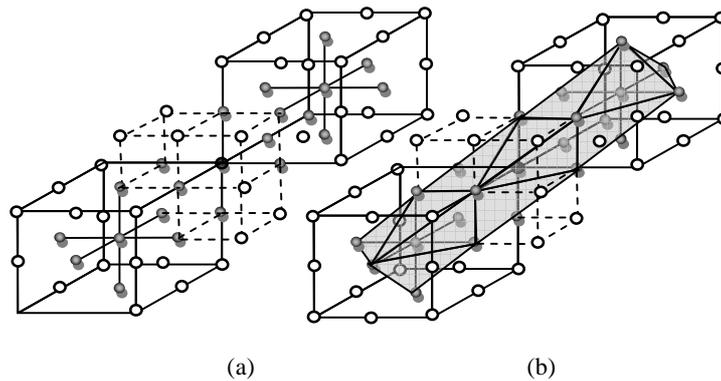


Figura 4.5 – (a) Geração das novas amostras para um sólido com um vértice crítico, com as respectivas classificações (círculos brancos são posições de vértice externos, círculos cinza são internos e o círculo preto corresponde a reamostragem na adjacência entre duas células). Observe que em torno do círculo preto passa a existir um conjunto de amostras internas que, quando estruturada, dará origem a uma célula afim homeomórfica à $B_1^3(0) \in \mathfrak{R}^3$. (b) Resultado da extração do bordo das células afins geradas. A região do bordo, cujas faces não são visíveis pelo observador, foi omitida para tornar o desenho mais claro.

4.2.1 – Classificação dos voxels

A classificação dos voxels do modelo volumétrico inicia-se com a detecção dos voxels que estão dentro e fora do volume. A simples detecção de voxels pintados pelo artista não é suficiente para o reconhecimento da parte interna e do bordo do volume, pois é muito comum que o artista não pinte todos os elementos internos do volume. O método usado nesta dissertação para a classificação de voxels internos, externos e de bordo foi proposto por Muniz (2010) e será descrito a seguir.

O problema de se efetuar a distinção entre voxels internos e externos pode ser resolvido através de um método simples (figura 4.6), utilizando preenchimento de regiões. Primeiramente, distinguem-se os voxels pintados pelo usuário (cor preta) dos voxels não pintados (cor branca), como na figura 4.6a. Para preencher os voxels não pintados, no interior do sólido, faz-se uma cópia dos dados originais em uma área de memória (BufferAuxiliar), adicionando-se uma borda de tamanho igual a um voxel. Nessa cópia (figura 4.6b), a cor dos voxels é invertida. Em seguida, aplica-se um processo de preenchimento, similar ao algoritmo *flood-fill*, a partir da posição (0,0,0), onde a cor dos voxels pretos é substituída por branco. Como resultado, isola-se os voxels interiores não pintados pelo usuário, os quais, agora, recebem a cor preta, como ilustrado na figura 4.6c. Finalmente, combinando-se o dado original, com a cópia alterada pelo processo de preenchimento, tem-se uma nova versão do modelo, onde o objeto forma um sólido compacto (figura 4.6d). O pseudocódigo a seguir descreve todas as etapas:

Algoritmo 4.1 Procedimento para distinguir voxels internos e externos.

Requer: Σ' , dimX, dimY, dimZ {onde dimX, Y e Z são as dimensões X, Y e Z de $\bar{\Sigma}$ }.

{ Cria-se uma cópia de Σ' com dados invertidos em BufferAuxiliar. }

novo BufferAuxiliar[dimX + 2, dimY + 2, dimZ + 2];

para voxel(x,y,z) representando um voxel de Σ' **faça**

BufferAuxiliar[x + 1, y + 1, z + 1] \leftarrow inverte(voxel.tipo);

fim para

{ Faz-se o preenchimento do elemento (0,0,0) do BufferAuxiliar com o valor C_EXTERIOR, a cor branca no texto. }

nova Fila();

Fila \leftarrow coordenadas(0,0,0);

BufferAuxiliar[0,0,0] \leftarrow C_EXTERIOR;

enquanto Fila não estiver vazia, desenfileirar as coordenadas (x,y,z) **faça**

se BufferAuxiliar[x + 1, y, z] \neq 0 **então**

Fila \leftarrow coordenadas (x + 1, y, z);

BufferAuxiliar[x + 1, y, z] \leftarrow C_EXTERIOR;

```

fim se
se BufferAuxiliar[x - 1,y, z]  $\neq$  0 então
    Fila  $\leftarrow$  coordenadas (x - 1, y, z);
    BufferAuxiliar[x - 1, y, z]  $\leftarrow$  C_EXTERIOR;
fim se
se BufferAuxiliar[x, y + 1,z]  $\neq$  0 então
    Fila  $\leftarrow$  coordenadas (x, y + 1, z);
    BufferAuxiliar[x, y + 1, z]  $\leftarrow$  C_EXTERIOR;
fim se
se BufferAuxiliar[x, y - 1, z]  $\neq$  0 então
    Fila  $\leftarrow$  coordenadas (x, y - 1, z);
    BufferAuxiliar[x, y - 1, z]  $\leftarrow$  C_EXTERIOR;
fim se
se BufferAuxiliar[x, y, z + 1]  $\neq$  0 então
    Fila  $\leftarrow$  coordenadas (x, y, z + 1);
    BufferAuxiliar[x, y, z + 1]  $\leftarrow$  C_EXTERIOR;
fim se
se BufferAuxiliar[x, y, z - 1]  $\neq$  0 então
    Fila  $\leftarrow$  coordenadas (x, y, z - 1);
    BufferAuxiliar[x, y, z - 1]  $\leftarrow$  C_EXTERIOR;
fim se
fim enquanto
{Escreve-se o resultado final.}
para voxel(x,y,z) representando um voxel de  $\Sigma'$  faça
    se (voxel.tipo = C_EXTERIOR) e (BufferAuxiliar[x+1,y+1, z+1] = C_INTERIOR) então
        voxel.tipo  $\leftarrow$  C_INTERIOR
    fim se
fim para

```

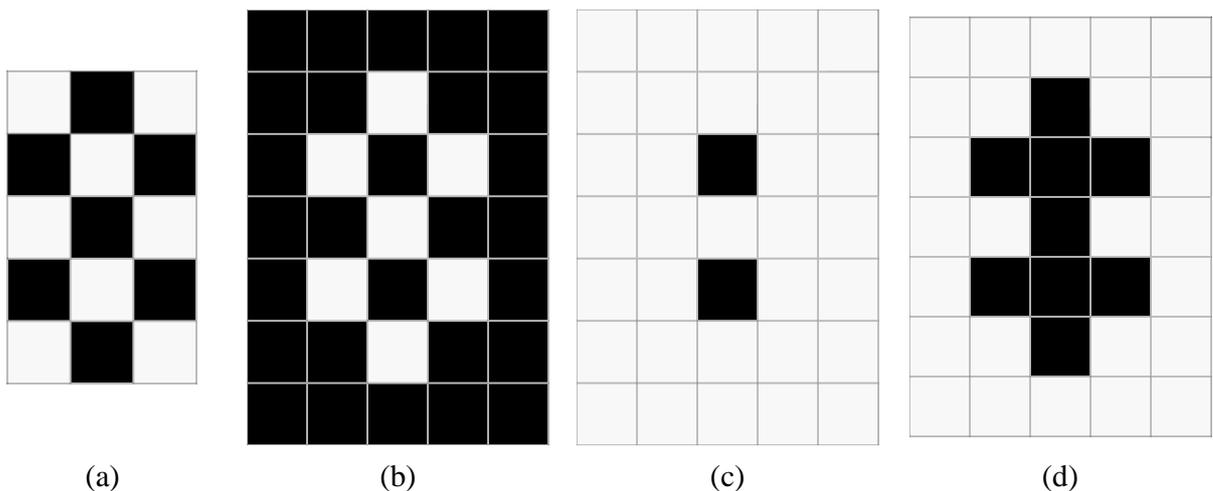


Figura 4.6 – Método de mapeamento de voxels internos e externos. (a) Dados de entrada em que voxels pretos são preenchidos pelo usuário e os brancos são transparentes. (b) Cópia, com o volume aumentado em um voxel em todas as direções e com as cores invertidas. (c) Resultado da operação do preenchimento do voxel origem (0,0) com cor branca. (d) Resultado obtido a partir da união dos voxels pretos em (a) com os pretos de (c).

Em seguida, detectam-se os voxels que estão no bordo do volume e os voxels que fazem parte das zonas de *refinamento*. Estes voxels correspondem respectivamente às células de refinamento, definidas na seção 3.5. Um exemplo de voxel correspondente a uma zona de refinamento pode ser visto na figura 4.7, extraída a partir da figura 4.6d.



Figura 4.7 – Nessa imagem, a região 5 é uma zona de refinamento, pois os voxels 2 e 6, que são de bordo, não contêm arestas em comum, mas estão ligadas por um vértice que incide na região 5. As demais regiões não são zonas de refinamento.

A classificação pode ser realizada através do algoritmo apresentado na forma do pseudocódigo abaixo:

<p>Algoritmo 4.2 Procedimento de detecção de zonas de refinamento.</p> <p>Requer: Σ' {Parâmetro informado anteriormente}</p> <p>para <i>voxel</i>(x,y,z) representando um voxel de Σ' faça {Codifica a adjacência de todos os 26 vizinhos do voxel em CódigoDeVizinhança Cada um de seus 26 bits indica a presença ou não de um vizinho.} <i>CódigoDeVizinhança</i> $\leftarrow 0$; {um inteiro de 26 bits.} {Varre todos os 26 vizinhos do voxel que está sendo avaliado.} para vizinho representando um voxel vizinhos de voxel (x,y,z) faça se <i>vizinho.tipo</i> = <i>C_BORDO</i> então {O bitMask indica qual dos 26 bits do CódigoDeVizinhança codifica a presença ou não desse vizinho} <i>CódigoDeVizinhança</i> \leftarrow <i>CódigoDeVizinhança</i> ou (<i>vizinho.bitMask</i>); fim se; fim para; {Verifica se o <i>CódigoDeVizinhança</i> confere com algum padrão de vizinhos conectados por vértices ou arestas, mas sem conexão de faces.} para <i>padrão</i> representando cada combinação de voxels vizinhos codificados em uma tabela faça se (<i>CódigoDeVizinhança</i> & <i>TabelaDeZonasDeRefinamento</i>[<i>padrão</i>]) = <i>TabelaDeZonasDeRefinamento</i>[<i>padrão</i>] então</p>

```

voxel.tipo ← C_REFINAMENTO
interromper para;
fim se;
fim para;
fim para;

```

4.2.2 – Reamostragem da região volumétrica e classificação de vértices.

Após a classificação dos voxels do volume, todos os voxels marcados como *bordo* ou *refinamento* serão subdivididos em 8 novos voxels, conforme ilustrado por um exemplo bidimensional na figura 4.8. Em seguida, cada novo voxel, após a subdivisão, será visto como uma região cúbica, formada por 8 vértices, que serão rotulados inicialmente como internos ou externos ao volume, de acordo com o seguinte critério:

- a) Internas: posições centrais na subdivisão de cada célula de bordo original e as posições centrais nas adjacências (vértices, arestas ou faces) entre duas células de bordo.
- b) Externas: demais posições

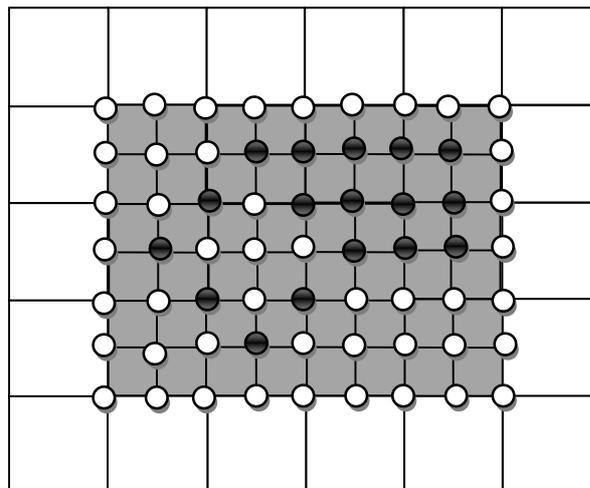


Figura 4.8 – Rotulação inicial dos vértices. As posições internas estão desenhadas como círculos pretos e as externas como círculos brancos.

No passo do operador de dilatação, os vértices à distância unitária de cada vértice rotulado como interno serão rotulados também como internos, independentemente de sua rotulação inicial, fazendo com que as amostras internas se expandam nas quatro direções principais no caso bidimensional do exemplo e nas seis direções principais do reticulado no caso tridimensional (Figura 4.9). É este passo que define a reamostragem que corrige a topologia dos vértices e arestas críticas.

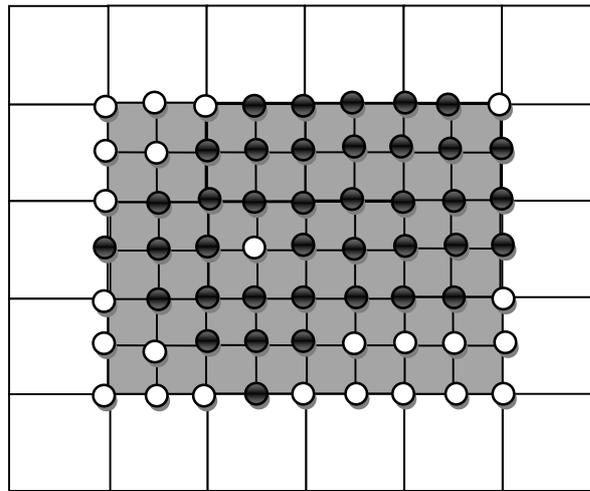


Figura 4.9 – Dilatação das posições interiores gerando a rotulação final.

É possível apresentar um argumento geométrico que explica como a reamostragem proposta neste trabalho elimina os vértices e arestas críticas, sem danificar a topologia das regiões topologicamente corretas do complexo.

O argumento apresentado é válido para o caso bidimensional, existindo a conjectura de que possa ser generalizado para o caso tridimensional.

Considere os vértices da subdivisão rotulados como internos na etapa de refinamento. Esses vértices serão os centroides de células em um reticulado dual rotacionado de 45 graus, conforme ilustrado na figura 4.10.

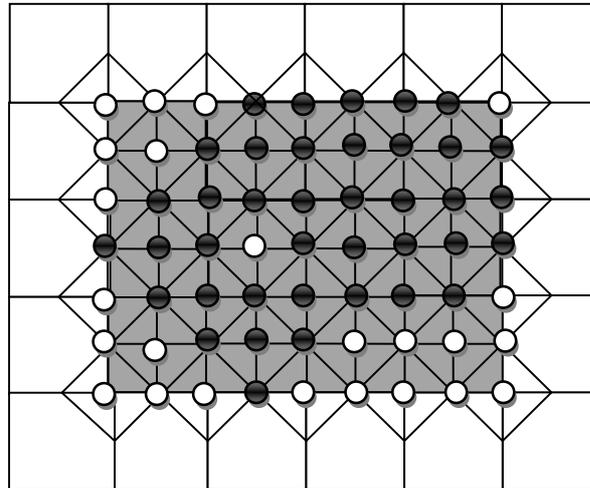


Figura 4.10 – Reticulado dual rotacionado de 45. As células têm como centro os vértices criados na etapa de refinamento antes da dilatação. Os demais vértices formam os pontos de interseção do reticulado.

