

Laboratório VISGRAF

Instituto de Matemática Pura e Aplicada

Uma Abordagem Estocastica para Objetos Solidos

*Esdras Medeiros, Thomas Lewiner
Helio Lopes, Luiz Velho*

Technical Report TR-07-01 Relatório Técnico

April - 2007 - April

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

Uma Abordagem Estocástica para Multiresolução de Objetos Sólidos com Controle Topológico

Esdras Medeiros, Luiz Velho*
Helio Lopes e Thomas Lewiner †

18 de abril de 2007

*IMPA–Instituto de Matemática Pura e Aplicada, {esdras, lvelho}@impa.br

†PUC–Pontifícia Universidade Católica do Rio de Janeiro, {lopes, lewiner}@mat.puc-rio.br

Resumo

Nesse artigo introduzimos um método de filtragem por operadores topológicos para uma representação em multiresolução de um objeto sólido. Em cada nível de resolução temos uma amostragem por discos de Poisson que possui características particulares. Com essa filtragem é possível controlar as mudanças de topologia e de resolução de forma unificada através dos operadores estelares e de *handle*.

Sumário

1	Introdução	4
1.1	Trabalhos Relacionados	5
1.2	Contribuições	5
1.3	Sumário	5
2	Conceitos e Notações Preliminares	6
2.1	Complexos Simpliciais	6
2.2	Triangulação de Delaunay e Diagrama de Voronoi	7
2.3	Alfa Complexos e <i>Alpha Shapes</i>	7
2.4	Alfa Sólido e Alfa Complexo Sólido	8
3	Estrutura de Dados	8
4	Operadores Topológicos	10
5	Amostragens por Discos de Poisson	11
5.1	ADP's em \mathbb{R}^2	11
5.2	ADP's em Regiões do Plano	12
6	Multiresolução com ADP's	16
6.1	Família Escalonada	16
6.2	Família de Malhas Escalonadas	18
6.3	Filtragem por Operadores Topológicos	19
7	Extensões	26
7.1	Aproximação do Bordo	26
7.2	Razão de Aspecto	27
7.3	Família Quasi-Escalonada	29
7.3.1	Família de Malhas Quasi-Escalonadas	29
7.3.2	Cadenciando uma Família Quasi-Escalonada	30
7.4	Retornando à Filtragem	31
8	Conclusão	32

1 Introdução

Objetos sólidos, em nosso contexto, são variedades compactas com a mesma dimensão do espaço ambiente. A fronteira é a sua parte mais importante porque determina tanto a forma geométrica quanto a topologia. A topologia, de maneira sucinta, pode ser entendida como a quantidade de buracos e de componentes conexas em um objeto sólido. Tem como principal propriedade a invariância por homeomorfismos.

Nosso trabalho de pesquisa sobre objetos sólidos relaciona quatro assuntos bastante conhecidos na comunidade científica em matemática e em computação gráfica: amostragem, reconstrução, modelagem em multiresolução e topologia.

Amostragem e reconstrução de *objetos gráficos* [1] são máximas em computação gráfica. Graças ao teorema da amostragem de Shannon [5], sinais com banda limitada podem ser completamente representados por amostras espaçadas uniformemente com taxa de amostragem menor que metade da largura de banda. No contexto de superfícies encontramos resultados equivalentes nos trabalhos de Amenta et al. [2] e F. Bernardini et al. [3]. Esses trabalhos determinam condições de amostragem que garantem a existência de uma reconstrução topologicamente equivalente à superfície original. Vale ressaltar que, diferentemente do teorema de Shannon para sinais onde a reconstrução é exata, a reconstrução de uma superfície é representada por uma aproximação simplicial. Em geral, por possuírem propriedades interessantes, essas representações são subconjuntos da triangulação de Delaunay. Como exemplos de tais subconjuntos podemos citar o *crust* [2], o *β -skeleton* [2] e o *alpha shapes* [16].

Os espaços de escala tem como idéia fundamental modelar um sinal $f : \mathbb{R}^d \rightarrow \mathbb{R}$ em uma família de escalas diferentes como $L : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$, onde L é uma transformação definida como a convolução de f com o núcleo Gaussiano. Essa transformação funciona como um filtro passa baixa. À medida que diminuimos a variância do núcleo gaussiano diminuimos também a largura de banda do sinal transformado, permitindo reduzir a taxa de amostragem necessária para o reconstruirmos. Assim, podemos representar um espaço de escala como uma família de amostragens com taxas crescentes. As reconstruções dos elementos da família geram uma sequência de infinita de sinais que, no limite, converge para o sinal original.

Segundo o trabalho de F. Bernardini [3] cada superfície possui uma taxa de amostragem particular que permite ser reconstruída por um *alfa sólido* topologicamente equivalente. Em nosso trabalho adquirimos os resultados equivalentes para objetos sólidos que possuem uma reconstrução simplicial por *alfa complexos sólidos* [6] (e variantes). Veremos mais adiante que um alfa complexo sólido está bastante relacionado ao alfa sólido. Mais precisamente o bordo de um alfa complexo sólido é um alfa sólido.

Fazendo um paralelo com espaços de escala em sinais, as reconstruções por alfa complexos sólidos de uma família de amostragens de um objeto sólido com taxas de amostragem diferentes possuem peculiaridades. À medida que aumentamos a taxa de amostragem podemos identificar dois tipos de detalhes que são realçados: **detalhes geométricos** caracterizados pelas formas do bordo e **detalhes topológicos** caracterizados pela quantidade de buracos. Nesse artigo veremos um modelo de representação em multiresolução de objetos sólidos que trata de forma unificada esses dois tipos de detalhes.

1.1 Trabalhos Relacionados

Existem diversos trabalhos relacionados a representações em escalas diferentes por malhas simpliciais. Nos trabalhos de Kobbelt [9] e Velho et al. [10], os diversos níveis de resolução são gerados por meio de regras de subdivisão. No trabalho de Puppo [12] é feito um estudo sobre multiresolução simplicial de domínios planares. No trabalho de Hoppe [13] encontramos representações em malhas que mudam progressivamente de resolução por meio de operações *edge-collapse*. A limitação dos trabalhos citados, além de outros semelhantes, é que a topologia já está definida pelo objeto representado (ou pela malha base) em todos os níveis de resolução. Em suma, não há mudança de topologia.

Quanto ao tratamento topológico em uma família de complexos simpliciais, o trabalho de Edelsbrunner et al. [14] é um dos que mais se aproximam ao nosso. Nesse trabalho é definida a *persistência topológica* obtida em uma *filtragem topológica*. Seus resultados tem diversas aplicações. Dentre elas destacamos a análise de ruído topológico em amostragens de proteínas e a simplificação de complexos de morse-smale. O ponto fraco desse artigo é que não há nenhuma abordagem relacionada a multiresolução.

1.2 Contribuições

Nosso trabalho tem como contribuição unir representações simpliciais de objetos sólidos bidimensionais em níveis de escalas distintas com controle de mudança de topologia.

Tratamos de amostragem e reconstrução em objetos sólidos com uma abordagem estocástica. Para isso, utilizamos amostragens por discos de Poisson porque nos permitem controlar a escala e também obter uma reconstrução triangular que têm razão de aspecto limitada para o caso bidimensional. Esta última característica implica que temos boa qualidade na malha triangulada.

O nosso principal resultado obtido é um algoritmo (com duas versões) que gera uma sequência de *operadores topológicos de malha* [11] por refinamento ou simplificação. Operadores topológicos podem ser de dois tipos: os estelares (mudam a resolução e a estrutura cominatória da malha) e de *handle* (mudam a topologia). A sequência gerada conseqüentemente vincula os níveis de resolução independentemente da topologia dos tipos de topologia existentes.

1.3 Sumário

O artigo está escrito como segue. Na seção 2 introduzimos e revisitamos conceitos que serão utilizados ao longo do texto. Dentre eles destacamos os complexos simpliciais, família de pontos encaixantes, triangulação de Delaunay, alfa complexo sólido e diagrama de Voronoi. Também estabelecemos algumas notações. Na seção 3 apresentamos a estrutura de dados implementada para representar as malhas triangulares. Na seção 4 estudamos os operadores topológicos que serão utilizados para inserir e remover pontos em uma malha. Esses operadores permitirão a interação entre níveis de resolução. Na seção 5 conceituamos amostragens por discos de Poisson, caracterizamos seu complexo dual e mostramos condições suficientes para termos uma reconstrução que seja topologicamente equivalente ao objeto sólido original. Estabelecemos também vários outros conceitos importantes sobre famílias escalonadas a fim de gerar uma filtragem por

operadores topológicos. Em um dos principais teoremas desta seção afirmamos que a sequência de operações geradas pelo algoritmo de refinamento é inverso à sequência de operações geradas pelo algoritmo de simplificação. Na seção 7 fazemos extensões de resultados anteriores para abranger uma maior classe de amostragens que permitem obter resultados mais práticos em termos de implementação e de qualidade na malha. Na seção 8 fazemos uma conclusão sobre o trabalho e enumeramos os trabalhos futuros.

2 Conceitos e Notações Preliminares

Nesta seção vamos descrever alguns conceitos básicos em topologia e em geometria computacional que serão usados ao longo desse trabalho. Também iremos introduzir o conceito de *Alfa Complexo Sólido*.

2.1 Complexos Simpliciais

Um k -simplexo $\sigma_T = \text{conv}(T)$ é a combinação convexa de um conjunto de pontos linearmente independentes $T \subset \mathbb{R}^n$, $\#T = k + 1$; $0 \leq k \leq n$; onde $\#$ denota cardinalidade. k é a dimensão do simplexo σ_T .

Definição 1. Um *Complexo Simplicial* K é uma coleção de simplexos que satisfazem as seguintes propriedades:

1. Se $\sigma_T \in K$ então $\sigma_U \in K$, $U \subset T$. Dizemos que σ_T é *face* de σ_U .
2. Se $\sigma_U, \sigma_V \in K$, então $\sigma_{T \cap V} = \sigma_U \cap \sigma_V$.

As duas propriedades acima implicam que $\sigma_{T \cap V} \in K$. Denominaremos de $|K|$ como o subespaço de \mathbb{R}^n coberto por K . Um *subcomplexo* L de K é um complexo simplicial tal que $L \subset K$. A dimensão de um complexo simplicial K é dada pela maior dimensão entre todos os simplexos de K . Denominamos por K^i como o suconjunto dos simplexos de K que tem dimensão i .

Um *Complexo Simplicial Sólido* não possui simplexos isolados, i.e., k -simplexos que não são faces de um simplexo de dimensão maior. Dado um complexo simplicial K , a coleção $\overline{K} \subset K$ é o complexo simplicial sólido maximal contido em K . Também denominamos tais complexos simpliciais sólidos de *malha*.

Em um complexo simplicial K de dimensão n dizemos que um simplexo de dimensão $n - 1$ é de *bordo* se ele é face de apenas um simplexo de dimensão n . O operador de bordo ∂ de um complexo simplicial K é tal que $\partial K = \{ \sigma \in K \text{ tal que } \sigma \text{ é de bordo ou é face de um simplexo de bordo} \}$. Observe que ∂K é também um complexo simplicial e $\partial K = \partial \overline{K}$.