Considere uma triangulação do reticulado dual rotacionado, conforme apresentado na figura 4.10. Todo triângulo determinado por três vértices internos será considerado um triângulo interno à região que determina o objeto. Se um triângulo contiver pelo menos um vértice rotulado como externo, então ele será considerado um triângulo do exterior.

Primeiramente é analisado o caso de células do bordo que não contêm vértices críticos, como na região selecionada na figura 4.11. Nessas sub-regiões, cada célula original dá origem a um conjunto de triângulos internos. Cada célula isoladamente induz no reticulado dual rotacionado quatro triângulos internos incidentes ao vértice central.

Quando duas células são conectadas por uma aresta, o vértice criado na etapa de refinamento, juntamente com sua dilatação, gera na vizinhança da adjacência um conjunto de 9 vértices internos, em um reticulado 3×3 , definindo uma região formada por 8 triângulos internos (região tracejada na figura 4.11), garantindo a geração de um subcomplexo 1-conectado, localmente homeomorfo a um disco (ou semi-disco), satisfazendo a definição de variedade.

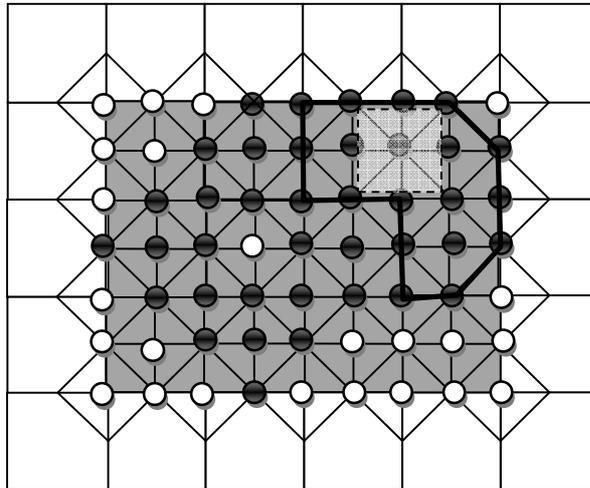


Figura 4.11 – Células geradas na região sem vértices críticos

Considera-se agora o caso de células com adjacência definida por um vértice crítico. Novamente, cada célula isoladamente contém pelo menos 4 triângulos internos, incidentes aos centroides das células.

No vértice crítico, o processo de dilatação induz 4 triângulos internos (regiões tracejadas da figura 4.12) que são adjacentes aos triângulos internos das duas células originais, também garantindo o surgimento de um sub-complexo 1-conectado, localmente homeomorfo ao disco unitário no plano.

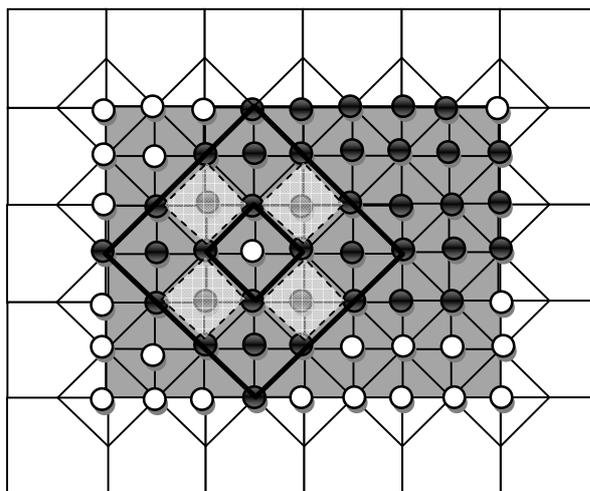


Figura 4.12 – Células geradas nas regiões com vértices críticos.

Conjectura-se que o mesmo argumento possa ser feito para o caso tridimensional, considerando um reticulado dual formado por células octaédricas subdivididas em tetraedros. Similarmente, um tetraedro será considerado interno se todos os seus vértices forem rotulados como internos e externo, caso contrário.

4.2.3 – Extração da malha

A extração da malha é feita em duas etapas: num primeiro passo é feita a geração das células afins (poliedros) que definem o novo complexo celular $\bar{\Sigma}$ associado ao volume intermediário \bar{V} ; na segunda etapa, são determinadas, dentre o conjunto de faces das células afins, aquelas que de fato compõem a superfície que delimita o volume, representada através de uma malha associada ao bordo $\partial\bar{\Sigma}$ de $\bar{\Sigma}$.

Geração das células afins

Cada novo voxel gerado pelo processo de subdivisão será considerado como uma região que dará origem a uma célula afim, isto é, um poliedro (figura 4.13). Cada região cúbica associada a um voxel receberá um número de 8 bits, que descreve a presença ou não de cada um de seus vértices no interior do volume. Esse número será utilizado para consultar uma *Lookup Table* que determinará a configuração da célula afim no interior da região cúbica.

A *Lookup Table*, descrita no Apêndice A, apresenta apenas 67 configurações válidas que são resumidas em 6 casos diferentes e suas respectivas rotações. Não se trata de um subconjunto da *Lookup Table* do *Marching Cubes*. Os casos, ilustrados na figura 4.13, consistem em: tetraedro (figura 4.13a), pirâmide quadrada (figura 4.13b), prisma triangular (figura 4.13c), união de duas pirâmides quadradas (figura 4.13d), o complementar de um tetraedro em um cubo (figura 4.13e) e o cubo (figura 4.13f).

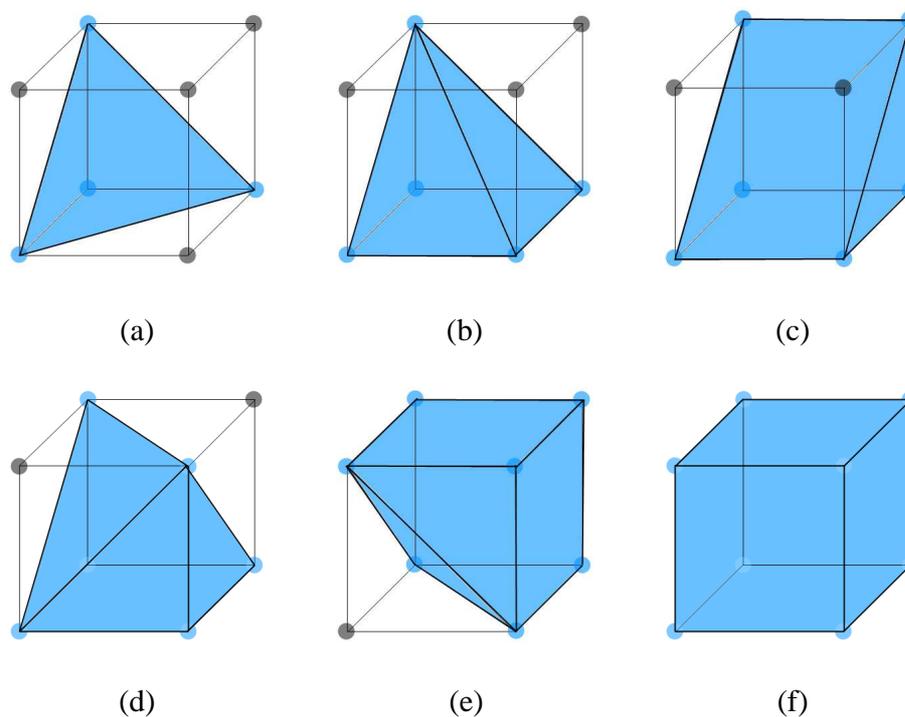


Figura 4.13– Os seis casos da *Lookup Table* da poligonização do método proposto. Vértices em azul fazem parte da superfície, enquanto os em preto estão fora. (a) Tetraedro, com 10 rotações. (b) Pirâmide quadrada, com 24 rotações. (c) Prisma triangular, com 12 rotações. (d) União de duas pirâmides quadradas, com 12 rotações. (e) Dual de tetraedro no cubo, com 8 rotações. (f) Cubo (caso 255 da tabela).

Geração da malha

O conjunto das faces de todas as células afins geradas determina um conjunto de faces candidatas a formarem a malha final que delimita o volume.

Existem duas condições para que uma face candidata faça parte da superfície poligonal resultante. A primeira é que ela não faça parte de uma face do cubo que incide em uma região interna do volume. A segunda condição é que ela só pode ser criada uma única vez. Se a face candidata já existir, ela deverá ser removida, pois faz parte do interior do volume.

Ao gerar cada face, é necessário calcular seus atributos. O cálculo do vetor normal da face é feito a partir do produto vetorial entre duas arestas no sentido anti-horário. A cor de cada face é calculada a partir da média ponderada da cor dos voxels que são vizinhos à parte da região que está sendo poligonizada. Utilizam-se pesos que variam de acordo com o tipo de adjacência do

voxel vizinho àquele que contém a face da malha. Durante a implementação dessa média, definiu-se peso 8 para a cor do voxel em que a região pertence, peso 4 para os vizinhos de face, 2 para os de aresta e 1 para os de vértice.

Após a criação de todas as faces, eliminam-se os vértices que o modelo não utiliza, isto é, que não incide em nenhuma das faces do objeto.

4.3 – Resultados

Nesta seção são apresentados resultados de cada uma das etapas do processo proposto para validar seu funcionamento. Por isso, etapas estratégicas do processo serão analisadas separadamente, para que haja a compreensão de quais problemas que podem ocorrer e como evitá-los.

4.3.1 – Classificação dos voxels

A solução adotada para detecção de regiões internas do modelo de voxels não funciona quando o artista modela um objeto cuja fronteira é uma superfície com bordo. O reconhecimento de regiões internas neste caso é um problema em aberto (DIGNE *et al.*, 2012) e não está no escopo deste trabalho. Esses casos podem ser contornados se o artista fechar os buracos usando uma cor transparente ou preta.

A figura 4.14 mostra a unidade Kirov Airship (© Electronic Arts), extraída do jogo Command & Conquer Red Alert 2, com e sem seu volume interno. O Kirov não tem buracos. Em compensação, a unidade Night Hawk, extraída do mesmo jogo na figura 4.15, tem vários buracos por onde saem paraquedistas, por exemplo. A parte interna do helicóptero, quando poligonizada, gerará polígonos inúteis que não serão vistos de fora do veículo. É importante ressaltar que a técnica proposta funciona corretamente em modelos com genus diferente de zero, como pode ser vista na figura 1.3.

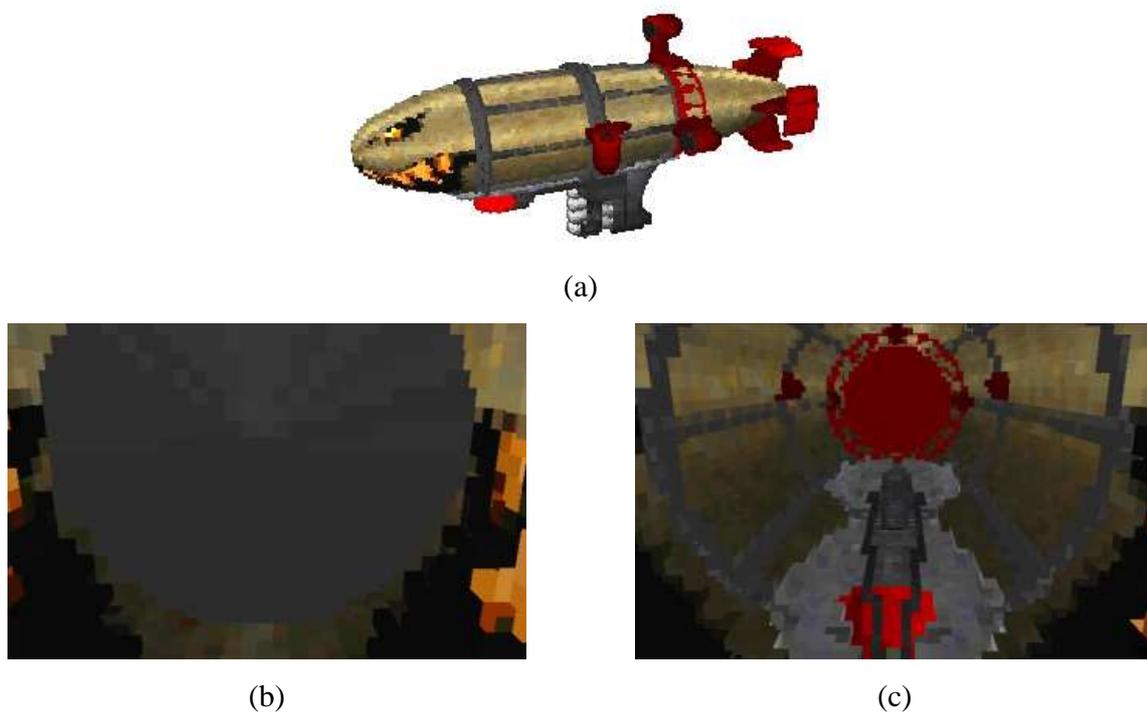


Figura 4.14 – Resultado do mapeamento de regiões internas com modelos sem buracos (Muniz, 2010). (a) Kirov Airship (© Electronic Arts). (b) Volume interno pintado pelo artista. (c) O volume interno removido por meio do mapeamento da seção 4.2.1.

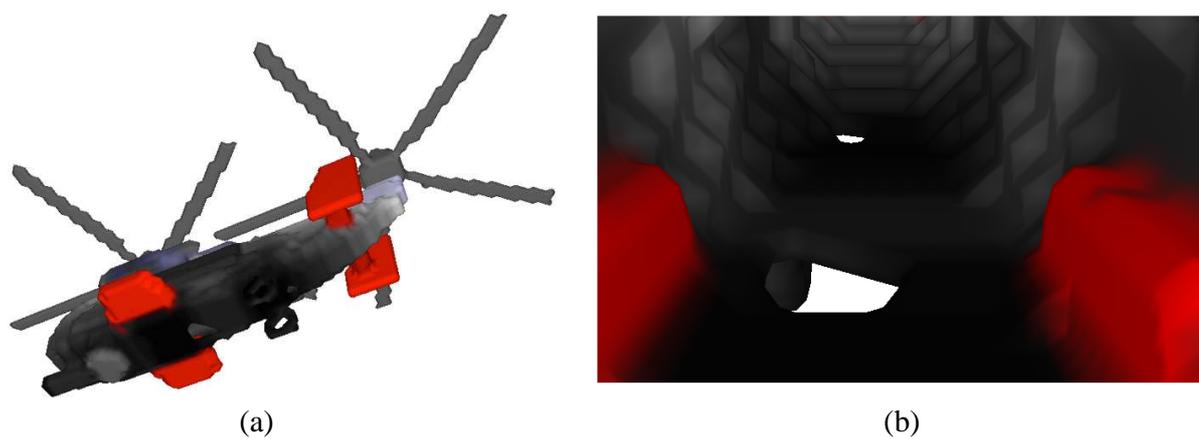
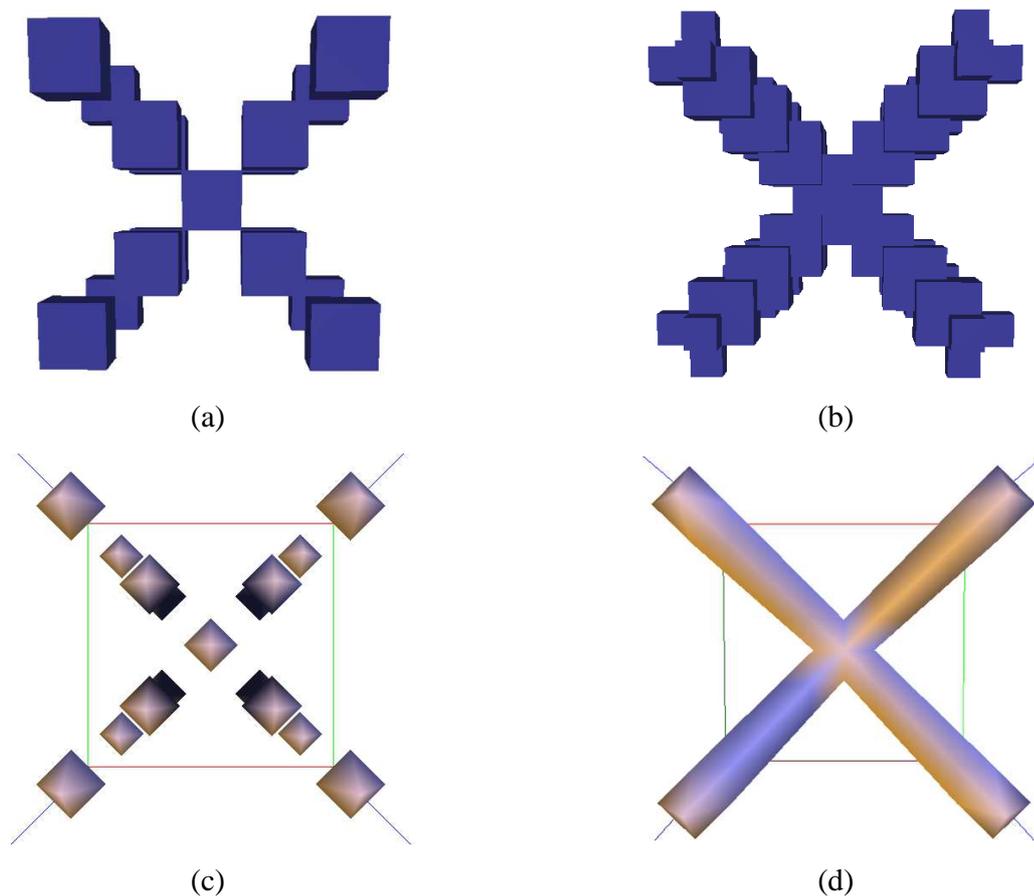


Figura 4.15 – Região interna de modelos com buracos como o Night Hawk (a) (© Electronic Arts) faz parte da superfície externa (b) do modelo gerado.

4.3.2 – Reamostragem da região volumétrica.

Aqui são apresentados alguns resultados do processo de reamostragem e seu impacto na extração da malha tanto pelo método proposto neste trabalho quanto por outros métodos da literatura.

A figura 4.16 ilustra um exemplo de dado com vértices críticos sendo poligonizado através do *Marching Cubes* de Lewiner *et al.* (2003) e Muniz *et al.* (2010), antes e após ser reamostrado. Já a figura 4.17 mostra o resultado do mesmo modelo de voxels com a técnica proposta nesta dissertação antes e após a suavização de malhas.



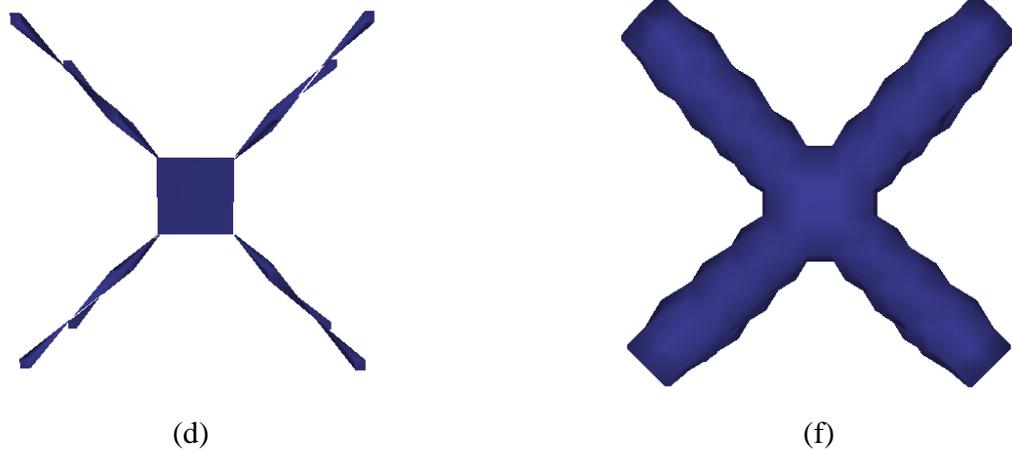


Figura 4.16 – Exemplo de poligonização que têm células que compartilham vértices. (a) Modelo original representado por cubos. (b) Modelo reamostrado representado por cubos. (c) Modelo original poligonizado por Lewiner et al. (2003). (d) Modelo reamostrado poligonizado por Lewiner et al. (2003) . (e) Modelo original poligonizado por Muniz et al. (2010). (f) Modelo reamostrado poligonizado por Muniz et al. (2010).

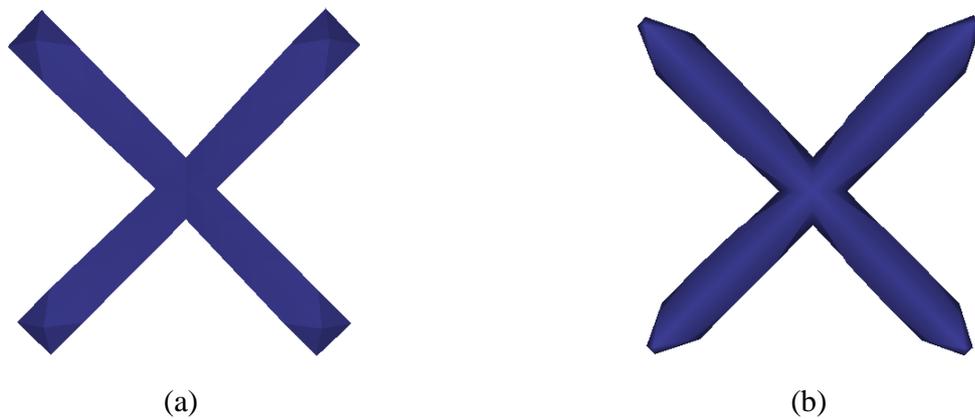


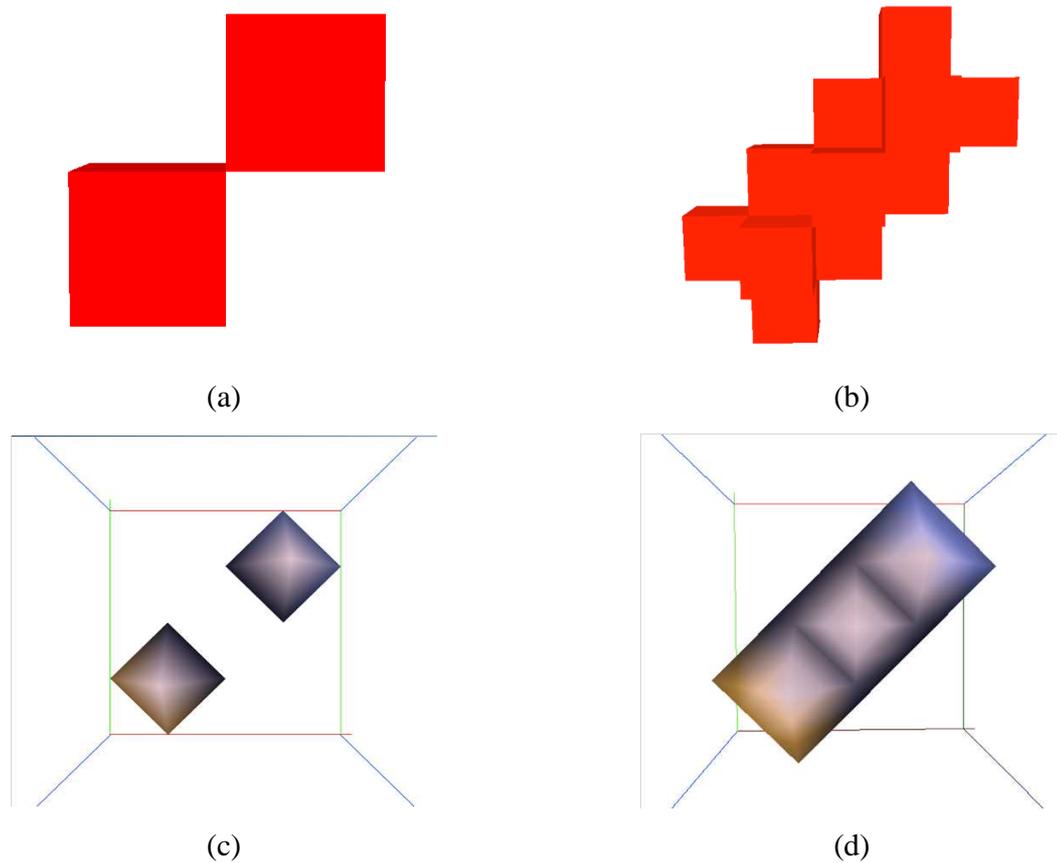
Figura 4.17 – Mesmo modelo da figura 4.16 poligonizado pela técnica proposta nesta dissertação. (a) Antes de suavizar as normais. (b) Depois de suavizar a malha.

Diferentemente de métodos baseados no *Marching Cubes*, a *Lookup Table* do método de poligonização proposto neste trabalho produz triângulos que passam pelos vértices do cubo classificados como internos (veja figura 4.13). Desta forma, evitam-se os casos ambíguos.

No caso de métodos similares ao *Marching Cubes*, a reamostragem garante que quase todos os casos de ambiguidade, mencionados no capítulo 2, não venham a ocorrer. A exceção encontra-se no caso ilustrado na figura 4.13d, que ocorre em zonas de refinamento vizinhas às células que compartilham 1-faces.

Na versão do *Marching Cubes* apresentada por Lewiner et al. (2003), que é capaz de lidar com ambiguidades, a ausência da reamostragem leva a geração de um resultado cuja geometria não é esperada (criação de duas componentes desconectadas, como pode ser visto na figura 4.18(c)). Após a reamostragem, o mesmo método produz um objeto com apenas uma componente conforme o esperado, mas ainda apresenta oscilações indesejadas na superfície.

No método proposto, o resultado é apropriado tanto do ponto de vista topológico quanto geométrico, como pode ser visto nas figuras 4.18(g) e 4.18(h).



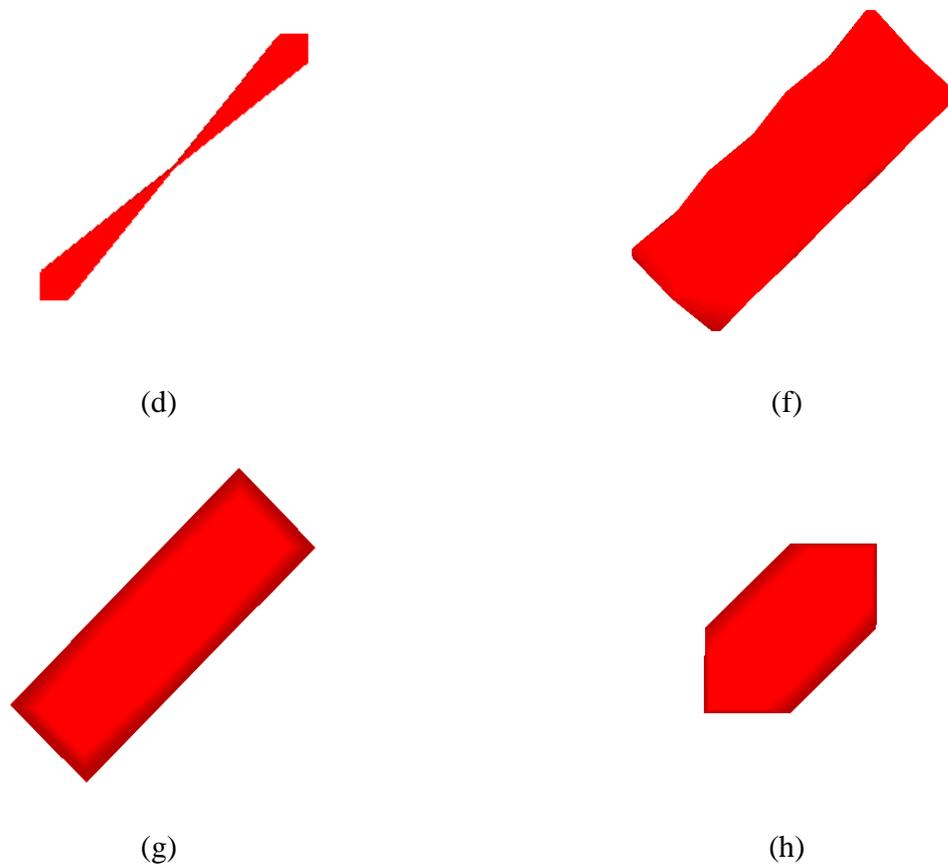


Figura 4.18 – Exemplo de poligonização de um modelo que têm células que compartilham arestas. (a) Modelo original representado por cubos. (b) Modelo reamostrado representado por cubos. (c) Modelo original poligonizado por Lewiner *et al.* (2003). (d) Modelo reamostrado poligonizado por Lewiner *et al.* (2003). (e) Modelo original poligonizado por Muniz *et al.* (2010). (f) Modelo reamostrado poligonizado por Muniz *et al.* (2010). (g) Modelo extraído pela técnica proposta antes de suavizar a malha. (h) Modelo extraído pela técnica proposta após a suavização de malha.

4.3.3 – Poligonização

Nesta seção avaliamos a quantidade e a qualidade dos triângulos gerados pelo método proposto nessa dissertação.

Apesar de as malhas de triângulos serem mais densas do que as obtidas com *Marching Cubes*, considerando o processo de refinamento, elas são mais bem-comportadas. Foram utilizadas três métricas para avaliar a qualidade da malha gerada, nos modelos GDI Nemesis, Kirov Airship MKII, Mad Dog e War Miner na tabela 4.1.

Tabela 4.1 – Qualidade e complexidade da malha pelo método proposto.

Modelo	Vértices	Triângulos	Aspect Ratio	Skewness	Smoothness	Razão de triângulos por voxel pintado
War Miner	28 004	64 858	1,482	0,234	0,007	10,112
Kirov Airship MKII	75 838	169 430	1,492	0,229	0,007	5,732
Mad Dog	93 983	205 676	1,478	0,222	0,007	8,310
GDI Nemesis	322 819	710 384	1,477	0,231	0,005	8,999

A métrica *Aspect Ratio* mede a razão entre a maior e a menor aresta de cada triângulo da malha. Se os valores obtidos fossem próximos de 1, isso faria com que a malha tivesse triângulos predominantemente equiláteros.

Skewness é uma medida de assimetria da distribuição dos triângulos da malha. Dado A_T a área de um triângulo T da malha e A_{CT} a área de um triângulo equilátero inscrito no círculo que circunscreve T , *Skewness* é a razão entre $(A_{CT} - A_T)$ por A_{CT} . Valores próximos de zero correspondem a uma malha mais simétrica.

Smoothness mede o módulo da diferença entre a área do triângulo e seus vizinhos imediatos. Quanto mais próximo de zero, mais suave é a malha.

A distribuição dos vértices faz com que o espaçamento entre eles seja uniforme. O tamanho das arestas é sempre 1, $\sqrt{2}$ ou $\sqrt{3}$. O ângulo entre as arestas é sempre múltiplo de 45° . O comportamento do *Aspect Ratio* no modelo Nemesis, mostrado na figura 4.19, é semelhante em todos os outros modelos.