Seja $S \subset \mathbb{R}^n$ um conjunto finito de pontos e K um complexo simplicial sólido. Se $K^0 \subset S$, por simplicidade vamos escrever (S, K) como notação (leia-se “par pontos S e complexo simplicial K ” ou simplesmente “par”). A coleção $\mathcal{F} = \{S_i\}_{i \in \{1, 2, \dots, n\}}$ é uma *família de pontos encaixantes* se $S_1 \subset S_2 \dots \subset S_n$. A cada um dos i 's chamamos de *nível*. A partir daí vamos definir uma *família de malhas encaixantes*.

Definição 2. Seja $\mathcal{F} = \{S_i\}_{i \in \{1, 2, \dots, n\}}$ uma família de pontos encaixantes. Dizemos que $\mathcal{M}(\mathcal{F}) = \{(S_i, K_i)\}_{i \in \{1, 2, \dots, n\}}$ é uma *Família de Malhas Encaixantes*.

Note que em uma família de malhas encaixantes as malhas possuem vértices em comum, no entanto, podem aumentar ou diminuir a quantidade de vértices à medida que i cresce. Veremos mais adiante que essa última definição é fundamental para formalizar estruturas em multiresolução com mudança de topologia.

Baseado na estrutura de complexos simpliciais vamos definir alguns artefatos simpliciais tais como Triangulações de Delaunay, Alfa Complexos e *Alpha Shapes*.

2.2 Triangulação de Delaunay e Diagrama de Voronoi

A triangulação ou diagrama de Delaunay de um conjunto de pontos no plano é o conjunto único de triângulos que conectam tais pontos e que satisfazem a propriedade do “círculo vazio”: o círculo circunscrito de cada triângulo não contém nenhum outro ponto. Num certo sentido é a maneira mais natural de triangular um conjunto de pontos. A seguir vamos dar uma definição geral baseada em complexos simpliciais.

Definição 3. Dado um conjunto $S \subset \mathbb{R}^n$ em posição geral, a triangulação de Delaunay de S é o complexo simplicial $DT(S)$ que consiste somente de

1. todos os k -simplexos, σ_T ($0 \leq k \leq n$), com $T \subset S$ tais que a circum-esfera (a menor esfera tal que todos os pontos estão no seu bordo) de T não contém nenhum outro ponto de S , e
2. todos os k -simplexos que são faces de outros simplexos também estão em $DT(S)$.

Definiremos o diagrama de Voronoi e vamos estabelecer sua relação com a triangulação de Delaunay.

Definição 4. Seja S um conjunto de n pontos no plano. Para cada $s \in S$, a *região de Voronoi* $V(s)$ de s é o conjunto de pontos que do plano que está mais próximo de p que outros pontos de S . O *diagrama de Voronoi* $V(S)$ é a partição do plano gerada pelas regiões de voronoi de S .

Temos então a proposição abaixo que é bastante conhecida na literatura.

Proposição 1. A triangulação de Delaunay de S é o grafo dual do diagrama de Voronoi S : dois pontos de S estão unidos por uma aresta na triangulação de Delaunay triangulation se e somente se suas regiões de Voronoi são incidentes no diagrama de Voronoi de S .

Demonstração. Veja [17]. □

2.3 Alfa Complexos e *Alpha Shapes*

Alfa complexos são complexos simpliciais que descrevem níveis de detalhe de aglomerados de pontos. Pela variação de um número real positivo α obtemos formas diferentes da mais refinada à mais grosseira. A forma mais refinada é o próprio conjunto de pontos, que é obtida quando $\alpha = 0$. Enquanto α aumenta, a forma cresce pela adição de simplexos desenvolvendo de cavidades que podem se juntar ou se separar. A forma mais grosseira é a triangulação de Delaunay que é obtida para grandes valores de α . Mais precisamente, Alfa Complexos têm a seguinte definição:

Definição 5. Seja $S \subset \mathbb{R}^n$ um conjunto de pontos em posição geral. Para $T \subset S$ com $\#T \leq n$, sejam b_T e μ_T a menor bola que contém os pontos de T e seu raio respectivamente. Dado $0 \leq \alpha \leq \infty$, o alfa complexo $\mathcal{C}_\alpha(S)$ de S é o subcomplexo de $\text{DT}(S)$ onde o simplexo $\sigma_T \in \text{DT}(S)$ está em $\mathcal{C}_\alpha(S)$ se:

1. $\mu_T < \alpha$ e $b_T \cap S = \emptyset$, ou
2. σ_T é uma face de outro simplexo em $\mathcal{C}_\alpha(S)$.

Observando as definições de triangulação de Delaunay e Alfa Complexo são imediatas as seguintes propriedades:

- P1. If $\alpha_1 \leq \alpha_2$ então $\mathcal{C}_{\alpha_1} \subset \mathcal{C}_{\alpha_2}$,
- P2. $\mathcal{C}_\alpha \subset \text{DT}(S)$, $\forall \alpha > 0$ e
- P3. $\mathcal{C}_\infty = \text{DT}(S)$.

O *alpha shapes* S_α é definido como $|C_\alpha(S)|$. Assim como em alfa complexos obtemos uma triangulação de Delaunay para grandes valores do parâmetro α assim também em *alpha shapes* obtemos precisamente o fecho convexo. De fato, um *alpha shape* é adequado para a generalização do conceito de fecho convexo, sendo bastante usado em muitas aplicações [16].

2.4 Alfa Sólido e Alfa Complexo Sólido

Em geral, o alfa complexo e o *alpha shape* são respectivamente complexos simplicias e polítopos compostos de simplexos de dimensões diversas. Bernardini et al.[4] definiu *solid alpha shape* (ou simplesmente *alpha solid*) como o *alpha shape* sem k -simplexos isolados. De forma semelhante definimos então o *Alfa Complexo Sólido* como o alfa sólido sem k -simplexos isolados. É um tipo de subcomplexo que é uma versão “regularizada” do alfa complexo e como vimos anteriormente é um complexo simplicial sólido maximal. Na figura 1 mostramos a diferença entre alfa complexo e alfa complexo sólido no caso 2D. Vamos denotar o alfa complexo sólido de um conjunto $S \subset \mathbb{R}^n$, dado $0 \leq \alpha \leq \infty$, como $\mathcal{C}_\alpha(S)$. Note que as propriedades P1, P2 e P3 são ainda válidas para alfa complexos sólidos.

3 Estrutura de Dados

Seguiremos a mesma estrutura de dados de Velho et. al [11] para representar malhas 2D ou triangulares. Esta estrutura tem a vantagem de unificar as funcionalidades dos operadores topológicos que veremos na seção 4. Vamos aqui reescrevê-las.

A malha é estruturada como $M = (V, E, F, B)$ onde V, E, F, B são coleções de vértices, arestas, faces, e curvas de bordo respectivamente.

```

struct Surface {
    Container<Face*> faces;
    Container<Edge*> edges;
    Container<Vertex*> vertices;
    Container<Edge*> bndries;
}

```

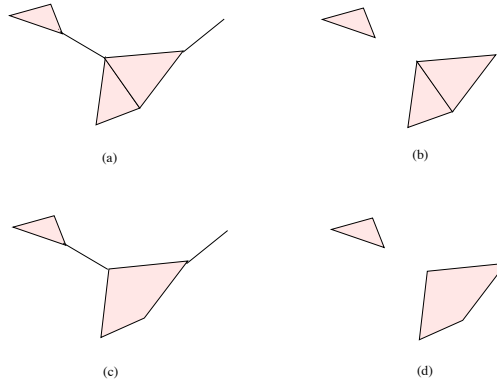


Figura 1: o alfa complexo (a) e seu alfa complexo sólido (b). O *alpha shape* (c) e seu alfa sólido (d).

A face armazena um ponteiro para a primeira *half-edge* do seu ciclo interno.

```
struct Face {
    Half_Edge* he;
}
```

Uma aresta é formada por duas *half-edges*. Se estiver representando uma aresta de bordo uma das *half-edges* aponta para uma face nula.

```
struct Edge {
    Half_Edge he[2];
}
```

A *half-edge* é o elemento central da estrutura de dados. Ela armazena um ponteiro para seu vértice inicial, um ponteiro para a próxima *half-edge* no ciclo da face e ponteiros para a aresta e face a qual ela pertence. Note que a *half-edge* irmã pode ser acessada pelo ponteiro da aresta pai.

```
struct Half_Edge {
    Vertex* org;
    Half_Edge* next;
    Face* f;
    Edge* e;
}
```

O vértice armazena um ponteiro para a *half-edge* incidente.

```
struct Vertex {
    Half_Edge* star_i;
    Data d;
}
```

Na coleção de curvas de bordo o representante de cada elemento é uma aresta que pertence à tal curva.

4 Operadores Topológicos

Nesta seção vamos introduzir um *framework* unificado de operações básicas em variedades de dimensão dois com ou sem bordo. Existem dois tipos de operadores sobre malhas: operadores que mudam a característica topológica (operadores de *handlebody*) e operadores que mudam apenas a resolução e estrutura combinatória (operadores estelares). Vamos apresentá-los sob o ponto de vista computacional, ou seja, como API's, e para uma completa explanação da teoria matemática veja o trabalho de Velho et. al[11]. O nosso principal objetivo é utilizar esses operadores para vincular os diversos níveis de uma família de malhas encaixantes. De maneira mais precisa, o salto de um certo nível para um nível vizinho constituirá uma sequência de aplicações desses operadores.

O nosso primeiro operador é $\text{add}(s)$ que retorna s , onde $s \in S$. Atuará apenas sobre os pontos S de (S, K) . Sua função consiste em adicionar o vértice s ao conjunto S . Analogamente temos $\text{remove}(s)$ como o operador que remove um vértice de S e que retorna tal vértice. Esses operadores são uma extensão dos operadores tradicionais que veremos adiante. Eles dão maior flexibilidade para mudar a quantidade de pontos entre níveis sem precisar mudar a estrutura da malha.

Os operadores de *handlebody* permitem cortar e colar pedaços de superfícies. São eles: $f = \text{create}(p0, p1, p2)$ (cria uma nova face f a partir dos pontos $p0$, $p1$ e $p2$ de S); $(p0, p1, p2) = \text{destroy}(f)$ (destrói uma face existente e retorna os seus três vértices); $e = \text{glue}(he0, he1)$ (“identifica” duas semi-arestas de bordo e transforma-as numa aresta de interior e retorna tal aresta), e $(he1, he2) = \text{unglue}(e)$ (divide uma aresta de interior em duas semi-arestas de bordo e as retorna).

Os operadores estelares permitem mudar a resolução (quantidade de pontos na malha) e a estrutura combinatória da malha. São eles: $e = \text{flip}(e)$ (faz um *swap* na aresta e e retorna a mesma aresta); $v = \text{split}(f)$ (trsecta a face f e retorna um novo vértice que é adicionado a S); $f = \text{weld}(v)$ (operador inverso dos operadores de *split*, retorna uma face e remove um vértice de S). Note que o operador flip está definido somente para arestas internas.

Sejam Δ um operador topológico e (S, K) . Denotamos por $\Delta(S, K)$ como a *ação* de Δ sobre (S, K) , que também gera um par. Ao todo, temos nove operadores topológicos sobre pare e podemos observar claramente que eles são invertíveis. Mais precisamente, dizemos que Δ^{-1} é o *inverso* de Δ se $\Delta^{-1} \circ \Delta = \Delta \circ \Delta^{-1} = Id$, onde Id é o operador *identidade*. Todos os operadores topológicos são invertíveis. Assim, temos: $\text{add}^{-1} = \text{remove}$, $\text{destroy}^{-1} = \text{create}$, $\text{glue}^{-1} = \text{unglue}$, $\text{flip}^{-1} = \text{flip}$ e $\text{split}^{-1} = \text{weld}$.

Não existe a propriedade comutativa entre operadores. De fato, se tomarmos por exemplo $\text{split} \circ \text{create}(p0, p1, p2)$, não tem como o operador split comutar com create pois há dependência de argumentos. Nesse caso é necessário que a face pré exista para que o operador split atue.

Dentre os operadores topológicos somente dois aumentam a resolução de uma malha: create e split . A diferença entre ambos é que o operador create depende de vértices que estão em S e necessita ser precedido do operador add . Isto não ocorre com o operador split que depende apenas de uma face que já está em K . De modo análogo podemos raciocinar quando há diminuição de resolução.

As definições a seguir formalizam as operações de *junção* e *disjunção* topológicas

sobre malhas encaixantes.

Definição 6. Sejam (S_0, K_0) e (S_1, K_1) . Dizemos que $(S_1, K_1) = (S_0, K_0) \oplus s$ é a *junção topológica* de um ponto s com (S_0, K_0) se $S_1 = S_0 \cup \{s\}$ e existe uma sequência de operações topológicas Δ_i tais que

$$\Delta_n \circ \Delta_{n-1} \dots \circ \Delta_1(S_0, K_0) = (S_1, K_1).$$

Definição 7. Sejam (S_0, K_0) e (S_1, K_1) . Dizemos que $(S_1, K_1) = (S_0, K_0) \ominus s$ é a *disjunção topológica* de um ponto s com (S_0, K_0) se $S_1 = S_0 - \{s\}$ e existe uma sequência de operações topológicas Δ_i tais que

$$\Delta_n \circ \Delta_{n-1} \dots \circ \Delta_1(S_0, K_0) = (S_1, K_1).$$

Segue diretamente da invertibilidade de operadores que a junção e disjunção topológica são também invertíveis. Mais precisamente:

Propriedade 1. $(S_1, K_1) = (S_0, K_0) \oplus s \Rightarrow (S_0, K_0) = (S_1, K_1) \ominus s$. De outra forma:

$$\Delta_n \dots \circ \Delta_1(S_0, K_0) = (S_1, K_1) \Rightarrow \Delta_1^{-1} \dots \circ \Delta_n^{-1}(S_1, K_1) = (S_0, K_0).$$

Da não comutatividade de operadores segue que a ordem de junção topológica de pontos influencia diretamente no resultado final da malha.

5 Amostragens por Discos de Poisson

Nesta seção vamos tratar de Amostragens por Discos de Poisson (ADP), uma importante classe de amostragem estocástica bastante usada em aplicações de Computação Gráfica. Por motivos didáticos definiremos ADP's para dois tipos de domínios: todo o plano \mathbb{R}^2 e regiões $R \in \mathbb{R}^2$ que sejam abertas, limitadas e conexas. Vamos também analisar sua relação geométrica e topológica com os alfa complexos sólidos.

5.1 ADP's em \mathbb{R}^2

Definição 8. Seja $S_\alpha = \{s_1, s_2, \dots\}$ uma amostragem do plano. Dizemos que S_α é uma Amostragem por Discos de Poisson se (i) $\cup_{s_i \in S_\alpha} B_\alpha(s_i) = \mathbb{R}^2$ e, além disso, (ii) $S_\alpha \cap B_\alpha(s_i) = \{s_i\}, \forall i$. À condição (i) denominaremos de *condição de cobertura* e à condição (ii) denominaremos de *condição de Poisson*.

Proposição 2. Existe uma ADP no plano.

Demonstração. Exemplos triviais de ADP's são os *lattices* regulares como o quadrangular (figura 2.a) e o triangular (figura 2.b). Exemplos mais elaborados podem ser criados por meio da abordagem algorítmica *dart throwing* [19]. Nesta abordagem tem-se um gerador aleatório de amostras na região e um validador, que verifica se elas satisfazem os critérios geométricos desejados. No nosso caso o critério é a condição de Poisson. Se uma amostra for validada então ela é incorporada à saída, caso contrário a descartamos. O critério de parada do algoritmo seria a condição de cobertura o que logicamente não é possível nesse caso o porque existe uma infinidade de pontos e, portanto, só é aplicável para domínios limitados que é o que veremos na próxima definição. Este algoritmo é um exemplo típico de amostragem estocástica também conhecido como ruído azul [20]. Veja por exemplo a figura 2.c. \square

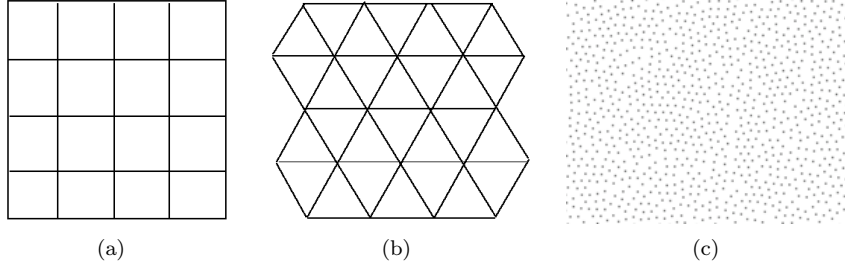


Figura 2: Exemplos de ADP's: (a) lattice quadrangular, (b) lattice triangular, (c) amostragem estocástica.

A seguir estabeleceremos a relação entre uma ADP seu alfa complexo sólido.

Proposição 3. : O Alfa Complexo Sólido de uma ADP S_α do plano, $\overline{C_\alpha(S_\alpha)}$, é uma cobertura do plano.

Demonstração. Basta mostrar que todo triângulo $\sigma \in DT(S_\alpha)$ satisfaz $r_\sigma < \alpha$ onde r_σ é o raio do círculo circunscrito de σ . Seja c_σ o centro do círculo circunscrito de σ . Se, por absurdo, $r_\sigma > \alpha$ então diretamente da propriedade de Delaunay teríamos $d(c_\sigma, S_\alpha) > \alpha$ e, portanto, $c_\sigma \notin \cup_{s_i \in S_\alpha} B_\alpha(s_i)$. \square

A proposição acima tem como consequência dois fatos importantes. A primeira é a qualidade da subdivisão planar em triângulos. Existe um limite superior para a razão de aspecto dada por $\text{aspect-ratio}(\sigma) = \frac{L^2}{\text{vol}(\sigma)}$ onde $\text{vol}(\sigma)$ é a área do triângulo σ e L é o comprimento da maior aresta de σ . No trabalho de Medeiros [7] onde prova-se que $\frac{L^2}{\text{vol}(\sigma)} \leq 4\sqrt{3}$ e a igualdade ocorre quando o maior ângulo é $2\pi/3$.

A segunda consequência é o poder de controle da escala dos elementos simpliciais. É fácil ver que pelas condições de amostragem os raios dos circuncírculos dos triângulos e os comprimentos das arestas estão variando no intervalo $[\frac{\alpha}{2}, \alpha]$. Daí, fica natural em pensarmos em multiresolução para representação de regiões sólidas.

5.2 ADP's em Regiões do Plano

Vamos definir amostragem por discos de Poisson para uma classe regiões do plano com bordo. A fronteira têm importância crucial: é ela quem define tanto a forma como a topologia da região.

Definição 9. Seja $S_\alpha = \{s_1, s_2, \dots, s_n\}$ uma amostragem de uma região sólida R ($R = A \cup \partial A$, A aberto e conexo). Dizemos que S_α é uma amostragem por discos de Poisson (ADP) se $R \subset \cup_{s_i \in S_\alpha} B_\alpha(s_i)$ (condição de cobertura) e, além disso, $S_\alpha \cap B_\alpha(s_i) = \{s_i\}$, $\forall i$ (condição de Poisson).

Observe que diferentemente da definição anterior a igualdade não ocorre na condição de cobertura. No entanto, dependendo do raio da amostragem¹ podemos aproximá-la da região tanto quanto se queira. É o que veremos na proposição 7.

¹Nos referiremos sempre ao parâmetro α da amostragem por discos de Poisson.

Assim como no plano, existem ADP's para regiões. Com a mesma idéia de gerar uma ADP usando o algoritmo de *dart throwing* temos também a existência de uma infinidade delas.

Daqui em diante ao nos referirmos a uma região sólida R ela será aberta, conexa, limitada, a menos que seja dito explicitamente.

Na definição abaixo, cada amostra de uma ADP de uma região R pode ser classificada segundo sua topologia.

Definição 10. Seja S uma amostragem de uma região R . Dizemos que $s \in S_\alpha$ é uma amostra de *bordo* (*interior*) se $s \in \partial R$ ($s \in \text{int}(R)$).

Quanto às técnicas de gerar uma ADP de uma região R podemos classificá-las como *aproximantes* ou *interpolantes*. A única diferença entre essas duas amostragens é que as interpolantes levam em consideração o bordo.

Definição 11. Seja $S_\alpha = \{s_1, s_2, \dots, s_n\}$ uma amostragem de uma região sólida R . Dizemos que S_α é uma ADP *interpolante* se existe $A \subset S_\alpha$ tal que $A \subset \partial R$ e $\partial R \subset \cup_{s_i \in A} B_\alpha(s_i)$. Vamos denotar uma ADP interpolante por \tilde{S}_α .

Note que usando novamente *dart throwing* ao restringirmos primeiramente o seu alvo ao bordo e depois amostrarmos o interior obtemos uma ADP de R interpolante de forma algoritmica.

A definição 9 nos remete a questões sobre aproximação geométrica do bordo e da topologia da região pelo alfa complexo sólido para um determinado α . Mas primeiramente vamos fazer algumas definições importantes que servirão de ferramentas para enunciá-las e demonstrá-las.

Definição 12. A *distância potencial* de um ponto $x \in \mathbb{R}^2$ de uma bola b é dada por $\pi_b(x) = \|x - c_b\|^2 - r_b^2$ onde c_b e r_b são o centro e a raio de b , respectivamente.