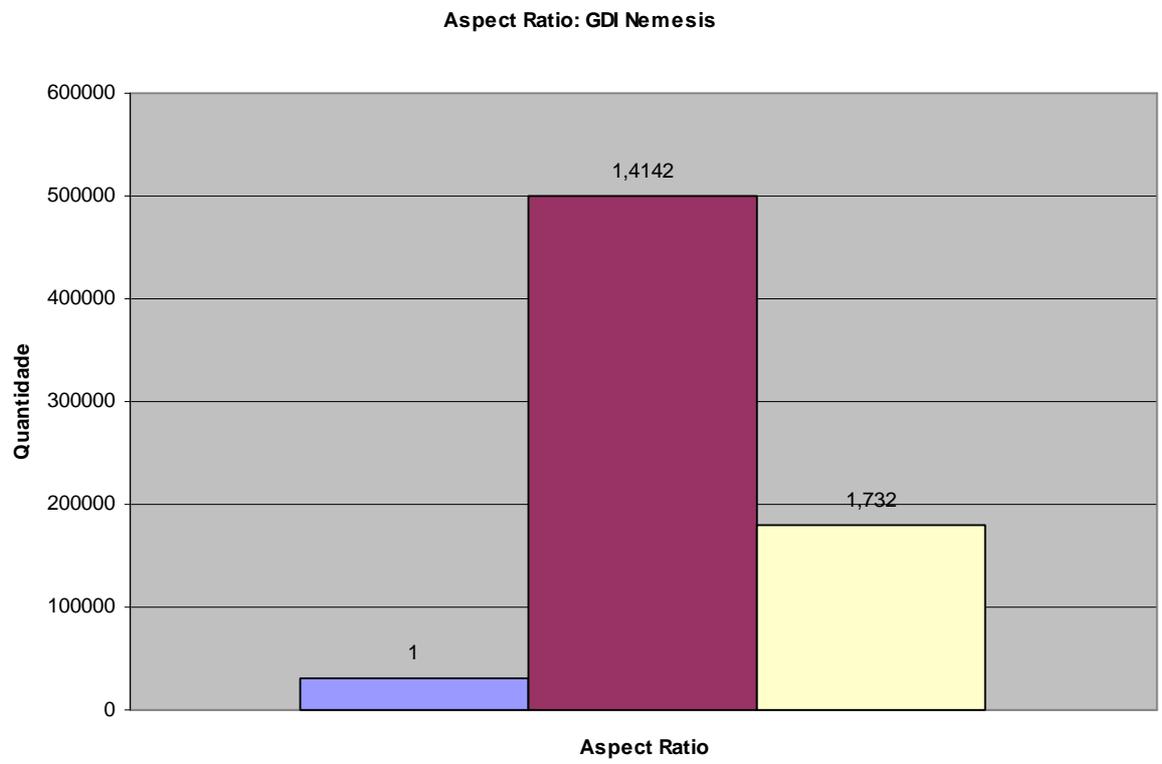


Figura 4.19 – Histograma do Aspect Ratio do modelo GDI Nemesis.

Capítulo 5 – Pós-Processamento

Ao término da extração da malha, as funções de atributo são extraídas dos voxels para as faces da malha (figura 5.1a) e obtidas posteriormente para cada vértice (figuras 5.1b e 5.1c). O cálculo de normais e cores será apresentado na seção 5.1. Por fim, o aspecto facetado da malha é corrigido através de um processo de suavização que será descrito na seção 5.2. O fluxograma na figura 5.2 descreve as etapas do pós-processamento. A seção 5.3 analisará os resultados das etapas descritas nas seções 5.1 e 5.2.

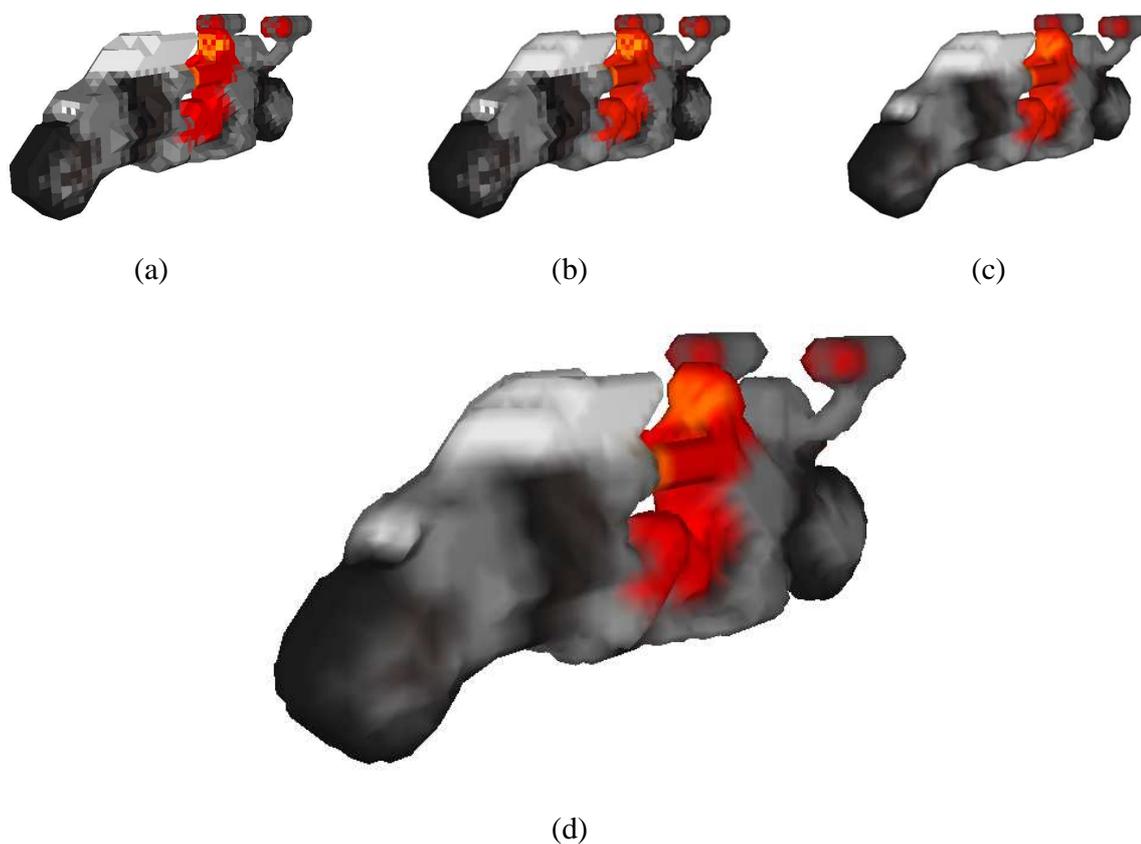


Figura 5.1 – Etapas da metodologia proposta. Modelo extraído de ROTD (2012). (a) Modelo obtido após a extração da malha poligonal, com função de atributos para cada face. (b) Atribuição de normais para cada vértice. (c) Atribuição de cores para cada vértice. (d) Suavização de malhas.

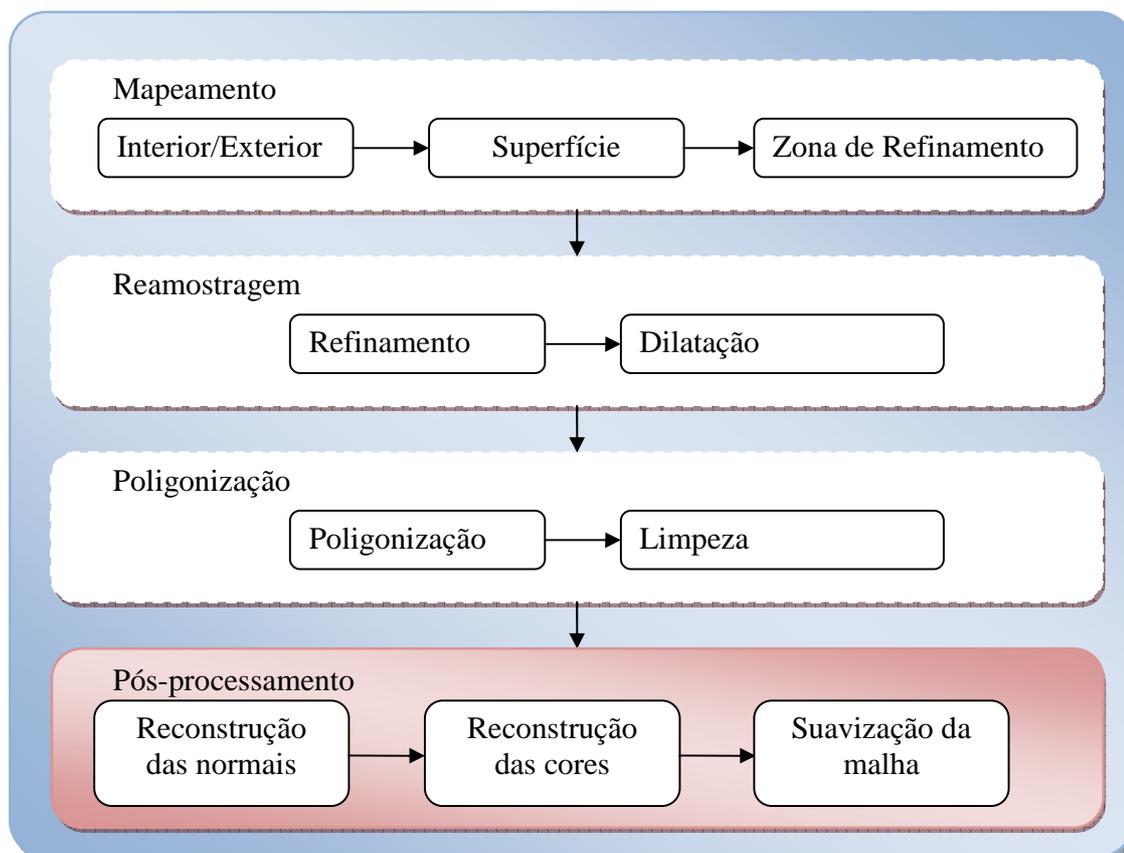


Figura 5.2 – Fluxograma do método proposto. Este capítulo explica as etapas de pós-processamento.

5.1 – Suavização de atributos

Depois de obter a superfície poligonizada, com cores e vetores normais para cada face, deve-se calcular o valor desses atributos para cada vértice. O procedimento aplicado para ambos os casos coincide com o método publicado em Muniz *et al.* (2010).

Para calcular o vetor normal de cada vértice, faz-se uma varredura nas faces, e soma-se o vetor normal de cada uma. Nesta soma, consideram-se apenas os vetores normais de superfícies diferentes que incidem no vértice. Caso duas ou mais faces tenham o mesmo vetor normal, ele é somado apenas uma vez, pois se subentende que essas faces fazem parte de uma única superfície. No final, normaliza-se o vetor normal.

Para gerar cores por vértices, faz-se uma varredura nas faces e somam-se os valores nos componentes vermelho, verde e azul de cada uma em seus vértices. No final, divide-se o resultado de cada vértice pelo número de vezes em que ele foi somado para cada atributo.

5.2 – Suavização da malha

Nesta dissertação, o objetivo das operações de filtragem de malhas poligonais é suavizá-las. O algoritmo proposto desloca a posição dos vértices, deformando a malha de tal forma que ela fique mais suave.

A frequência da posição de um vértice v , em um objeto gráfico, está relacionada à curvatura da superfície na região em torno do vértice v . Por sua vez, a definição de curvatura para superfícies representadas por malhas é uma tarefa complexa, visto que envolve a noção de diferenciabilidade de segunda ordem, que não pode ser diretamente definida em malhas poligonais, que são representações lineares por partes.

Na literatura, uma das formas mais utilizadas para representação de malhas no domínio espectral e para estimar suas frequências, está fundamentada no uso de operadores Laplacianos discretos. De acordo com Taubin (1995), a base de Fourier é uma autofunção do operador Laplaciano Beltrami, o que justifica seu uso para estimar frequências e sua aplicação em diferentes problemas de processamento geométrico como, por exemplo, suavização de malhas poligonais, dentre outras.

Existem duas formas tradicionais de se aproximar o Laplaciano: o operador *Laplaciano Uniforme*, proposto por Taubin (1995), que dá um peso igual a todas as arestas que incidem no vértice e o *Laplaciano das cotangentes* Meyer *et al.* (2003) que usa pesos que dependem da geometria da vizinhança do vértice.

O problema de suavização de malhas pode ser resolvido através de uma formulação baseada na equação da difusão (BOTTSCH *et al.*, 2010).

$$\frac{\partial f(x, t)}{\partial t} = \lambda \Delta f(x, t) \quad (5.1)$$

Essa equação descreve a evolução da função f ao longo do tempo t em função do Laplaciano $\Delta f(x, t)$ multiplicado por um coeficiente λ .

O método proposto nesse trabalho não soluciona o problema de suavização com base na equação de difusão. Entretanto, tal equação inspirou a proposta de um processo de evolução de superfícies representadas por malhas, que desloca os seus vértices iterativamente com base em dois parâmetros: direção e intensidade.

A direção do deslocamento de um vértice é o seu vetor normal \vec{n} . Para suavizar a malha, o deslocamento será no sentido que diminui o tamanho das arestas que incidem no vértice, o que tornará a geometria da região mais suave.

Para determinar esse sentido, deve-se detectar se a superfície em torno do vértice é côncava ou convexa. Isso é feito por meio do cálculo do produto escalar entre a direção de cada aresta e a direção do vetor normal. Se o produto escalar de alguma aresta tiver resultado menor que 0, então a região será côncava. Caso contrário, será convexa. A figura 5.3 ilustra a intuição para determinar se uma região em torno de um vértice é côncava ou convexa. Se a região for côncava, então o sentido do deslocamento será o sentido oposto do vetor normal. Caso contrário, o deslocamento é feito no sentido do vetor normal do vértice.

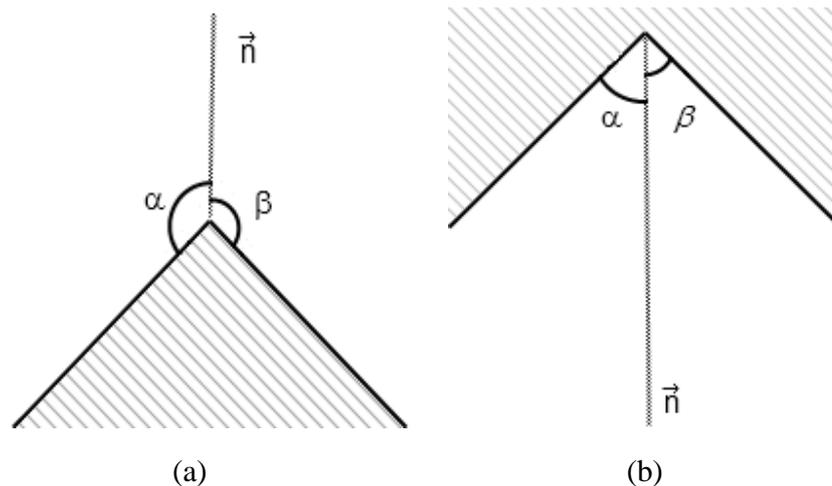


Figura 5.3 – Intuição para detecção de regiões côncavas e convexas na superfície. Seja α e β ângulos de arestas com o vetor normal \vec{n} . (a) Nas regiões côncavas, o módulo de pelo menos um dos ângulos entre arestas e normais é maior do que 90° . (b) Nas regiões convexas, o módulo de todos os ângulos entre arestas incidentes ao vértice com a normal é menor do que 90° .

A intensidade do deslocamento é determinada a partir da estimativa da curvatura da região analisada e um fator de escala λ , que será o tamanho da deformação máxima que pode ser feita, sem transformar uma região côncava em convexa, ou vice-versa.

Para determinar a escala λ considera-se um volume centralizado em torno do vértice v que será deslocado (figura 5.4). A base desta região é dada pelas faces na estrela de v ,

projetadas sobre o plano π , paralelo ao plano tangente e que passa por um vértice v_λ , onde v_λ é dado por:

$$v_\lambda = v + \min \| \langle v_i - v, \vec{n} \rangle \| \vec{n} \quad (5.2)$$

onde v_i são os vértices na vizinhança-im $N_1(v)$. A escala adotada para o deslocamento λ é dada por $\lambda = \|v_\lambda - v\|$. É importante destacar que λ é um parâmetro adaptativo que varia localmente, conforme as características da malha em torno de um vértice v .

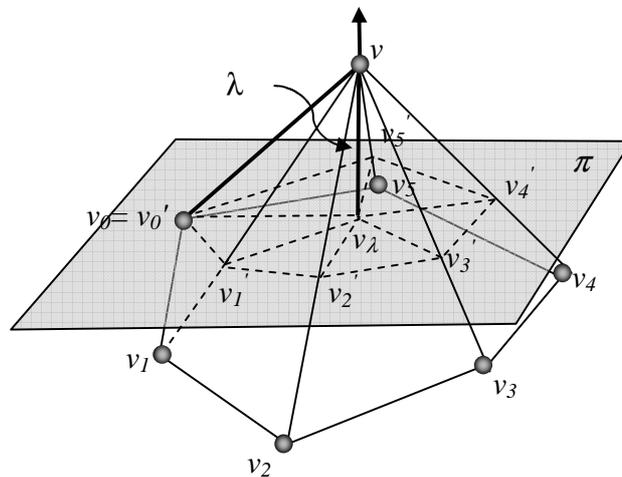


Figura 5.4 – Região analisada em torno do vértice v . Neste exemplo, v_λ, v'_i , onde $0 \leq i \leq 5$, são pontos que estão no plano paralelo ao plano tangente passando v_λ , que é a base da região amostrada. A escala λ é a distância de v_λ para v .

A estimativa da curvatura é dada por uma medida de variação das normais das faces em relação à normal do vértice, de cuja estrela elas fazem parte.

Mais precisamente, o cálculo da estimativa da curvatura de um vértice v é feito por meio de um somatório dos valores absolutos dos complementos do cosseno dos ângulos entre \vec{n} e a normal \vec{n}_i de cada face f_i , ponderados pelo ângulo entre as arestas de f_i na estrela de v . Esta operação está ilustrada na figura 5.5.

$$\bar{\kappa} \approx \frac{1}{2\pi} \sum_{i, f_i \in N_1(v_\lambda)} \alpha_i |1 - \langle \vec{n}, \vec{n}_i \rangle|, \quad (5.3)$$

$$\alpha_i = \angle(e_i, e_j)$$

onde $e_i, e_j \in f_i$ e $e_i, e_j \in \text{star}(v_\lambda)$

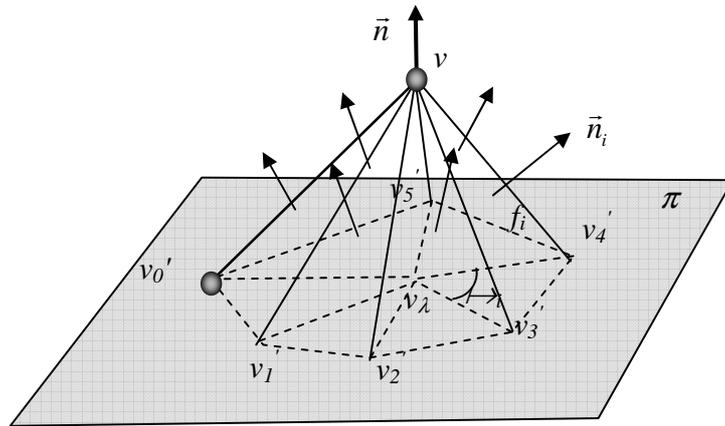


Figura 5.5 Estimativa da curvatura em v . Nesta figura n_i , são as normais nas faces f_i e α_i é o ângulo das arestas de f_i que incidem em v_λ .

A ponderação por α_i tem como objetivo fazer com que a estimativa da curvatura dê menos importância aos triângulos estreitos que incidem em v do que triângulos com maior ângulo no ponto de incidência.

A estimativa da curvatura aqui proposta gera um valor que não depende diretamente da escolha de uma região em torno de um vértice v , diferentemente da fórmula do Laplaciano das cotangentes. Por outro lado, por considerar uma medida angular, não sofre dos mesmos problemas que o Laplaciano Uniforme, cujos resultados para malhas não uniformes pode ser indesejado.

Pode-se afirmar então que o processo de evolução da superfície aqui proposto é regido pelo processo iterativo.

$$x_i \leftarrow x_i + \lambda(v_i) \bar{\kappa}(v_i) \vec{n}, \quad (5.4)$$

onde x_i , são as coordenadas de um vértice v_i da malha.

O pseudocódigo a seguir explica como a curvatura é estimada e utilizada, com a escala e a direção de deslocamento do vértice para encontrar a nova posição do vértice.

Algoritmo 5.1 Procedimento de cálculo da nova posição do vértice.

Requer: $G = (V, E) \in \partial \bar{\Sigma}$ { Grafo G com V vértices, E arestas }

Requer: N { Vetores normais dos vértices de V }

Requer: λ { Parâmetro informado anteriormente }

Requer: *DireçãoDoDeslocamento* { Direção do vetor normal de cada vértice ou exatamente o oposto, se a região for côncava }

para v representando um vértice do conjunto V e n representando o vetor normal de v **faça**

CurvaturaEstimada \leftarrow 0;

para f representando *todas as faces incidentes em v* e fn representando o vetor normal de f **faça**

// Projeta as duas arestas da face originárias no vértice no plano paralelo ao plano tangente e

// calcula o ângulo entre elas.

Aresta1 \leftarrow *normalizar*(f.Proximo (v) – v);

Aresta1 \leftarrow *normalizar*(*Aresta1* – (n * dot(n, *Aresta1*)));

Aresta2 \leftarrow *normalizar*(f.Anterior (v) – v);

Aresta2 \leftarrow *normalizar*(*Aresta2* – (n * dot(n, *Aresta2*)));

ÂnguloNoPlano \leftarrow *arccos*(dot(*Aresta1*, *Aresta2*));

// Calcula o módulo do produto escalar entre a normal do vértice com a normal da face.

AnguloNaFace \leftarrow *abs*(dot(n, fn));

CurvaturaEstimada \leftarrow *CurvaturaEstimada* + [(1 – *AnguloNaFace*) * *ÂnguloNoPlano*];

fim para;

// Calcula o deslocamento final

v.*Posição* \leftarrow v.*Posição* + ((1 / 2 π) * λ * *CurvaturaEstimada* * *DireçãoDoDeslocamento*);

fim para;

5.3 – Análise de resultados

Esta seção descreve os resultados obtidos nas etapas de suavização de cores, normais e malha, mostrando se estão adequados com o que se espera de cada etapa e comparando com outras técnicas.

5.3.1 – Suavização de atributos

O objetivo da suavização de normais é gerar a direção da orientação dos vértices para obter superfícies com aparência suavizada. Já a suavização de cores detecta a cor de cada vértice para obter superfícies com aparência suavizada. Ambas são feitas através de interpolação linear e, por isso, dão uma aparência mais suave ao modelo. O resultado dessa etapa com o War Miner pode ser visto na figura 5.6.

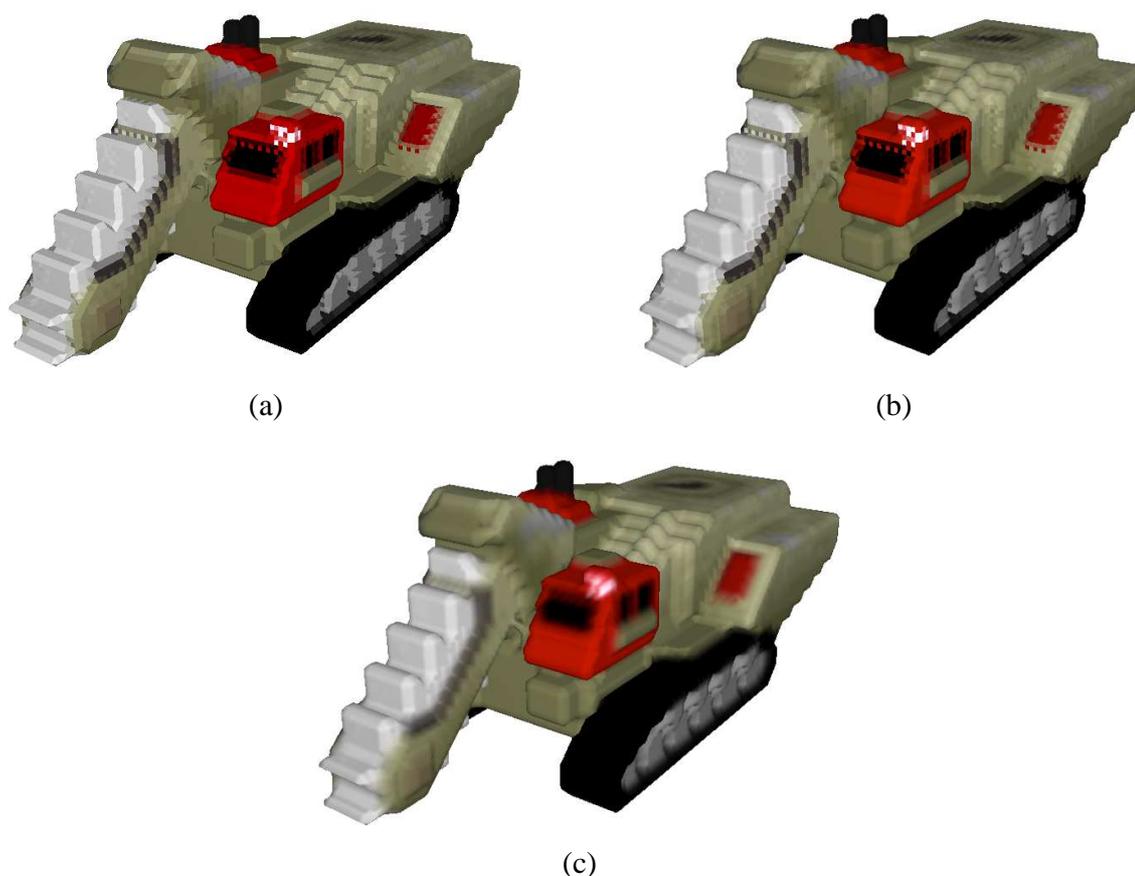


Figura 5.6 – Suavização de atributos no War Miner. (a) Modelo após extração da superfície. (b) Modelo após suavização das normais. (c) Modelo após a suavização de cores.

5.3.2 – Suavização da malha

O objetivo da suavização de malha neste trabalho é alterar as posições dos vértices para obter superfícies mais suaves, pois, até então, todos os ângulos entre arestas eram múltiplos de 45° . Deve-se respeitar o tipo de região, côncava ou convexa, em torno de cada vértice, sem alterar esse comportamento.

A abordagem aqui proposta é diferente da suavização de malhas feita no trabalho de Muniz *et al.* (2010), em que se buscava reconstruir o formato da superfície usando um filtro de passa-banda (Lanczos) composto com um operador *Laplaciano Uniforme*.

A suavização de malhas de Muniz *et al.* (2010) é calibrada para usar um valor de escala que é incompatível com a resolução dos modelos gerados nesta pesquisa, podendo resultar em vértices e arestas não variedade devido ao colapso deles em algumas regiões. A figura 5.7 ilustra o resultado do método de suavização de malhas proposto neste trabalho e um caso em que a suavização de malhas de Muniz *et al.* (2010) falha.

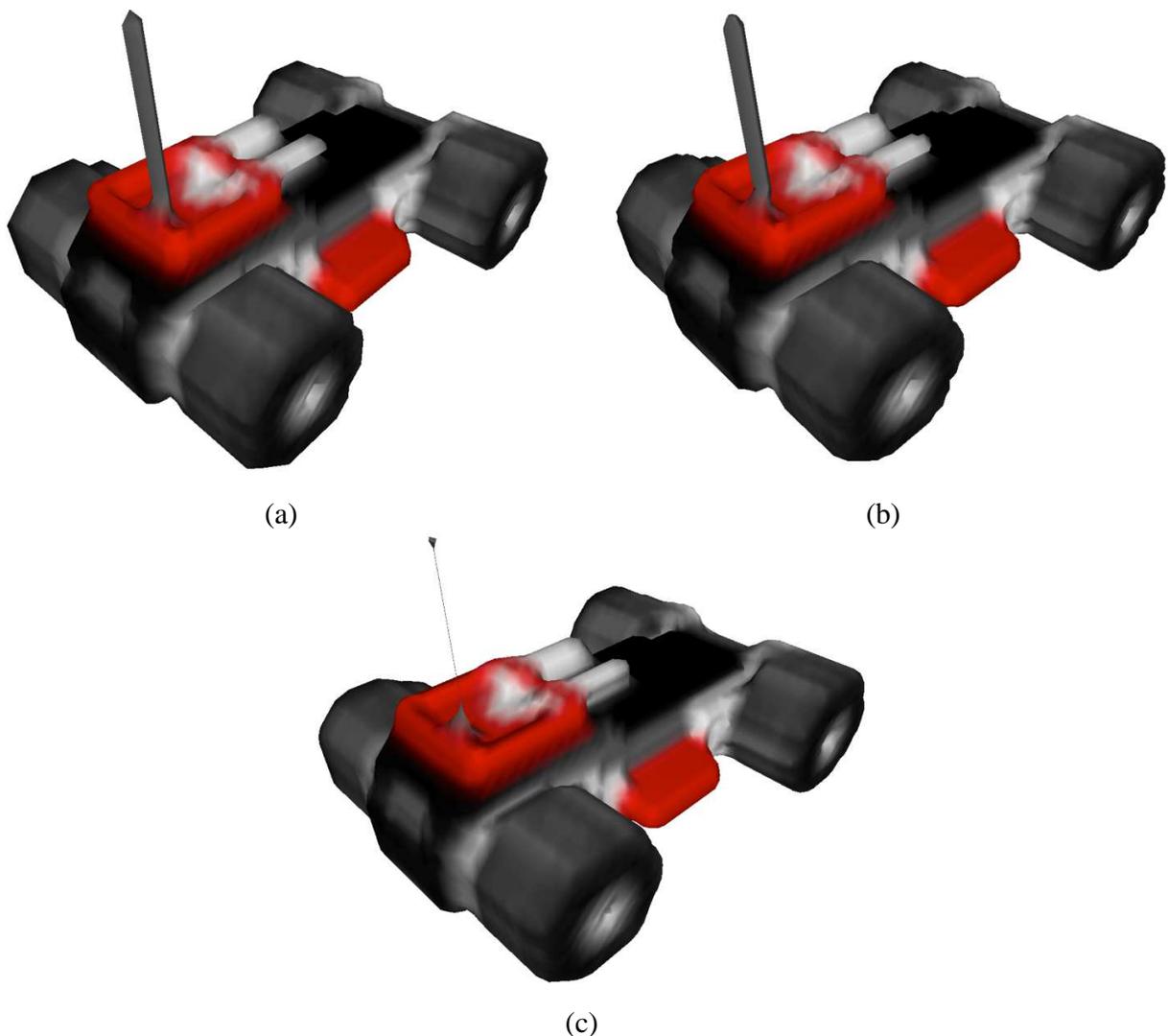


Figura 5.7 – Suavização de malha no Nod Buggy (© Electronic Arts). (a) Modelo obtido da etapa de extração da superfície de bordo. (b) Suavização da malha com o método proposto na seção 5.2. (c) Suavização da malha de (a) com o método de Muniz *et al.* (2010) gera problemas na antena traseira do bugre, com várias arestas se colapsando.