Uma observação importante é que um ponto $x \in \mathbb{R}^2$ pertence a bola se e só se $\pi_b(x) \leq 0$, e ele pertence ao bordo da bola se e só se $\pi_b(x) = 0$. Dado um conjunto finito de bolas \mathcal{B} , podemos dividir o espaço em regiões:

Definição 13. A *região potencial de Voronoi* de uma bola $u \in \mathcal{B}$ é o conjunto de pontos do plano para os quais u minimiza a distância potencial,

$$V_u = \{x \in \mathbb{R}^2 \mid \pi_u(x) \leq \pi_v(x), \forall v \in \mathcal{B}\}$$

O diagrama formado pelas regiões potenciais de Voronoi tem sido chamado na literatura de *diagrama potencial* para distinguir do diagrama de Voronoi. Não é difícil mostrar que o conjunto de pontos igualmente distantes de duas bolas u e v é um hiperplano definido por $\pi_u = \pi_v$. No contexto em que estamos onde consideramos os discos de uma ADP, temos bolas de mesmo raio e, portanto, o diagrama potencial coincide com o diagrama de Voronoi. Observe que as regiões potenciais de Voronoi decompõem a união de bolas de uma ADP em regiões convexas do tipo $V \cap u$, como ilustrado na figura 4.

Definição 14. O *complexo dual* K de S é a coleção de simplexes

$$K = \{\text{conv}\{u_c \mid u \in T\} \mid T \subseteq S, \cap_{u \in T} (u \cap V_u) \neq \emptyset\}.$$

Proposição 4. O complexo dual de uma ADP S_α de uma região R é um alfa complexo.

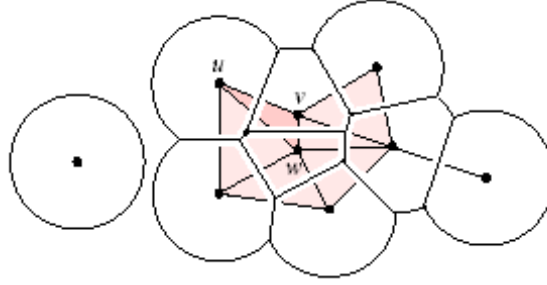


Figura 3: União de nove bola, decomposição convexa usando regiões potenciais de Voronoi e seu complexo dual.

Demonstração. Veja [15]. □

Temos então uma maneira diferente de definir o alfa complexo de um conjunto de pontos associados a esferas, diferente do que foi definido em 5. Dependendo do objetivo a ser alcançado pode-se usar a definição mais apropriada.

Agora, vamos definir algumas ferramentas que nos permitirão chegar a uma conclusão sobre a topologia dos alfa complexos. Na verdade, queremos descobrir que tipo de relação topológica existente entre a união das bolas de uma ADP e seu o seu alfa complexo dual.

Definição 15. Uma *deformação por retração* de um espaço \mathbb{X} em um subespaço \mathbb{A} é uma família de mapas $f_t : \mathbb{X} \rightarrow \mathbb{A}, t \in [0, 1]$ tal que f_0 é a identidade, $f_1(\mathbb{X}) = \mathbb{A}$ e $f_t|_{\mathbb{A}}$ é a identidade, para todo t . A família deve ser contínua, de tal forma que o mapeamento associado $\mathbb{X} \times [0, 1] \rightarrow \mathbb{X}, (x, t) \mapsto f_t(x)$ é contínuo.

Em outras palavras, começando do espaço original \mathbb{X} no tempo 0, deformamos continuamente o espaço até transformá-lo no subespaço \mathbb{A} no tempo 1. Uma deformação por retração é um tipo especial de homotopia, onde a exigência do espaço final ser um subespaço é relaxada.

Definição 16. Uma *homotopia* é uma família de mapas $f_t : \mathbb{X} \rightarrow \mathbb{Y}, t \in [0, 1]$, tal que seu mapeamento associado $F : \mathbb{X} \times [0, 1] \rightarrow \mathbb{Y}$ dado por $F(x, t) = f_t(x)$ é contínuo. Assim, $f_0, f_1 : \mathbb{X} \rightarrow \mathbb{Y}$ são *homotópicos* via homotopia f_t . Denotaremos isto como $f_0 \simeq f_1$.

Suponha que tenhamos a retração como na definição 15. Se considerarmos $i : \mathbb{A} \rightarrow \mathbb{X}$ inclusão, temos que $f_1 \circ i \simeq id$ e $i \circ f_1 \simeq id$. Isto nos permite classificar \mathbb{X} e seu subespaço \mathbb{A} como tendo a mesma conectividade. Isto é o um caso especial de equivalência homotópica.

Definição 17. Uma mapeamento $f : \mathbb{X} \rightarrow \mathbb{Y}$ é chamado de *equivalência homotópica* se existe um mapeamento $g : \mathbb{Y} \rightarrow \mathbb{X}$, tal que $f \circ g \simeq id$ e $g \circ f \simeq id$. Assim, \mathbb{X} e \mathbb{Y} são *homotópicos equivalentes* e têm o mesmo *tipo de homotopia*. Este fato é denotado por $\mathbb{X} \simeq \mathbb{Y}$.

Agora podemos enunciar um importante resultado sobre homotopia entre dois espaços que conhecemos.

Proposição 5. Sejam S_α uma ADP de uma região R e $\mathcal{B} = \cup_{s_i \in S_\alpha} B_\alpha(s_i)$. Então $\mathcal{B} \simeq |C_\alpha(S_\alpha)|$.

Demonstração. Não é difícil ver que idéia da demonstração é exibir uma retração que leve o espaço \mathcal{B} no espaço $|C_\alpha(S_\alpha)|$. Veja [15] para maiores detalhes. \square

A relevância da proposição está na invariância de *homologia* entre os dois espaços. Esse tipo de relação não preserva a dimensão intrínseca uma vez que os alfa complexos podem ter simplexes isolados de dimensões menores. Mas existe uma relação mais forte que é o *homeomorfismo* e nesse caso há coincidência de topologia.

Definição 18. Um *homeomorfismo* $f : \mathbb{X} \rightarrow \mathbb{Y}$ é uma função bijetora, tal que f e f^{-1} são contínuas. Dizemos que \mathbb{X} é *homeomorfo* a \mathbb{Y} , $\mathbb{X} \approx \mathbb{Y}$ e que \mathbb{X} e \mathbb{Y} têm o mesmo *tipo de topologia*.

Para ter alguma garantia de equivalência topológica entre a região e o alfa complexo sólido vamos usar a idéia de *eixo medial* e *LFS (local feature size)*.

Definição 19. O *eixo medial* de uma curva F é o fecho de um conjunto de pontos no plano que tem dois ou mais pontos mais próximos em F .

Definição 20. A *Local Feature Size, LFS(p)*, de um ponto $p \in F$ é a distância euclidiana de p ao ponto mais próximo m do eixo medial.

Na proposição abaixo destacamos uma condição para ADP's de regiões sólidas com bordo suave² para que o complexo dual tenha o mesmo tipo de topologia da região.

Proposição 6. Se o raio de uma ADP S_α de uma região R (com bordo suave) for menor que $\frac{1}{2} \inf_{p \in \partial R} LFS(p)$ então $|C_\alpha(S_\alpha)| \approx R$.

Demonstração. Veja Medeiros [8]. \square

Quanto à proximidade geométrica de ∂R e $\partial C_\alpha(S_\alpha)$ podemos estabelecer uma convergência global a partir do parâmetro α .

Proposição 7. Sejam $\alpha > 0$ e S_α uma ADP de R . A distância de Hausdorff $d_H(R, \partial |C_\alpha(S_\alpha)|)$ da região R com a região coberta pelo alfa complexo de S_α é menor que α .

Demonstração. Basta notar que se $\forall p \in R$ então existe $q \in S_\alpha$ tal que $p \in B_\alpha(q)$ e, portanto, $d_H(p, |C_\alpha(S_\alpha)|) < \alpha$. \square

Quanto aos algoritmos para calcular alfa complexos pode-se usar a triangulação de Delaunay como auxílio. Mas de posse da proposição 6 podemos usar o algoritmo Ball-Pivoting restrito ao plano (RBPA) [6] para construir o alfa complexo sólido em tempo linear ao número de pontos sem que seja necessário calcular a triangulação de previamente Delaunay.

Proposição 8. Seja $S \subset \mathbb{R}^2$ um conjunto de pontos em posição geral e $\overline{C_\alpha(S)}$ seu alfa complexo sólido. Considere \mathcal{T}_α como a saída do RBPA sendo executado no plano. Então $\overline{C_\alpha(S)} = \mathcal{T}_\alpha$.

²Esta condição é pra garantir que existe $\epsilon > 0$ tal que $LFS(p) > 0, \forall p \in \partial R$.

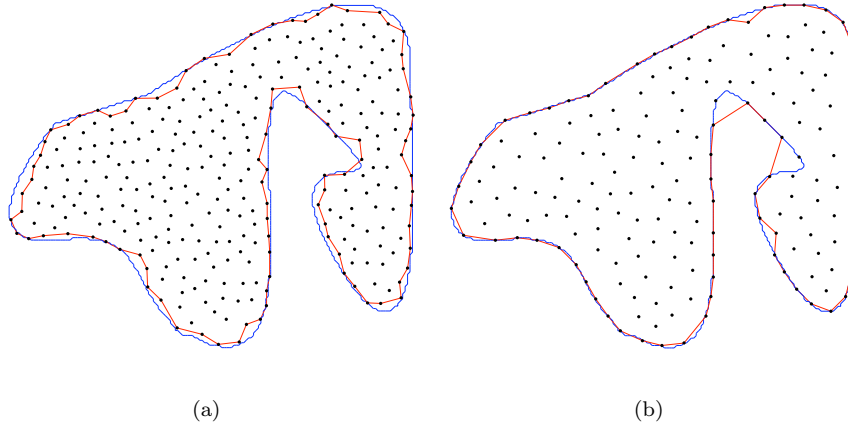


Figura 4: Exemplos de reconstruções de amostragem aproximante (a) e amostragem interpolante (b).

Demonstração. Veja Medeiros et al. [6]. □

Assim, dado α como na proposição 6, pela proposição anterior, calculamos o alfa complexo sólido com o mesmo tipo de topologia da região.

Como visto anteriormente, há restrições quanto ao conjunto de objetos sólidos para os quais existe de $\alpha > 0$. De fato, ao exigirmos que o bordo da região seja suave, é porque não queremos que ocorram “bicos” (*sharp features*) pois teríamos $\inf_{p \in \partial R} LFS(p) = 0$ e não há como determinar uma cota superior para α . Estamos trabalhando para estender as garantias de reconstrução do sólido da proposição 6 para regiões com um conjunto finito de bicos.

Um fato importante de se observar é que os complexos duais de ADP’s aproximantes são mais ruidosos na fronteira do que os complexos duais de ADP’s interpolantes (veja figura 4). Faremos uma maior explanação desse fato na seção 7.

6 Multiresolução com ADP’s

Feita a análise de uma só ADP de uma região, a partir da próxima seção vamos começar a interagir as diversas ADP’s usando as escalas dadas pelo parâmetro α .

6.1 Família Escalonada

Para uma determinada região R existe uma infinidade de ADP’s dependendo de α e da aleatoriedade do algoritmo que gera a amostragem. Vamos introduzir o conceito de *Família Escalonada* que é um caso particular de família de pontos encaixantes. Tal conceito tem o propósito estruturar um conjunto de ADP’s de tal forma que eles sejam encaixantes a fim de nos dar a idéia de amostragem em multiresolução com mudança de escala.

Definição 21. Seja $\mathcal{F} = \{S_{\alpha_i}\}_{i \in \{1, 2, \dots, n\}}$ uma família de ADP’s de uma região R . Dizemos que \mathcal{F} é *escalorada* se $S_{\alpha_1} \subset S_{\alpha_2} \dots \subset S_{\alpha_n}$, com $\alpha_1 > \alpha_2 > \dots > \alpha_n$.

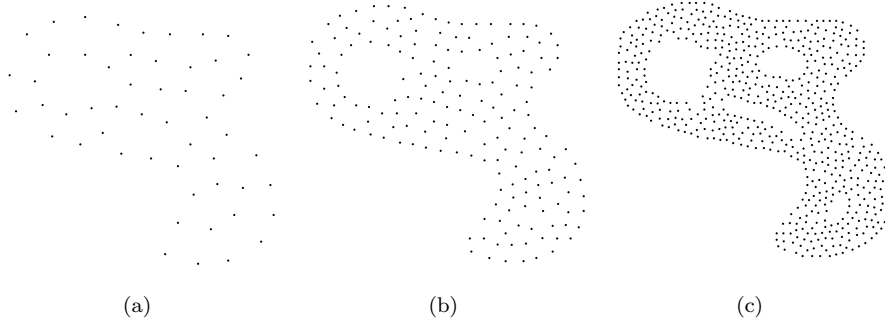


Figura 5: Exemplo de família escalonada com três níveis.

A cada um dos i 's denominamos de *níveis de escala* e seus respectivos α_i 's de *escalas*.

Para um conjunto de reais positivos $\alpha_1 > \alpha_2 > \dots > \alpha_n$, descrever um algoritmo de que gera uma família escalonada é bem simples: Dada a amostra S_{α_k} basta aplicar o algoritmo *dart throwing* em $(R - \cup_{s_i \in S_{\alpha_k}} B_{\alpha_{k+1}}(s_i))$. Na figura 5 temos um exemplo de família escalonada com três níveis de escala.

Um caso especial de família escalonada ocorre quando a diferença de resolução entre os níveis é de apenas um ponto.

Definição 22. Seja $\mathcal{F} = \{S_{\alpha_i}\}_{i \in \{1, 2, \dots, n\}}$ uma família escalonada de R . Dizemos que F é *cadenciada* se $S_{\alpha_{i+1}} = S_{\alpha_i} \cup \{s_i\}$ com $\alpha_{i+1} < \alpha_i, \forall i$.

Se for possível, a partir de uma família escalonada, gerar uma família cadenciada então dizemos que tal família é *cadenciável*.

Uma pergunta que surge diretamente da definição acima é: toda família escalonada é cadenciável?. A resposta é negativa e temos um contra exemplo pra isso na figura 6. Em 6.a temos um nível de ADP com raio $\frac{\sqrt{2}}{2} < \alpha < 1$ de um quadrado de lado 1. Em 6.b temos um nível em que os pontos estão sobre um grid de espaçamento $\frac{1}{3}$ com perturbação. Em destaque exibimos o melhor candidato à cadência. Em 6.c mostramos geometricamente a impossibilidade de satisfazer a condição de cobertura.

Apesar de não ser possível cadenciar uma família escalonada, afirmamos que é sempre possível para uma definição mais fraca de ADP's que são as (α, β) -ADP's. É o que veremos na seção 7.

Se não há condições suficientes de cadenciar uma família escalonada tampouco há garantias de gerar uma família cadenciada utilizando o mesmo algoritmo que gera uma família escalonada, dados $\alpha_1 > \alpha_2 > \dots > \alpha_n$ reais positivos. Isso acontece porque as escalas já estão determinadas, daí, como há a limitação de gerar apenas uma amostra entre dois níveis, a condição de cobertura pode não ser satisfeita. De certa maneira existem "saltos" ou descontinuidades entre os níveis consecutivos da família dadas pelas escalas. A boa notícia é que, dado um objeto sólido, podemos gerar uma família cadenciada sem as escalas pré-determinadas. Para facilitar o entendimento do gerador de família cadenciada, vamos definir família cadenciada com escalas contínuas.

Definição 23. Seja $\mathcal{F} = \{S_{\alpha_t}\}_{t \in [a, b]}$ uma família de ADP's de R . Dizemos que F é *cadenciada continuamente* se existem $b = \alpha_1 > \alpha_2 > \dots > \alpha_n = a$ tais que

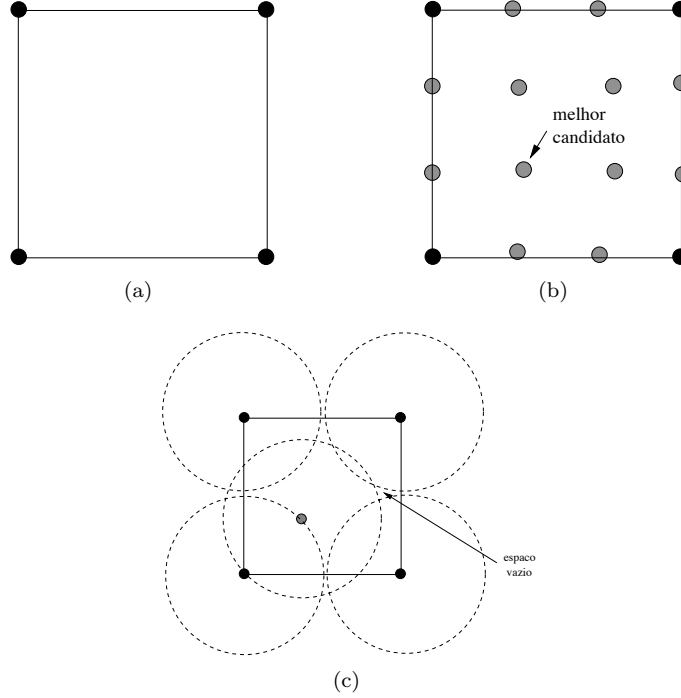


Figura 6: Contra exemplo para cascadiamento.

$S_{\alpha_{i+1}} = S_{\alpha_i} \cup \{s_i\}$ e se $s \in (\alpha_{i+1}, \alpha_i)$ então $S_s = S_{\alpha_i}$.

É mais natural, a partir da definição acima, pensar num algoritmo que gera uma família cascadiada continuamente, dados $b > a > 0$. Para tanto, começamos com uma ADP de raio $b > 0$ da região R . Agora defina $f : [a, b] \rightarrow \mathbb{R}$ tal que $f(t) = \text{vol}(R - \mathcal{B}_t)$, com $\mathcal{B}_t = \cup_{s_i \in S_{\alpha_t}} B_{\alpha_t}(s_i)$. Observe que inicialmente temos $f(b) = 0$ até certo instante $t = t_0$ quando então a função f assumirá valores crescentes. Nesse instante t_0 um buraco “infinitesimal” foi aberto e a condição de cobertura foi desfeita para discos de raio t_0 . Mas ao inserirmos um novo ponto p nesse buraco, teremos tanto a condição de Poisson como a de cobertura satisfeitas para o mesmo raio. Repetimos então o mesmo processo até chegar ao menor raio possível a .

6.2 Família de Malhas Escalonadas

As família de malhas escalonadas que introduziremos a seguir têm propriedades que estão associadas diretamente a uma família escalonada. Elas são um caso particular de uma família de malhas encaixantes.

Definição 24. Seja S_α uma ADP de uma região R e K_α um complexo simplicial sólido. Se $K_\alpha^0 \subset S_\alpha$ pomos (S_α, K_α) (leia-se por simplicidade “ α -par”).

Definição 25. Seja $\mathcal{F} = \{S_{\alpha_i}\}_{i \in \{1, 2, \dots, n\}}$ uma família escalonada de uma região R . Dizemos que $\mathcal{M}(\mathcal{F}) = \{(S_{\alpha_i}, K_{\alpha_i})\}_{i \in \{1, 2, \dots, n\}}$ é uma *família de Malhas Escalonadas* se $(S_{\alpha_i}, K_{\alpha_i})$. Assim como em famílias escalonadas, a cada um dos i 's denominamos de *níveis de escala* e seus respectivos α_i 's de *escalas*.

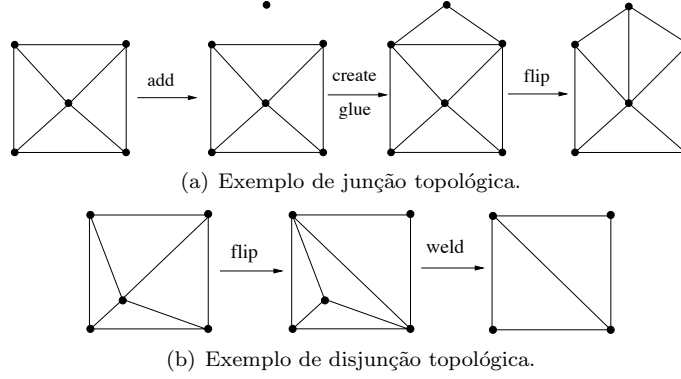


Figura 7: Exemplos de inserção e remoção de pontos.

Exemplo 1. Seja $F = \{S_{\alpha_i}\}_{i \in \{1, 2, \dots, n\}}$ uma Família Escalonada. Dizemos que $C_{\alpha}\mathcal{F} = \{(S_{\alpha_i}, C_{\alpha_i}(S_{\alpha_i}))\}_{i \in \{1, 2, \dots, n\}}$ é uma família de *Alfa Complexos Sólidos Escalonada* se $C_{\alpha}(S_{\alpha_i}) = \{\text{Alfa Complexo Sólido de } S_{\alpha_i}\}$.

Neste exemplo temos uma estrutura bem mais forte: além de vértices em comum, pelas propriedades P1, P2 e P3 da seção 2.3, há também um subconjunto de faces coincidentes entre níveis. Vamos explorar esse exemplo posteriormente quando falarmos de filtragem por operadores topológicos.

6.3 Filtragem por Operadores Topológicos

No contexto de ADP's, inserir pontos numa malha consiste em acrescentar mais detalhes. Se o ponto for de bordo, teremos melhor definição da fronteira e se for de interior teremos uma mudança na estrutura combinatória com um acréscimo na quantidade de triângulos. De forma análoga também podemos remover um ponto de uma malha que consiste em perder detalhes. Se o ponto for de bordo, teremos menor definição da fronteira e se for de interior haverá decréscimo na quantidade de triângulos. Como visto anteriormente a inserção e a remoção de pontos equivalem respectivamente a junção e a disjunção topológica. Na figura 7 exibimos exemplos de inserção e remoção de pontos por meio de uma sequência de operadores topológicos.

Tendo em mãos operações de junção ou disjunção topológicas de pontos podemos definir uma sequência de $(S_{\alpha_i}, K_{\alpha_i})$ cadenciada.