A malha obtida nesse processo continua tendo um bom *Aspect Ratio*, não havendo a necessidade de remalhamento, como o que é feito em outras técnicas como o do Bischoff *et al.* (2005).

Também é importante ressaltar que com uma única iteração, o método é conservador demais para que ocorra uma distorção significativa das normais. Portanto, um novo cálculo das normais é opcional e, dificilmente o usuário perceberá a diferença. Caso haja a intenção de reaplicar essa suavização de malhas, é recomendável recalcular as normais de cada face e vértice já que esse procedimento é altamente dependente desses dados.

Capítulo 6 – Resultados

Neste capítulo serão mostrados experimentos feitos com cada etapa do processo avaliando seu desempenho e eficácia. Para compreensão desses testes, o ambiente de experimentação e métricas de qualidade são descritos com detalhes.

6.1 – Ambiente de experimentação

Os experimentos foram realizados através do software Voxel Section Editor III (VXLSE3, 2012), que também é conhecido como VXLSE III. Este editor de modelos em *voxels* gratuito foi criado para editar unidades dos jogos Command & Conquer Tiberian Sun (figura 6.1a) e Command & Conquer Red Alert 2 (figura 6.1b), que usam um formato de dados volumétrico proprietário criado pela Westwood Studios, atualmente pertencente à Electronic Arts.



Figura 6.1 – Imagens dos jogos (© Electronic Arts): (a) Command & Conquer Tiberian Sun. (b) Command & Conquer: Red Alert 2

Desde 2009, esse programa (figura 6.2) vem sendo utilizado para o estudo de formas de converter seus modelos volumétricos em malhas poligonais, sendo algumas delas publicadas em trabalhos como Muniz (2010) e Muniz *et al.* (2010), além desta dissertação.

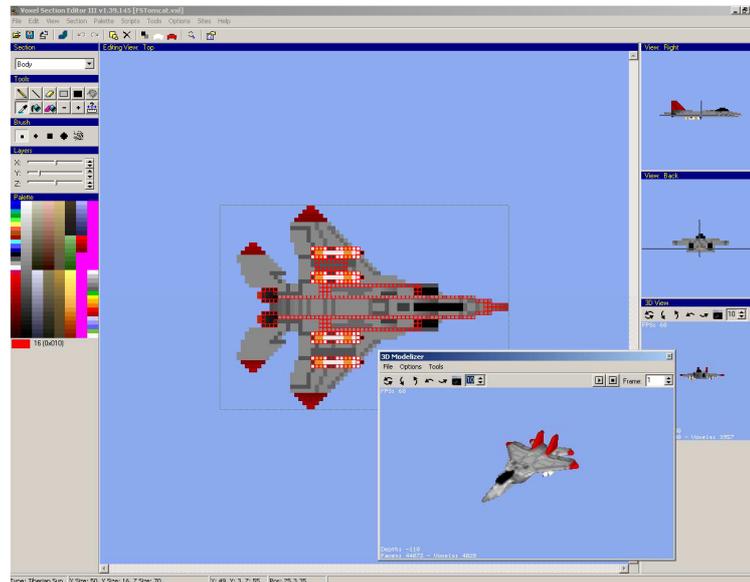


Figura 6.2 – Editor de modelos volumétricos Voxel Section Editor III. A versão 1.39.145 foi utilizada nos testes.

Os modelos volumétricos de Stanford (2012) são incompatíveis com o Voxel Section Editor III, pois, além de não ler o formato utilizado por Stanford, o programa tem limitações de tamanho do volume, além de paleta de cores e normais provenientes do formato proprietário do modelo de voxels da Westwood Studios.

Por esse motivo, foram utilizados alguns modelos incluindo: modelos extraídos dos jogos da franquia Command & Conquer a partir de programas de terceiros gratuitos, como o XCC Utilities (XCC, 2012); modelos criados por usuários que modificam esses jogos e disponibilizam seus modelos em sites como Project Perfect Mod (PPMVOXELS, 2012), YR Argentina (YRARG, 2012), Revora (REVORAVOXELS, 2012); ou criados pelo autor da dissertação para testar problemas topológicos.

Os veículos dos jogos contêm em média entre 2 mil e 60 mil voxels usados e a parte interna do volume é parcialmente pintada. Já os que são produzidos pelos usuários podem passar de 200 mil voxels usados somente em suas superfícies.

6.2 – Técnicas para comparação de resultados

Conforme visto anteriormente no capítulo 3, o único trabalho que conhecemos com o objetivo de resolver o problema investigado nessa dissertação é a solução proposta por Muniz *et al.* (2010), que está implementada no Voxel Section Editor III. Por esse motivo, ela será utilizada neste capítulo para comparação de resultados com a técnica proposta nesta dissertação.

6.3 – Análise de desempenho

O objetivo dessa análise é verificar se o tempo de execução dessa técnica é viável em programas de edição de objetos gráficos que precisam de uma resposta para o usuário em tempo interativo. Além disso, faz-se uma comparação da solução proposta nesta dissertação com outras soluções para averiguar o melhor momento de corrigir os problemas topológicos do modelo de voxels oriundos de aliasing.

Para testar o tempo de execução das etapas da técnica proposta, foram utilizados modelos de alta resolução ilustrados na figura 6.3. Todos eles foram feitos por entusiastas. A tabela 6.1 mostra as características de cada modelo, sendo o Nemesis o modelo mais complexo.

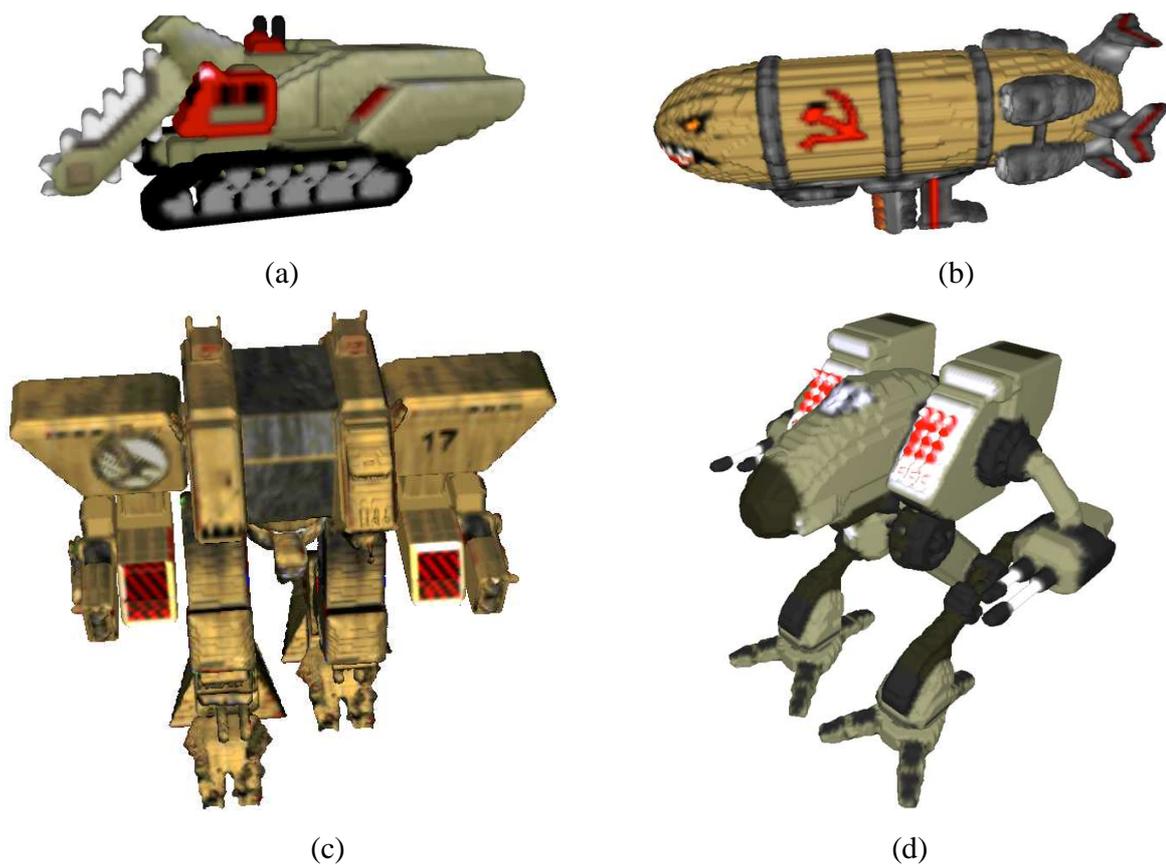


Figura 6.3 – Resultados obtidos com modelos de alta resolução utilizados nos testes de desempenho. (a) War Miner (WARMINER, 2012). (b) Kirov Airship MKII (KIROV, 2012). (c) GDI Nemesis (NEMESIS, 2012). (d) Mad Dog (MADDOG, 2012).

Tabela 6.1 – Complexidade dos modelos utilizados para verificação do desempenho dos métodos.

Nome do Modelo	Dimensões	Total de Voxels	Voxels Sólidos
War Miner	29 x 26 x 65	49 010	6 409
Kirov Airship MKII	42 x 67 x 140	393 960	29 558
GDI Nemesis	155 x 90 x 94	2 973 200	78 937
Mad Dog	100 x 97 x 71	688 700	24 570

O hardware utilizado como ambiente de testes é um desktop com um processador de três cores (AMD Phenom x3) com 2.4ghz por core, 8 gb de memória RAM DDR2 800Mhz, placa de vídeo GeForce 560 Ti com 1gb e sistema operacional Windows XP Professional SP2 64 bits.

Constatou-se que o gargalo do método proposto ocorre na extração da superfície de bordo por motivos de alocação de memória. Durante esse procedimento, o modelo GDI Nemesis alocou aproximadamente 570mb de memória RAM, enquanto os outros modelos usaram menos de 350mb. O uso considerável de memória deve-se aos *buffers* utilizados para mapear o volume, para evitar a criação de vértices com identificação repetida e para eliminar faces internas. As outras etapas usam menos da metade de memória usada pela extração da superfície de bordo. Os resultados são obtidos em tempo interativo e o custo computacional é baixo para computadores atuais, o que viabiliza o uso dessa técnica em editores de modelos geométricos. A tabela 6.2 mostra o tempo de execução das etapas principais nesses modelos.

Tabela 6.2 – Tempo de execução de cada etapa do método proposto (em ms).

Nome do Modelo	Extração da Superfície	Suavização de Normais	Suavização de Cores	Suavização da Malha	Total
War Miner	403	37	47	93	579
Kirov Airship MKII	2 356	105	122	251	2 835
GDI Nemesis	13 964	442	511	1 012	15 930
Mad Dog	3 464	133	149	319	4 065

O método de Muniz *et al.* (2010) que está sendo comparado neste trabalho, sem a etapa da reamostragem que foi proposta para corrigir problemas topológicos, tem um tempo de execução muito menor, mostrado na tabela 6.3. As operações de suavização de normais e cores são as mesmas do método proposto.

Tabela 6.3 – Tempo de execução de cada etapa do método usado para comparação (em ms).

Nome do Modelo	Extração da Superfície	Suavização da Malha	Suavização de Normais	Suavização de Cores	Total
War Miner	82	10	13	22	127
Kirov Airship MKII	563	34	44	64	706
GDI Nemesis	3 128	122	147	247	3 645
Mad Dog	730	42	60	82	915

O tempo de execução menor provém do fato que a técnica obtém um modelo poligonal com uma resolução maior devido à subdivisão durante a etapa de extração da superfície de bordo. A tabela 6.4 exhibe o tempo de execução do método de Muniz *et al.* (2010) nos modelos War Miner e Mad Dog após a reamostragem dos dados volumétricos. Percebe-se que a técnica proposta nesta dissertação obtém um resultado topologicamente correto em um tempo de execução menor. A tabela 6.4 não incluiu o GDI Nemesis e Kirov Airship MKII, devido às limitações do tamanho do volume no Voxel Section Editor III.

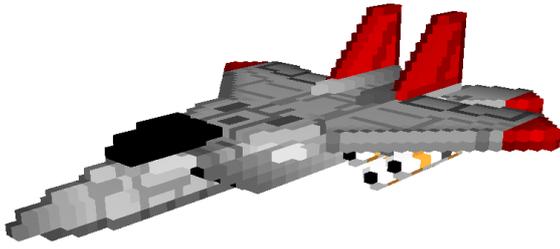
Tabela 6.4 – Tempo de execução de cada etapa do método usado para comparação com reamostragem (em ms).

Nome do Modelo	Extração da Superfície	Suavização da Malha	Suavização de Normais	Suavização de Cores	Total
War Miner	640	43	50	88	822
Mad Dog	6 309	175	225	343	7 053

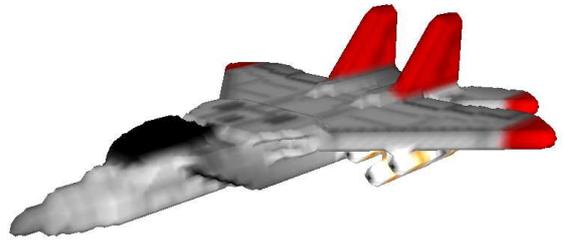
Não é necessário conhecer o tempo que a reamostragem nos dados volumétricos leva para ser executada para se concluir que é mais rápido refinar o campo escalar que define o volume durante a construção da superfície de bordo. Além disso, o método atual obtém superfícies mais compactas do que as que são obtidas pelo método de Muniz *et al.* (2010) após a reamostragem dos dados volumétricos.

6.4 – Resultados obtidos

A figura 6.4 ilustra os resultados obtidos com diversos modelos mostrados nesta dissertação, além de seus conjuntos de voxels originais.