Definição 26. *Filtragem por Operadores Topológicos* de uma região R é uma família $\mathcal{M}(\mathcal{F}) = \{(S_{\alpha_i}, K_{\alpha_i})\}_{i \in \{1, 2, \dots, n\}}$ de malhas escalonadas de R tal que $\{S_{\alpha_i}\}$ é cadenciada e $(S_{\alpha_{i+1}}, K_{\alpha_{i+1}}) = (S_{\alpha_i}, K_{\alpha_i}) \oplus s_i$ com $s_i = S_{\alpha_{i+1}} - S_{\alpha_i}$.

Definição 27. Algoritmo por *refinamento (simplificação)* de uma filtragem por operadores topológicos é aquele que gera operações topológicas através de uma sequência de junção (disjunção) topológica de pontos.

Teorema 1. Para toda família de Alfa Complexos Sólidos Escalonada e Cadenciada $C_{\alpha}\mathcal{F} = \{(S_{\alpha_i}, C_{\alpha_i}(S_{\alpha_i}))\}_{i \in \{1, 2, \dots, n\}}$ existe um algoritmo por refinamento e um algoritmo por simplificação que geram um filtragens por operadores topológicos de $C_{\alpha}\mathcal{F}$ cujas operações topológicas de refinamento são inversas às operações topológicas de simplificação.

Demonstração. Exibiremos os dois algoritmos e depois mostraremos que as sequências geradas são inversas.

Para cada algoritmo vamos utilizar algumas estruturas de dados auxiliares. Em $DT(S_\alpha)$ as faces possuem um atributo booleano f (f de filtragem) que indicará se estas pertencem ou não ao alfa complexo sólido corrente. Temos uma lista de operadores que armazena a saída da filtragem L_f .

Algoritmo de refinamento: A idéia é usar uma adaptação do algoritmo de construção de uma triangulação de Delaunay por inserção de pontos [21]. Usaremos uma fila de prioridades de *half-edges* L_{he} que mantém a ordem em que estas serão flipadas³. A chave de prioridade é dada pela potência com sinal trocado do novo ponto a ser inserido s_i com o círculo circunscrito a $he.mate.face.f$, para todo $he \in L_{he}$. Se durante a construção de L_{he} alguma *half-edge* he tiver chave não negativa então ela não será incluída em L_{he} . Invés disso comparamos o raio do círculo circunscrito de $he.face$ com α_i . Caso seja maior e $he.face.f = verdadeiro$ então geramos a operação `destroy` e atualizamos $he.face.f$ para `falso`. Caso seja menor e $he.face.f = falso$ então geramos a operação `create` e atualizamos $he.face.f$ para `verdadeiro`. A este teste denominamos `test_triangle_ref`.

Em cada inserção de ponto s_i primeiramente testamos se este cai fora de $Del(S_{\alpha_{i-1}})$ ou dentro de alguma face de $DT(S_{\alpha_{i-1}})$. Caso esteja dentro de alguma face σ tal que $\sigma.f = verdadeiro$ então geramos um `split` (II.R). O três novos triângulos vão herdar o atributo `verdadeiro`. Antes de inserir três *half-edges* he_1, he_2, he_3 em L_{he} fazemos um teste de compatibilidade seguindo a ordem dada pela chave dessas *half-edges*. Este teste consiste em analisar os atributos booleanos f de faces adjacentes em uma *half-edge* he candidata a flip e seguir regras. A idéia é que quando ambos os atributos são verdadeiros então geramos um flip o que não ocorre quando ambos são falsas. Caso tenham atributos opostos então fazemos essa compatibilidade. Se $he.mate.face.f = verdadeiro$ então geramos o operador `create` para $he.face$ e atualizamos $he.face.f$ para `verdadeiro`. Analogamente, se $he.mate.face.f = falso$ então geramos o operador `destroy` para $he.face$ e atualizamos $he.face.f$ para `falso`. A essa função de compatibilização denominaremos `make_compatibility_ref` (V.R). Veja figura 8 e algoritmo 3. A cada flip existem sempre duas novas *half-edges* a serem inseridas em L_{he} para as quais fazemos a compatibilização seguindo a ordem no sentido anti-horário. É importante frisar que esse teste é feito sempre antes da inserção de novas *half-edges* na fila. Se alguma *half-edge* he não for inserida então fazemos o `test_triangle_ref` em $he.face$.

Caso esteja o ponto s_i dentro de alguma face de $DT(S_{\alpha_{i-1}})$ tal que $\sigma.f = falso$ então não geramos um operador `split`. Invés disso geramos `add` (III.R). O algoritmo procede então da mesma forma realizando operações flip e testes de compatibilidade.

Caso o ponto s_i esteja fora de $DT(S_{\alpha_{i-1}})$ então é gerada uma operação `add` para adicionar o ponto (IV.R). Faces novas são inseridas com atributo f `falso` por meio de operações `create` seguindo a ordem do sentido anti-horário das arestas visíveis ao ponto s_i (*loop* da linha 11 do algoritmo 1). Ao mesmo tempo, são feitas operações de compatibilidade e atualizada fila de prioridades L_{he} . O algoritmo procede então realizando operações flip com testes de compatibilidade.

³Neologismo que significa que uma aresta sofreu uma operação de flip.

Algorithm 1: Algoritmo de refinamento: inserindo um ponto

Entrada: $s_i, \alpha_i, DT(S_{\alpha_{i-1}})$

```
1 if  $s_i \in DT(S_{\alpha_{i-1}})$  then
2    $\text{split}(\sigma, s_i)$  ;
3   for cada nova face F do
4      $F.f = \sigma.f$  ;
5   if  $\sigma.f$  then
6     gerar  $\text{split}(\sigma, s_i)$  ;
7   for  $he = he_1, he_2, he_3 \in \sigma$  do
8     \ * a função  $\text{apply\_power}$  está descrita no algoritmo 2 * \
9      $\text{apply\_power}(s_i, he.next)$  ;
10 else
11   for cada he visível a  $s_i$  em  $\partial DT(S_{\alpha_{i-1}})$  do
12      $f = \text{create}(s_i, he.org, he.dst)$  ;
13      $f.f = \text{false}$  ;
14      $\text{apply\_power}(s_i, he)$  ;
15 while  $L_{he} \neq \emptyset$  do
16    $he \leftarrow L_{he}.top$  ;
17    $key \leftarrow he.key$  ;
18    $L_{he}.pop$  ;
19   if  $key > 0$  then
20     if  $he.mate.face.f$  e  $he.face.f$  then
21       gerar  $\text{flip}(he)$  ;
22        $\text{flip}(he)$  ;
23        $\text{apply\_power}(s_i, he.next)$  ;
24        $\text{apply\_power}(s_i, he.mate.prev)$  ;
25   else
26      $\text{test\_triangle\_ref}(he.face)$  ;
27 for cada  $\sigma \in DT(S_{\alpha_i})$  do
28    $\text{test\_triangle\_ref}(\sigma)$ ;
```

Por fim, ao encerrarem-se as possíveis operações na vizinhança de s_i pecorremos as outras faces de $DT(S_{\alpha_i})$ e fazemos o `test_triangle_ref` em cada uma delas. A ordem de chamada é dada pelo comprimento dos raios dos círculos circunscritos de cada face, começando do menor para o maior (I.R)(*loop* da linha 27 o algoritmo 1). No algoritmo 1 temos o pseudo-código que resume todos os passos descritos para a inserção de um ponto.

Algoritmo de simplificação: A idéia é usar um algoritmo de remoção de pontos [18] e adaptá-lo para o que queremos. O algoritmo de remoção deve retirar todos os triângulos incidentes a s_i e retriangular no “sentido de Delaunay” o polígono estrelado $H = \{q_0, q_1, \dots, q_{k-1}, q_k = q_0\}$ criado por estas retiradas. Três vértices consecutivos $q_i q_{i+1} q_{i+2}$ ao longo do bordo de H são ditos que formam uma *orelha* se o segmento $q_i q_{i+2}$ está dentro de H . Uma orelha de H é dita de

Algorithm 2: função *apply_power*

Entrada: s_i , half-edge he

```
1 if not  $he.edge.is\_bdry$  then
2   key  $\leftarrow$  power( $s_i, he.mate.face$ ) ;
3   if  $key > 0$  then
4     make_compatibility( $he$ ) ;
5      $L_{he}.insert(he, key)$  ;
6   else
7     test_triangle( $he$ ) ;
8 else
9   test_triangle( $he$ ) ;
```

Algorithm 3: make_compatibility_ref

Entrada: half-edge he

```
1 if  $he.mate.face = verdadeiro$  e  $he.face = falso$  then
2   gerar create( $he.face$ ) ;
3    $he.face \leftarrow verdadeiro$  ;
4 if  $he.mate.face = falso$  e  $he.face = verdadeiro$  then
5   gerar destroy( $he.mate.face$ ) ;
6    $he.mate.face \leftarrow falso$  ;
```

Delaunay se o seu círculo circunscrito não contém nenhum vértice de H em seu interior. O algoritmo usa o seguinte lema:

Lema 1. Considere um polígono $H = \{q_0, q_1, \dots, q_k = q_0\}$ e um ponto p tal que as arestas $q_i q_{i+1}$ pertencem à triangulação de Delaunay de $\{q_0, q_1, \dots, q_{k-1}, p\}$. Se $|pot(p, circ(q_i, q_{i+1}, q_{i+2}))|$ (potência de ponto de p em relação ao círculo circunscrito a $q_i q_{i+1} q_{i+2}$) é mínimo, então $q_i q_{i+2}$ é uma aresta de Delaunay da triangulação de $\{q_0, q_1, \dots, q_{k-1}\}$.

Demonstração. Veja [18]. □

Assim como no refinamento, temos uma fila de prioridades de *orelhas* L_{ear} cujos elementos são uma estrutura do tipo *candidate_ear* formada por três *half-edges*: $s_i q_i, s_i q_{i+1}, s_i q_{i+2}$ (veja abaixo a definição da estrutura que será usada nos algoritmos 4 e 5). Desse modo L_{ear} mantém a ordem em que arestas $s_i q_{i+1}$ serão flipadas, sendo que a chave de prioridade é dada pela potência de ponto de s_i (ponto a ser removido) com o círculo circunscrito a $q_i q_{i+1} q_{i+2}$. O Lema acima nos garante que no topo da fila estará uma orelha que pertence à triangulação de Delaunay.

```
struct Candidate_Ear {
  Half_Edge*  $he$  1,  $he$  2,  $he$  3 ;
}
```

Algorithm 4: Algoritmo de simplificação: removendo um ponto

Entrada: s_{i+1} , α_i , $DT(S_{\alpha_{i+1}})$

```
1 for cada  $\sigma \in DT(S_{\alpha_{i+1}} - link(s_{i+1}))$  do
2    $\lfloor$  test_triangle_spl( $\sigma$ );
3   gerar fila de prioridades  $L_{ear}$  ;
4   while ( $L_{ear}.size > 3$  e  $\sigma_i \in DT(S_{\alpha_{i+1}})$ ) ou
   ( $L_{ear} \neq \emptyset$  e  $\sigma_i \in \partial DT(S_{\alpha_{i+1}})$ ) do
5      $candidate\_ear \leftarrow L_{ear}.top$  ;
6      $L_{ear}.pop$  ;
7     if counter_clockwise( $candidate\_ear$ ) then
8       make_compatibility_spl( $candidate\_ear.he_2$ ) ;
9       if  $candidate\_ear.he_2.face.f$  e  $candidate\_ear.he_2.mate.face.f$ 
       then
10         $\lfloor$  gerar flip( $candidate\_ear.he_2$ ) ;
11         $\lfloor$  flip( $candidate\_ear.he_2$ ) ;
12     $L_{ear}.update$  ;
13 if  $\sigma_i \in int(DT(S_{\alpha_{i+1}}))$  then
14    $test \leftarrow test\_triangle\_spl(link(s_{i+1}))$  ;
15   for cada face vizinha  $F$  de  $s_{i+1}$  do
16     if  $test$  e not  $F.f$  then
17        $\lfloor$  gerar create( $F.f$ ) ;
18     if not  $test$  e  $F.f$  then
19        $\lfloor$  gerar destroy( $F.f$ ) ;
20   if  $test$  then
21      $\lfloor$  gerar weld( $\sigma_i$ ) ;
22   weld( $\sigma_i$ ) ;
23 else
24   for cada face vizinha  $F$  de  $\sigma_i$  do
25     if  $F.f$  then
26        $\lfloor$  gerar destroy( $F.f$ ) ;
27      $\lfloor$  destroy( $F.f$ ) ;
```

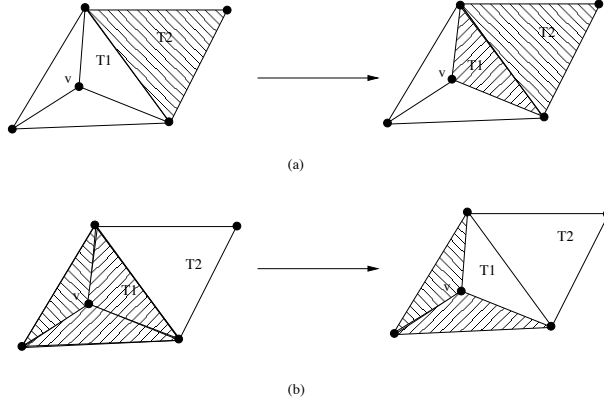


Figura 8: Casos da função `make_compatibility_ref`. v é o ponto inserido por uma operação `split`, T1 e T2 são triângulos a serem compatibilizados. Os triângulos hachurados têm atributo booleano verdadeiro e os não hachurados tem atributo falso. Em (a) e (b) temos os dois possíveis casos tratados pelo procedimento.

Primeiramente, antes de remover um ponto, aplicamos o `test_triangle_spl` em cada uma das faces de $DT(S_{\alpha_{i+1}})$ fora da vizinhança de s_i pela ordem decrescente dos círculos circunscritos (*loop* da linha 1 do algoritmo 4). Neste teste comparamos o raio do círculo circunscrito de uma face σ com α_i . Caso seja maior e $\sigma.f = \text{verdadeiro}$ então geramos a operação `destroy` e atualizamos $\sigma.f$ para `falso`. Caso seja menor e $\sigma.f = \text{falso}$ então geramos a operação `create` e atualizamos $\sigma.f$ para `verdadeiro` (I.S). Note que este teste é equivalente ao teste do algoritmo de refinamento `test_triangle_ref`.

Em seguida, vamos aplicando as operações de `flip` seguindo a ordem da fila de prioridades L_{ear} . Observe que se $s_i \in DT(S_{\alpha_{i+1}})$ então chegaremos em três orelhas. Dada uma orelha $q_i q_{i+1} q_{i+2}$ candidata a ser flipada, antes de a fazê-la, executamos o teste `test_triangle_spl` para verificar se ela estará na triangulação. Sejam σ_1 e σ_2 as duas faces vizinhas a $s_i q_{i+1}$ pela ordem no sentido anti-horário. Dependendo do resultado no teste de $q_i q_{i+1} q_{i+2}$ então verificamos as possíveis compatibilidades de σ_1 e σ_2 através de uma função denominada `make_compatibility_spl` (V.S). Em outras palavras, Esta função analisa as faces σ_1 e σ_2 e as compatibiliza com o resultado no teste de $q_i q_{i+1} q_{i+2}$. Se $\sigma_1.f = \text{falso}$ então geramos um operador `create` e atualizamos $\sigma_1.f$ para `verdadeiro`. Da mesma forma fazemos para σ_2 . Depois geramos um `flip`. Se $q_i q_{i+1} q_{i+2}$ não passar pelo teste `test_triangle_spl` então verificamos se $\sigma_1.f = \text{verdadeiro}$, geramos uma operação de `destroy` e atualizamos $\sigma_1.f$ para `falso`. Também o mesmo é feito para σ_2 . Nesse caso o operador `flip` não é gerado. Veja figura 9.

Quando todos os flips possíveis forem executados, se o vértice s_i estiver no interior de $DT(S_{\alpha_{i+1}})$ então ele terá valência três. Sejam q_1 , q_2 e q_3 os vértices vizinhos a s_i . Se o triângulo $q_1 q_2 q_3$ passar pelo `test_triangle_spl` então as faces vizinhas a s_i que tem o atributo $f = \text{falso}$ vão gerar operadores `create` e terão atualizadas seus atributos para `verdadeiro`. A ordem de gerar tais operadores é dada pela potencia de ponto s_i em relação às faces adjacentes ao triângulo $q_1 q_2 q_3$. Depois disso, geramos um operador `weld` (I.I.S). Se o triângulo $q_1 q_2 q_3$ não passar pelo teste então fazemos os mesmos passos, exceto gerar `weld`. Nesse

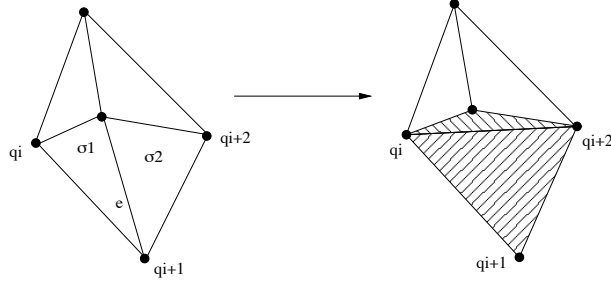


Figura 9: Um dos casos da função `make_compatibility_spl`. $q_i q_{i+1} q_{i+2}$ é uma orelha candidata a flip que estará na filtragem. σ_1, σ_2 são faces com atributo booleano falso. São gerados os operadores `create` para σ_1, σ_2 e depois o operador `flip` na aresta e .

caso, para os triângulos vizinhos a s_i com atributo $f = \text{verdadeiro}$ são gerados operadores `destroy` e atualizamos os atributos para $falso$. Também seguem a ordem da potência de ponto e depois geramos um operador `remove` (III.S). O mesmo acontece se o ponto s_i estiver no bordo sendo que as operações de `destroy` seguem o sentido horário (IV.S) (*loop* da linha 24 do algoritmo 5).

Algorithm 5: Função `make_compatibility_spl`

Entrada: $ear^* candidate_ear$

```

1  $he \leftarrow candidate\_ear.he_2$  ;
2 if  $test\_triangle\_spl(candidate\_ear)$  then
3   if  $he.face.f = falso$  then
4     gerar create( $he.face$ ) ;
5      $he.face.f = verdadeiro$  ;
6   if  $he.mate.face = falso$  then
7     gerar create( $he.mate.face$ ) ;
8      $he.mate.face.f = verdadeiro$  ;
9 else
10  if  $he.face.f = falso$  then
11    gerar destroy( $he.face$ ) ;
12     $he.face.f = falso$  ;
13  if  $he.mate.face = falso$  then
14    gerar create( $he.mate.face$ ) ;
15     $he.mate.face.f = falso$  ;

```

Inversão: Identificamos as operações inversas em partes determinadas dos algoritmos de refinamento e simplificação. Observe que os flips são ordenados nas filas L_{he} e L_{ear} de maneira simétrica pela potência de ponto. Há simetria entre as funções de compatibilização identificadas por (V.R), (V.S). É fácil ver também, que há simetria em (I.R = I.S⁻¹), (II.R = II.S⁻¹), (III.R = III.S⁻¹) e (IV.R = IV.S⁻¹). \square

Os dois algoritmos podem ser facilmente estendidos para funções monótonas $g : \mathbb{R} \rightarrow \mathbb{R}$ não decrescentes sobre os raios das amostragens de uma família escalonada. Para isso, basta adaptarmos as funções `test_triangle_ref` e `test_triangle_spl` para comparar os raios dos círculos circunscritos das faces segundo $g(\alpha)$. Isso permite fazer um **controle topológico da filtragem**, i.e., a função g controla todas as mudanças de topologia entre os níveis da filtragem.

Corolário 1. Seja $g : \mathbb{R} \rightarrow \mathbb{R}$ uma função monótona não decrescente. Sejam $\mathcal{F} = \{S_{\alpha_i}\}$ uma família cadenciada e $C_\alpha F = \{C_{g(\alpha_i)}(S_{\alpha_i})\}$ a família de Alfa Complexos Sólidos de F com a função g . Então, existe um algoritmo por refinamento e um algoritmo por simplificação que geram um filtragem por operadores topológicos de $C_{g(\alpha)}\mathcal{F}$ cujas operações topológicas de refinamento são inversas às operações topológicas de simplificação.

Note que a triangulação de Delaunay é equivalente ao corolário acima com uma função $g(x) = \infty$.

7 Extensões

Até aqui tratamos de ADP's e alfa complexos sólidos. Nesta seção vamos definir um tipo de amostragem mais geral que são as (α, β) -ADP's. Também estenderemos a definição de Alfa Complexo Sólido para Quasi-Alfa Complexo Sólido onde nesse último as informações topológicas dos pontos (bordo ou interior) são levadas em consideração. Nosso principal objetivo é chegar ao Teorema 3 que estende o Teorema 1 utilizando estas duas novas extensões.

7.1 Aproximação do Bordo

Vamos definir quando um α -par $(\tilde{S}_\alpha, K_\alpha)$ é uma *boa* aproximação numa região, lembrando que o símbolo \sim significa dizer que S_α é uma amostragem interpolante.

Definição 28. Seja $(\tilde{S}_\alpha, K_\alpha)$ um α -par de uma região R . Considere $P = \overline{\partial K_\alpha}$. Se todos os vértices de P forem de bordo então dizemos que K_α é uma *boa* aproximação de R .

Como foi discutido anteriormente, a partir dos complexos duais, as ADP's aproximantes são mais ruidosas no bordo que as ADP's interpolantes porque não toma pontos representativos do bordo. Consequentemente, quando o parâmetro α tende a zero a sequência dada por uma família de alfa complexos sólidos de ADP's interpolantes que são boas aproximações garantem convergência das normais, o que não ocorre com famílias de ADP's aproximantes. Assim, em termos, geométricos as amostragens interpolantes que são boas aproximações são melhores.

Nem sempre um α -par $(\tilde{S}_\alpha, \overline{C_\alpha(S_\alpha)})$ é uma *boa* aproximação. Na figura 4.b nota-se um ponto de interior que faz parte do bordo da reconstrução. A fim de contornar esse problema, vamos definir um Quasi-Alfa Complexo de dimensão 2 com o objetivo de garantir que ele tenha características combinadas das definições de um Alfa Complexo Sólido e de uma boa aproximação.