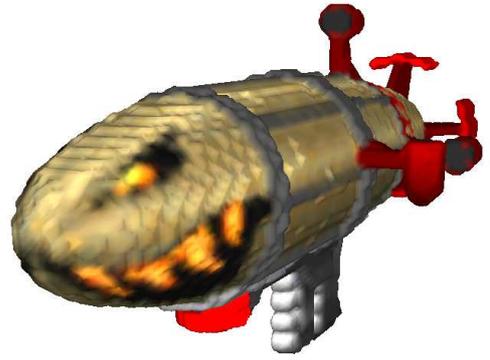
(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

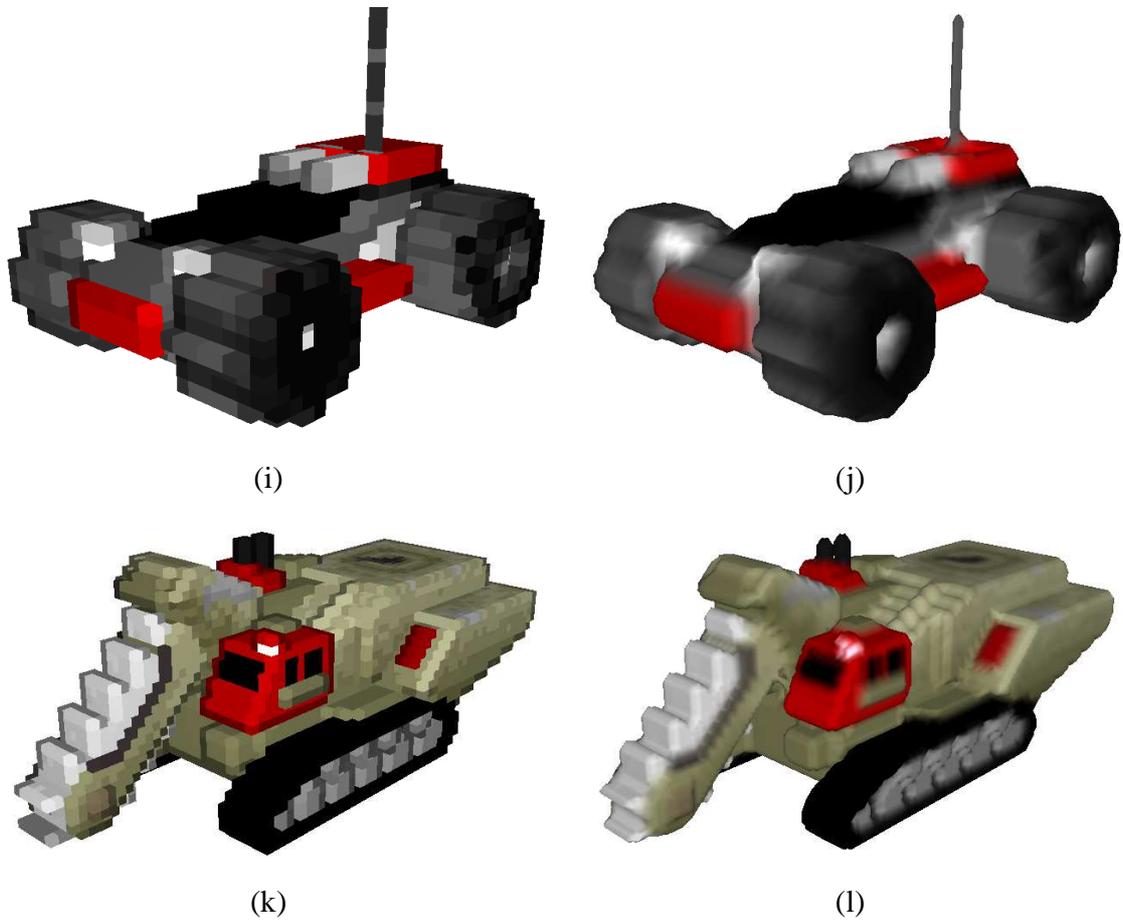


Figura 6.4 – Modelos originais e resultados obtidos de vários modelos. (a) FS TomCat original. (b) FS TomCat após a suavização. (c) Kirov Airship original. (d) Kirov Airship após a suavização. (e) ROTD Buggy original. (f) ROTD Buggy após suavização. (g) Nighthawk original. (h) Nighthawk após suavização. (i) Nod Buggy original. (j) Nod Buggy após a suavização. (k) War Miner original. (l) War Miner após a suavização.

Capítulo 7 – Conclusão

A proposta deste trabalho constituiu em resolver o problema de extração de superfícies poligonais a partir de modelos volumétricos gerando variedades combinatórias orientáveis que são consistentes com as relações de adjacências entre os voxels originais. Pretende-se, com isso, usar técnicas de manipular e esculpir voxels para complementar as ferramentas de modelagem geométrica existentes. A resolução desse problema deve levar em conta desafios de lidar com volumes com baixa resolução de dados.

Ao longo da pesquisa para este projeto, foram encontradas referências sobre algoritmos de extração de superfície poligonais a partir de isosuperfícies, de reamostragem de imagens com correção topológica de problemas oriundos de aliasing, do uso de técnicas de filtragem espacial em malhas e de outras formas de corrigir malhas cujo bordo não é uma variedade combinatória orientável de dimensão 2.

A partir do conhecimento adquirido com essas técnicas, elaborou-se assim a metodologia e as etapas a serem cumpridas que constituíram em extrair uma malha de triângulos, aplicar suavizações de normais e cores e, finalmente, remover características facetadas da malha gerada.

Assim, buscou-se mostrar e analisar os vários métodos pesquisados e implementar uma metodologia para encontrar o melhor resultado. Primeiramente, buscou-se uma forma de extrair a superfície de bordo do modelo volumétrico, mapear o volume para detectar as regiões que receberão triângulos, refinando-as para detectar os seus vértices de forma com que se obtém volumes que representem variedades combinatórias de dimensão 3, sem que haja descontinuidades nas adjacências dos voxels originais. Essa etapa foi a mais importante da pesquisa.

Na segunda parte, suavizaram-se as normais e as cores enfatizando o uso de interpolação linear para reduzir o contraste desses atributos entre os vértices. Em seguida, suavizou-se a malha. Por fim, mostraram-se os resultados do trabalho, por meio de testes de desempenho global e de qualidade de cada etapa usando o Voxel Section Editor III com análises, além de explicar a razão de suas técnicas alternativas não terem sido incluídas no método proposto.

O objeto deste trabalho limitou-se a obter um método de transformar modelos baseados em voxels em geometrias baseada em bordo, em que cada vértice tem uma posição, normal e cor.

Consegue-se conservar as relações de adjacências entre os voxels originais e resolver todos os problemas de topologia, independentemente do genus do complexo celular de entrada. Todas as etapas são executadas em tempo interativo. O procedimento de mapeamento do volume pode detectar regiões internas como superfície se o modelo tiver superfícies com bordo. A solução desse problema está fora do escopo deste trabalho, podendo ficar a cargo do artista preencher os espaços vazios da superfície do objeto.

A etapa de refinamento do método de extração do bordo da superfície pode ser transformada em uma reamostragem que corrige problemas topológicos no modelo de voxels, porém sua aplicação é mais cara do que refinar os dados durante a extração da malha. Essa reamostragem impede que quase todos os problemas de ambiguidade do *Marching Cubes* ocorra no volume, com exceção do caso 3. A malha gerada é densa e robusta para as métricas utilizadas, com arestas de tamanho e ângulos previsíveis, descartando a necessidade de um remalhamento. Além disso, o resultado gerado é uma variedade combinatória orientável, mesmo com a presença do caso 3 do *Marching Cubes* após a execução da etapa de refinamento.

A técnica de suavização de malhas sugerida nesta pesquisa distingue-se das técnicas anteriores por utilizar uma estimativa de curvatura baseada em variação das normais nas faces incidentes a um vértice e por utilizar um fator de escala do deslocamento dos vértices que varia localmente. A operação de suavização requer apenas uma única iteração e faz mudanças insignificantes na direção das normais. Além disso, garante-se que o tipo de concavidade das regiões não seja alterado após a aplicação da técnica.

7.1 – Trabalhos Futuros

O primeiro trabalho a ser considerado é apresentar uma justificativa formal para o correto funcionamento da reamostragem e poligonização no que diz respeito à geração das 3-variedades a partir do modelo inicial. Outro possível trabalho é avaliar a aplicação da técnica de reamostragem proposta para o redimensionamento de imagens, sem a necessidade de gerar malhas. De fato, esta linha já vem sendo investigada conforme o resultado apresentado na figura 7.1.



Figura 7.1 – Yoshi da figura 2.10 aumentado em 8 vezes, estendendo o conceito de reamostragem da região volumétrica proposto neste trabalho para imagens.

Outra possível expansão deste trabalho consiste em estudos para extrair o atlas de texturas desses modelos e para obter texturas de forma autônoma, para vários tipos de *shaders* para geração de *bump mapping*, *relief mapping*, entre outros. O tratamento de modelos de voxels com múltiplos materiais é também um tópico que deve ser considerado no futuro, além de formas de preservar arestas.

Finalmente, espera-se no futuro investigar técnicas para transferir detalhes de modelos detalhados representados por bordo, previamente obtidos, para os modelos reconstruídos a partir de modelos volumétricos de baixa resolução.

BIBLIOGRAFIA

(3DCOAT, 2012) 3D-Coat, Ferramenta comercial de modelagem e edição de modelos com geometria tridimensional desenvolvido pela 3D-Coat. Informações e versão demo disponíveis em <<http://3d-coat.com/>>. Acesso em 08 de julho de 2012.

(3DSMAX, 2012) 3ds Max, Ferramenta comercial de modelagem e edição de modelos com geometria tridimensional desenvolvido pela Autodesk. Informações e versão trial disponível em <http://usa.autodesk.com/adsk/servlet/pc/index?id=19520751&siteID=123112&s_tnt=36218:1:0>. Acesso em 08 de julho de 2012.

(BISCHOFF e KOBELT, 2006) BISCHOFF, S., KOBELT, L., “Extracting Consistent and Manifold Interfaces from Multi-valued Volume Data Sets”, *Bildverarbeitung für die Medizin* 2006, Springer Berlin Heidelberg, pp. 281–285.

(BISCHOFF *et al.*, 2005) BISCHOFF, S., PAVIC, D., KOBELT, L. “Automatic Restoration of Polygon Models”; *ACM Transactions on Graphics*", Out. 2005, vol. 24, No. 4, pp. 1332-1352.

(BLENDER, 2012) Blender, Ferramenta gratuita de código aberto de modelagem e edição de modelos com geometria tridimensional desenvolvido pelo Blender Foundation. Disponível em <<http://www.blender.org/>>. Acesso em 08 de julho de 2012.

(BOTSCH *et al.*, 2010) BOTSCH, M., KOBELT, L., PAULY, M. , ALLIEZ, P., LEVY, B. “Polygonal Mesh Processing”, A.K. Peters/CRC Press, ISBN 978-1-56881-426-1, 2010.

(BOURKE, 2012) Polygonizing a scalar field(Marching Cubes), Tutorial extraído do site PaulBourke.net. Disponível em: <<http://paulbourke.net/geometry/polygonise/>>. Acesso em 8 de agosto de 2012.

(CHERNYAEV, 1995) CHERNYAEV, E. V. “Marching Cubes 33: Construction of Topologically Correct Isosurfaces”; Technical Report CERN CN 95-17, CERN, 1995.

(CINEMA4D, 2012) Cinema 4D Studio, Ferramenta comercial de modelagem e edição de modelos com geometria tridimensional desenvolvido pela Maxon. Informações e versão demo disponíveis em <<http://www.maxon.net/products/cinema-4d-studio/who-should-use-it.html>>. Acesso em 08 de julho de 2012.

(DEPIXELIZING_COMP, 2012) Depixelizing Pixel Art, Página que compara a técnica de vetorização de imagens desenvolvida por Kopf e Lischinski com o outras técnicas de reamostragem de imagens. Disponível em: <http://research.microsoft.com/en-us/um/people/kopf/pixelart/supplementary/multi_comparison.html>. Acesso em 24 de julho de 2012.

(DIGNE *et al.*, 2012) DIGNE, J., COHEN-STEINER, D., ALLIEZ, P., DESBRUN, M., GOES, F. de. “Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets”; INRIA Research Report 7991, Jun 2012.

(FSTOMCAT, 2012) ::YR Argentina: -- ::The Place For All Your RA2/YR Needs::, Modelo de voxels FS Tomcat, criado pelo usuário ihateharriers, hospedado na versão antiga do site YR Argentina. Disponível em: <<http://yrarg.cncguild.net/old/index.php?page=voxels/vxlc1>>. Acesso em 10 de agosto de 2012.

(JU et al., 2002) JU, T., LOSASSO, F., SCHAEFER, S., AND WARREN, J. "Dual contouring of hermite data."; ACM Transactions on Graphics", 2002, vol. 21, No. 3, pp. 339-346.

(KIROV, 2012) Kirov Airship Mark II (RA2). Modelo de voxels público criado pelo usuário Arikado do Project Perfect Mod Forums. Disponível em <<http://www.ppmsite.com/forum/viewtopic.php?t=18995>>. Acesso em 22 de agosto de 2012.

(KOBBELT *et al.*, 2001) KOBBELT, L. P., BOTSCH, M., SCHWANECKE, U., SEIDEL, H. P., "Feature sensitive surface extraction from volume data" ; Computer Graphics; Siggraph 2001 Conference Proceedings; Ago. 2001; ACM Siggraph; pp. 57-66.

(KOPF e LISCHINSKI, 2011) KOPF, J., LISCHINSKI, D. "Depixelizing Pixel Art"; ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2011; Jul., 2011; vol. 30, No. 4; ACM SIGGRAPH; Article No. 99.

(LAGE, 2006) FERREIRA, M. de O. L., "Estrutura de Dados Topológicas Escalonáveis para Variedades de Dimensão 2 e 3", Dissertação de Mestrado de Marcos de Oliveira Lage Ferreira, Departamento de Matemática, Centro Técnico Científico, Pontifícia Universidade Católica do Rio de Janeiro, Fev 2006.

(LEWINER *et al.*, 2003) LEWINER, T., LOPES, H., VIEIRA, A. W., TAVARES, G. "Efficient implementation of marching cubes cases with topological guarantees"; Journal of Graphics Tools; A.K.Peters; Dez. 2003; vol. 8, No. 2; pp 1-15.

(LEWINER *et al.*, 2010) LEWINER, T., LOPES, H., NEDEIROS, E., TAVARES, G., VELHO L. "Topological mesh operators"; Computer Aided Geometric Design 27(1); Jan 2010; pp. 1-22.

(LORENSEN e CLINE, 1987) LORENSEN, W. E., CLINE, H. E. "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.

(MADOG, 2012) Mad Dog Mech. Modelo de voxel público criado pelo usuário WeeRaby2k do site Yuri's Revenge: Argentina. Disponível em <<http://yrarg.cncguild.net/old/index.php?page=voxels/vxlg9>>. Acesso em 22 de agosto de 2012.

(MAYA, 2012) Maya, Ferramenta comercial de modelagem e edição de modelos com geometria tridimensional desenvolvido pela Autodesk. Informações e versão trial disponível em <<http://usa.autodesk.com/maya/>>. Acesso em 08 de julho de 2012.

(MEYER et al., 2003) MEYER, M., DESBRUN, M., SCHRÖDER P., BARR A. H. Barr. "Discrete Differential-Geometry Operators for Triangulated 2-Manifolds." In Visualization and

Mathematics III, edited by Hans-Christian Hege and Konrad Polthier, pp. 35–57. Heidelberg: Springer-Verlag, 2003.

(MODDB, 2012) Textures in Red Alert 3 tutorial – ModDB, Modelo utilizado em um tutorial criado pelo usuário ferriswheel42 do ModDB.com com o objetivo de explicar como gerar modelos poligonais com texturas para o jogo Red Alert 3. Tutorial disponível em <<http://www.moddb.com/mods/red-alert-3-paradox/tutorials/textures-in-red-alert-3>>. Acesso em 28 de julho de 2012.

(MUNIZ *et al.*, 2010) MUNIZ, C. E. V., MONTENEGRO, A. A., CLUA, E. G. W. “Fast polygonization and texture map extraction from volumetric objects based on surface fairing using a modified discrete laplacian operator”, IX Brazilian Symposium on Computer Games and Digital Entertainment, 2010, Florianópolis. Proceedings of the IX Brazilian Symposium on Computer Games and Digital Entertainment. Porto Alegre : Sociedade Brasileira da Computação, 2010. v. 9. p. 115-122.

(MUNIZ, 2010) MUNIZ, C. E. V., “Extração de Modelos Poligonais a Partir de Modelos Volumétricos”, Projeto Final de Graduação de Carlos Eduardo Vaisman Muniz, Instituto de Computação, Universidade Federal Fluminense, 2010.

(NEMESIS, 2012) GDI Nemesis (TS). Modelo de voxel público criado pela usuária areaSZ do Project Perfect Mod Forums. Disponível em <<http://www.ppmsite.com/forum/viewtopic.php?t=5793>>. Acesso em 22 de agosto de 2012.

(NEWMAN e YI, 2006) NEWMAN, T., YI, H., “A survey of the marching cubes algorithm”, Computers & Graphics, Vol. 30, No, Elsevier, Out 2006, pp. 854-879.

(NINTENDO, 2012) Nintendo.com. Site oficial da Nintendo, detentora dos direitos autorais do Yoshi, Mario, Boo e outros personagens. Disponível em <<http://www.nintendo.com/>>. Acesso em 21 de setembro de 2012.

(PPMVOXELS, 2012) Project Perfect Mod Voxels. Fórum onde membros da comunidade de modificação dos jogos Command & Conquer Tiberian Sun e Red Alert 2 publicam seus modelos volumétricos. Disponível em: <<http://www.ppmsite.com/forum/index.php?f=111>>. Acesso em 14 de agosto de 2012.

(REVORAVOXELS, 2012) Revora Voxel/SHP Downloads. Fórum onde membros da comunidade de modificação dos jogos Command & Conquer Tiberian Sun e Red Alert 2 publicam seus modelos volumétricos. Disponível em: <<http://forums.revora.net/index.php?showforum=328>>. Acesso em 14 de agosto de 2012.

(ROSSIGNAC e CARDOZE, 1999) ROSSIGNAC, J., CARDOZE, D. “Matchmaker: Manifold BReps for non-manifold r-sets”; Proceeding SMA '99 Proceedings of the fifth ACM symposium on Solid modeling and applications; ACM New York, NY, USA ©1999; pp 31-41.

(ROTD, 2012) Return of the Dawn, Modelo volumétrico criado por Reaperr da modificação Return of the Dawn para o jogo Command & Conquer Tiberian Sun. Disponível em: <<http://www.ppmsite.com/forum/index.php?f=116>>. Acesso em 28 de julho de 2012.

(SCHAEFER e WARREN, 2004) SCHAEFER, S., WARREN, J., “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.

(STANFORD, 2012) The Stanford 3D Scanning Repository. Página que armazena modelos tridimensionais escaneados pela Universidade de Stanford para fins de pesquisas. Disponível em <<http://graphics.stanford.edu/data/3Dscanrep/>>. Acesso em 23 de agosto de 2012.

(TAUBIN, 1995) TAUBIN, G. “A Signal Processing Approach To Fair Surface Design”, Computer Graphics; Siggraph '95 Conference Proceedings; Set. 1995; ACM Siggraph; pp. 351-358.

(VXLSE3, 2012) VOXEL SECTION EDITOR III. Ferramenta gratuita de modelagem e edição de modelos volumétricos subdivididos em pixels. Disponível em <<http://www.ppmsite.com/index.php?go=vxlseinfo>>. Acesso em 08 de julho de 2012.

(VXLSE3SVN, 2012) VOXEL SECTION EDITOR III SVN, Código fonte do Voxel Section Editor III. Disponível em <<http://svn.ppmsite.com/listing.php?reaname=OS%20Voxel%20Tools&path=%2F&sc=0>>. Acesso em 08 de julho de 2012.

(WARMINER, 2012) WarMiner (RA2). Modelo de voxels público criado pelo usuário Gangster do Project Perfect Mod Forums. Disponível em <<http://www.ppmsite.com/forum/viewtopic.php?t=7158>>. Acesso em 22 de agosto de 2012.

(WIKIPEDIA, 2012) Marching Squares – Wikipedia, the free Encyclopedia, Explicação sobre o método Marching Squares e suas variações. Disponível em: <http://en.wikipedia.org/wiki/Marching_squares>. Acesso em 8 de agosto de 2012.

(XCC, 2012) XCC Homepage. Página que desenvolve e hospeda utilitários usados para modificar jogos da série Command & Conquer. Disponível em: <<http://xhp.xwis.net/>>. Acesso em 14 de agosto de 2012.

(YRARG, 2012) Yuri's Revenge: Argentina. Site onde membros da comunidade de modificação dos jogos Command & Conquer Tiberian Sun e Red Alert 2 publicam seus modelos volumétricos. Disponível em: <<http://yrarg.cncguild.net/>>. Acesso em 14 de agosto de 2012.
Advancing Fronts.”, Procedures of the National Academy of Science 93 (1996), pp. 1591–1595.

Apêndice A – Lookup Table

A Lookup Table da técnica de extração do bordo de modelos volumétricos proposta neste trabalho foi gerada a partir do sistema de coordenadas mostrada na figura A.1. Neste trabalho definiu-se que o eixo x horizontal é positivo para o lado direito, o eixo y vertical é positivo para cima e o eixo z de profundidade é positivo para fora da tela.

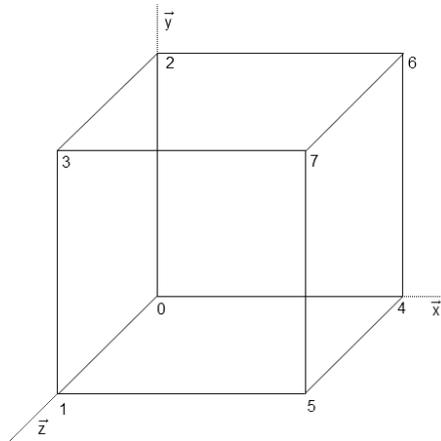


Figura A.1 – Sistema de coordenadas usado para gerar a Lookup Table e a numeração dos vértices.

A tabela A.1 mostra todas as configurações que devem aparecer na Lookup Table implementada neste trabalho. Os casos estão ilustrados na figura 4.13. A rotação está descrita nos atributos base e topo.

Tabela A.1 – Lista de todos os casos da Lookup Table e como cada um é poligonizado.

Nº Binário	Decimal	Polígonos Gerados	Caso	Topo	Base
00010111	23	{(0,1,2),(0,2,4),(0,4,1),(1,4,2)}	a	-	-
00011111	31	{(0,1,3,2),(0,4,1),(0,2,4),(1,4,3),(2,3,4)}	b	4	Esq.
00101011	43	{(0,1,3),(1,3,5),(0,3,5),(0,5,1)}	a	-	-
00101111	47	{(0,1,3,2),(0,5,1),(0,2,5),(1,5,3),(2,3,5)}	b	5	Esq.
00110111	55	{(0,4,5,1),(0,1,2),(0,2,4),(1,5,2),(2,5,4)}	b	2	Bai.
00111011	59	{(0,4,5,1),(0,1,3),(0,3,4),(1,5,3),(3,5,4)}	b	3	Bai.
00111111	63	{(0,4,5,1),(0,1,3,2),(2,3,5,4),(0,2,4),(1,5,3)}	c	2,3	Bai.
01001101	77	{(0,3,2),(0,2,6),(2,3,6),(0,6,3)}	a	-	-

01001111	79	{(0,1,3,2),(0,6,1),(0,2,6),(1,6,3),(2,3,6)}	b	6	Esq.
01010111	87	{(0,2,6,4),(0,4,1),(0,1,2),(1,6,2),(1,4,6)}	b	1	Atr.
01011101	93	{(0,2,6,4),(0,3,2),(0,4,3),(2,3,6),(3,4,6)}	b	3	Atr.
01011111	95	{(0,1,3,2),(0,2,6,4),(1,4,6,3),(0,4,1),(2,3,6)}	c	4,6	Esq.
01101001	105	{(0,5,3),(0,6,5),(3,5,6),(0,3,6)}	a	-	-
01101111	111	{(0,1,3,2),(0,5,1),(0,6,5),(0,2,6),(1,5,3),(2,3,6),(3,5,6)}	d	4,7	Esq.
01110001	113	{(0,4,5),(0,6,4),(4,6,5),(0,5,6)}	a	-	-
01110011	115	{(0,4,5,1),(0,6,4),(0,1,6),(1,5,6),(4,6,5)}	b	6	Bai.
01110101	117	{(0,2,6,4),(0,5,2),(0,4,5),(2,5,6),(4,6,5)}	b	5	Atr.
01110111	119	{(0,4,5,1),(0,2,6,4),(1,5,6,2),(0,1,2),(4,6,5)}	c	2,6	Bai.
01111011	123	{(0,4,5,1),(0,1,3),(0,6,4),(0,3,6),(1,5,3),(3,5,6),(4,6,5)}	d	2,7	Bai.
01111101	125	{(0,2,6,4),(0,3,2),(0,5,3),(0,4,5),(2,3,6),(3,5,6),(4,6,5)}	d	1,7	Atr.
01111111	127	{(0,4,5,1),(0,1,3,2),(0,2,6,4),(1,5,3),(2,3,6),(3,5,6),(4,6,5)}	e	-	-
10001110	142	{(1,3,2),(1,7,3),(2,3,7),(1,2,7)}	a	-	-
10001111	143	{(0,1,3,2),(0,7,1),(0,2,7),(1,7,3),(2,3,7)}	b	7	Esq.
10010110	150	{(1,4,7),(2,7,4),(1,7,2),(1,2,4)}	a	-	-
10011111	159	{(0,1,3,2),(0,4,1),(0,2,4),(1,7,3),(1,4,7),(2,7,4),(2,3,7)}	d	5,6	Esq.
10101011	171	{(1,5,7,3),(0,5,1),(0,1,3),(0,7,5),(0,3,7)}	b	0	Fre.
10101110	174	{(1,5,7,3),(1,3,2),(1,2,5),(2,7,5),(2,3,7)}	b	2	Fre.
10101111	175	{(0,1,3,2),(1,5,7,3),(0,2,7,5),(0,5,1),(2,3,7)}	c	3,7	Esq.
10110010	178	{(1,5,7),(4,7,5),(1,4,5),(1,7,4)}	a	-	-
10110011	179	{(0,4,5,1),(0,7,4),(0,1,7),(1,5,7),(4,7,5)}	a	7	Bai.
10110111	183	{(0,4,5,1),(0,1,2),(0,2,4),(1,5,7),(2,7,4),(2,1,7),(4,7,5)}	d	4,5	Bai.
10111010	186	{(1,5,7,3),(1,3,4),(1,5,4),(3,7,4),(4,7,5)}	b	4	Fre.
10111011	187	{(0,4,5,1),(1,5,7,3),(0,3,7,4),(0,1,3),(4,7,5)}	c	3,7	Bai.

10111110	190	{(1,5,7,3),(1,2,3),(1,2,4),(1,4,5),(2,7,4),(2,3,7),(4,5,7)}	d	0,6	Fre.
10111111	191	{(0,4,5,1),(0,1,3,2),(1,5,7,3),(0,2,4),(2,7,4),(2,3,7),(4,7,5)}	e	-	-
11001101	205	{(2,3,7,6),(0,3,2),(0,7,3),(0,2,6),(0,6,7)}	b	0	Cim.
11001110	206	{(2,3,7,6),(1,3,2),(1,2,6),(1,7,3),(1,6,7)}	b	1	Cim.
11001111	207	{(0,1,3,2),(2,3,7,6),(0,6,7,1),(0,2,6),(1,7,3)}	c	6,7	Esq.
11010100	212	{(2,7,6),(4,6,7),(2,6,4),(2,4,7)}	a	-	-
11010101	213	{(0,2,6,4),(0,7,2),(0,4,7),(2,7,6),(4,6,7)}	b	7	Atr.
11010111	215	{(0,2,6,4),(0,4,1),(0,1,2),(1,4,7),(2,7,6),(2,1,7),(4,6,7)}	d	3,5	Atr.
11011100	220	{(2,3,7,6),(2,4,3),(2,6,4),(3,4,7),(4,6,7)}	b	4	Cim.
11011101	221	{(0,2,6,4),(2,3,7,6),(0,4,7,3),(0,3,2),(4,6,7)}	c	3,7	Atr.
11011110	222	{(2,3,7,6),(1,3,2),(1,7,3),(1,2,4),(1,4,7),(2,6,4),(4,6,7)}	d	0,5	Cim.
11011111	223	{(0,1,3,2),(0,2,6,4),(2,3,7,6),(0,4,1),(1,7,3),(1,4,7),(4,6,7)}	e	-	-
11101000	232	{(3,5,7),(5,6,7),(3,7,6),(3,6,5)}	a	-	-
11101010	234	{(1,5,7,3),(1,6,5),(1,3,6),(3,7,6),(5,6,7)}	b	6	Fre.
11101011	235	{(1,5,7,3),(0,5,1),(0,1,3),(0,6,5),(0,3,6),(3,7,6),(5,6,7)}	d	2,4	Fre.
11101100	236	{(2,3,7,6),(2,5,3),(2,6,5),(3,5,7),(5,6,7)}	b	5	Cim.
11101101	237	{(2,3,7,6),(0,3,2),(0,5,3),(0,6,5),(0,2,6),(3,5,7),(5,6,7)}	d	1,4	Cim.
11101110	238	{(1,5,7,3),(2,3,7,6),(1,2,6,5),(1,3,2),(5,6,7)}	c	2,6	Fre.
11101111	239	{(0,1,3,2),(1,5,7,3),(2,3,7,6),(0,5,1),(0,6,5),(0,2,6),(5,6,7)}	e	-	-
11110001	241	{(4,6,7,5),(0,6,4),(0,4,5),(0,7,6),(0,5,7)}	b	0	Dir.
11110010	242	{(4,6,7,5),(1,6,4),(1,4,5),(1,7,6),(1,5,7)}	b	1	Dir.
11110011	243	{(0,4,5,1),(4,6,7,5),(0,1,7,6),(0,6,4),(1,5,7)}	c	6,7	Bai.
11110100	244	{(4,6,7,5),(2,4,5),(2,6,4),(2,7,6),(2,5,7)}	b	2	Dir.

11110101	245	{(0,2,6,4),(4,6,7,5),(0,5,7,2),(0,4,5),(2,7,6)}	c	6,7	Atr.
11110110	246	{(4,6,7,5),(1,7,2),(1,2,4),(1,4,5),(1,5,7),(2,6,4),(2,7,6)}	d	0,3	Dir.
11110111	247	{(0,4,5,1),(0,2,6,4),(4,6,7,5),(0,1,2),(1,7,2),(1,5,7),(2,7,6)}	e	-	-
11111000	248	{(4,6,7,5),(3,4,5),(3,6,4),(3,7,6),(3,5,7)}	b	3	Dir.
11111001	249	{(4,6,7,5),(0,5,3),(0,6,4),(0,4,5),(0,3,6),(3,7,6),(3,5,7)}	d	1,2	Dir.
11111010	250	{(1,5,7,3),(4,6,7,5),(1,3,6,4),(1,4,5),(3,7,6)}	C	6,7	Fre.
11111011	251	{(0,4,5,1),(1,5,7,3),(4,6,7,5),(0,1,3),(0,6,4),(0,3,6),(3,7,6)}	e	-	-
11111100	252	{(2,3,7,6),(4,6,7,5),(2,4,5,3),(2,6,4),(3,5,7)}	c	6,7	Cim.
11111101	253	{(0,2,6,4),(2,3,7,6),(4,6,7,5),(0,3,2),(0,5,3),(0,4,5),(3,5,7)}	e	-	-
11111110	254	{(1,5,7,3),(2,3,7,6),(4,6,7,5),(1,3,2),(1,2,4),(1,4,5),(2,6,4)}	e	-	-
11111111	255	{(0,1,3,2),(0,4,5,1),(0,2,6,4),(1,5,7,3),(2,3,7,6),(4,6,7,5)}	f	-	-