Definição 29. Seja S uma amostragem qualquer com informações topológicas nos pontos (de bordo ou de interior) de uma região R . Dado $\alpha > 0$, denominamos de $QC_\alpha(S)$, um Quasi-Alfa Complexo Sólido (QACS), como um complexo simplicial sólido de S que satisfaz as seguintes propriedades:

1. $QC_\alpha(S) \subset Del(S)$
2. $\sigma_{(s_i, s_j, s_k)} \in QC_\alpha(S)^2 \iff$
 - (a) ou $\mu_\sigma < \alpha$, onde μ_σ é o raio do círculo circunscrito de σ ;
 - (b) ou $\{s_i, s_j, s_k\} \cap int(R) \neq \emptyset$;
 - (c) ou $\exists B_l$ e B_m , componentes conexas distintas do bordo de R , tais que $B_l \cap \{s_i, s_j, s_k\} \neq \emptyset$ e $B_m \cap \{s_i, s_j, s_k\} \neq \emptyset$;

Observe que pela propriedade 2(a) segue que $\overline{C_\alpha(S)} \subset QC_\alpha(S)$. Da definição acima, temos também o seguinte lema:

Lema 2. Se S é uma ADP interpolante de uma região R então:

1. $QC_\alpha(S)$ é uma boa aproximação;
2. Se B_l é uma componente conexa do bordo de $QC_\alpha(S)$ então todos os seus pontos são de uma mesma componente conexa do bordo de R .

Demonstração. O item (1) segue diretamente da propriedade 2.b da definição 29. O item (2) é consequência direta da propriedade 2.c da definição 29. \square

Como queríamos, um α -par $(\tilde{S}_\alpha, QC_\alpha(\tilde{S}_\alpha))$ é uma sempre boa aproximação e, consequentemente, garante a convergência de normais. Também temos que $QC_\alpha(\tilde{S}_\alpha)$ tem propriedades importantes de ser um subconjunto da triangulação de Delaunay e um super-conjunto do alfa complexo sólido. A desvantagem advém do fato de que eles não têm um limite superior para a razão de aspecto do conjunto dos triângulos. É o que vamos procurar resolver na próxima seção.

7.2 Razão de Aspecto

A razão de aspecto sem limite superior acontece porque pontos de interior amostrados muito próximos do bordo eliminam em sua vizinhança a possibilidade de amostrar pontos de bordo que seriam mais representativos (veja figura 10). Para contornar esse problema, precisamos de uma estratégia de amostragem que garanta que pontos de interior não sejam amostrados em uma certa vizinhança $\epsilon(\alpha)$ do bordo da região. Ao tentarmos adotar essa idéia tivemos que usar uma versão mais fraca de ADP's que são as (α, β) -ADP's.

Definição 30. Sejam $S_{\alpha\beta} = \{s_1, s_2, \dots, s_n\}$ uma amostragem de uma região sólida R ($R = A \cup \partial A$, A aberto e conexo) e $0 < \beta \leq \alpha$. Dizemos que $S_{\alpha\beta}$ é uma (α, β) -ADP se (i) $R \subset \cup_{s_i \in S_{\alpha\beta}} B_\alpha(s_i)$ (condição de cobertura) e, além disso, (ii) $S_{\alpha\beta} \cap B_\beta(s_i) = \{s_i\}$, $\forall i$ (condição de Poisson).

Note que quando o parâmetro β se aproxima de α , a amostragem fica mais semelhante com uma ADP. A idéia de escala fica implícita no parâmetro β . A seguir vamos exibir uma maneira de gerar uma (α, β) -ADP de uma região garantindo que $\beta = \alpha/2$.

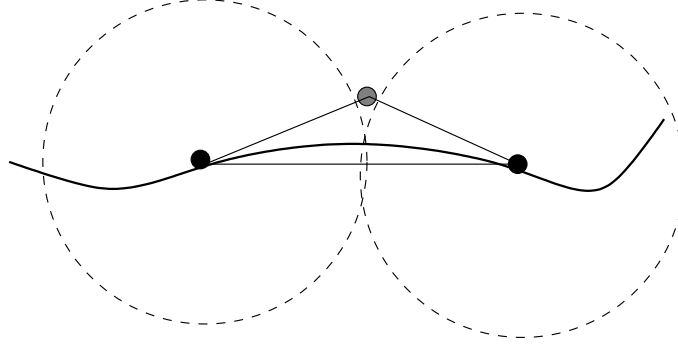


Figura 10: Neste exemplo o ponto cinza claro está no interior e está próximo ao bordo. O triângulo plotado pode ter uma razão de aspecto muito grande.

Proposição 9. Dado α , podemos gerar uma $(\alpha, \alpha/2)$ -ADP de uma região R com os seguintes passos:

1. $I = \{ \text{ADP de } R - \cup_{s \in \partial R} B_\alpha(s) \}$;
2. $B = \{ \text{ADP de } \partial R \}$;
3. $P = \{ \text{"ADP" de } R - \cup_{s \in B \cup I} B_\alpha(s) \text{ com projeção condicionada no bordo} \}$.
A projeção condicionada é feita após um amostra s ser gerada e validada. Se $\text{dist}(s, \partial R) < \frac{\alpha}{2}$ então escolha s' como o ponto mais próximo de ∂R , senão, escolha s .

Demonstração. Por construção é claro que $R \subset \cup_{s \in I \cup B \cup P} B_\alpha(s)$. Vamos mostrar que $\beta = \alpha/2$. Seja s' a projeção de s em ∂R . Seja $p \in \partial R$ o ponto da amostragem mais próximo de s . Sabemos que $ss' < \alpha/2$ e que $ps > \alpha$. Pela desigualdade triangular, temos que $ps' + ss' > ps \Rightarrow ps' > ps - ss' > \alpha - \alpha/2 > \alpha$. Logo $\beta = \alpha/2$. \square

Observe que se $s \in \text{int}(R)$ pelos passos 2 e 3 concluímos que $\text{dist}(s, \partial R) > \alpha/2$. Daí, podemos dizer que existe um vizinhança $\epsilon(\alpha) = \alpha/2$ do bordo que não tem pontos de interior. Como mencionado anteriormente isso melhora o problema de existirem triângulos muito finos (alta razão de aspecto) ao longo do bordo um α -par $(\tilde{S}_\alpha, QC_\alpha(\tilde{S}_\alpha))$. No momento não temos como exibir um limite superior mas conjecturamos que seja ≤ 8 . Temos gerado vários exemplos e todos exibem resultados satisfatórios e abaixo desse limite.

Definição 31. Seja $S_{\alpha, \beta} = \{s_1, s_2, \dots, s_n\}$ uma amostragem de uma região sólida R . Dizemos que $S_{\alpha, \beta}$ é uma (α, β) -ADP *interpolante* se existe $A \subset S_{\alpha, \beta}$ tal que $A \subset \partial R$ e $\partial R \subset \cup_{s_i \in A} B_\alpha(s_i)$. Vamos denotar uma (α, β) -ADP interpolante por $\tilde{S}_{\alpha, \beta}$.

É fácil ver que a (α, β) -ADP da proposição 9 é uma amostragem interpolante conforme a definição acima.

7.3 Família Quasi-Escalonada

Da mesma forma que definimos as famílias escalonadas para ADP's, podemos de forma análoga definir Família Quasi-Escalonada (FQE) e suas derivações.

Definição 32. Seja $\mathcal{F} = \{S_{\alpha_i\beta_i}\}_{i \in \{1,2,\dots,n\}}$ uma família de ADP's de uma região R . Dizemos que F é *escalonada* se $S_{\alpha_1\beta_1} \subset S_{\alpha_2\beta_2} \dots \subset S_{\alpha_n\beta_n}$, com $\beta_1 > \beta_2 > \dots > \beta_n$. A cada um dos i 's denominamos de *níveis de escala* e seus β_i 's de *escalas*.

A seguir esquematizamos um algoritmo que constrói uma família de malhas quasi-escalonada.

Proposição 10. Dado $\alpha_1 > 0$, podemos gerar uma família de malhas quasi-escalonada $\mathcal{F} = \{S_{\alpha_i\beta_i}\}_{i \in \{1,2,\dots,n\}}$ de uma região R tal que:

1. $\alpha_{i+1} = \alpha_i/2$;
2. $\beta_{i+1} = \beta_i/2$;
3. $\beta_i = \alpha_i/2$.

com os seguintes passos:

1. Faça $i = 1$; utilize a proposição 9 para gerar $S_{\alpha_i\beta_i}$; faça $i = i + 1$ e $\alpha_i = \alpha_i/2$;
2. Utilize a proposição 9 gerar $S_{\alpha_i\beta_i} - S_{\alpha_{i-1}\beta_{i-1}} \subset (R - \cup_{s_j \in S_{\alpha_{i-1}\beta_{i-1}}} B_{\beta_i}(s_j))$;
3. Faça $\alpha_{i+1} = \alpha_i/2$; $i = i + 1$, e volte para o passo 1 até $i = n$;

Demonstração. Conforme a proposição 9 segue que $\beta_i = \alpha_i/2$ para cada i . \square

7.3.1 Família de Malhas Quasi-Escalonadas

Vamos agora definir as estruturas associadas às FQE's.

Definição 33. Seja $S_{\alpha\beta}$ uma (α, β) -ADP de uma região R e $K_{\alpha\beta}$ um complexo simplicial sólido. Dizemos que $(S_{\alpha\beta}, K_{\alpha\beta})$ é um (α, β) -par se $K_{\alpha\beta}^0 \subset S_{\alpha\beta}$.

Definição 34. Seja $\mathcal{F} = \{S_{\alpha_i\beta_i}\}_{i \in \{1,2,\dots,n\}}$ uma família escalonável de uma região R . Dizemos que $\mathcal{M}(\mathcal{F}) = \{(S_{\alpha_i\beta_i}, K_{\alpha_i\beta_i})\}_{i \in \{1,2,\dots,n\}}$ é uma *família de Malhas Quasi-Escalonada* se $(S_{\alpha_i\beta_i}, K_{\alpha_i\beta_i})$ é um (α, β) -par para todo i . Assim como famílias escalonadas, a cada um dos i 's denominamos de *níveis de escala* e seus β_i 's de *escalas*.

Exemplo 2. Seja $F = \{\tilde{S}_{\alpha_i\beta_i}\}_{i \in \{1,2,\dots,n\}}$ uma Família Quasi-Escalonada. Dizemos então que $QC_{\alpha}F = \{(\tilde{S}_{\alpha_i\beta_i}, C_{\alpha_i}(\tilde{S}_{\alpha_i\beta_i}))\}_{i \in \{1,2,\dots,n\}}$ é uma família de *Quasi-Alfa Complexos Sólidos Escalonada* (FQACSE) onde $QC_{\alpha}(\tilde{S}_{\alpha_i\beta_i}) = \{\text{Quasi-Alfa Complexo Sólido de } \tilde{S}_{\alpha_i\beta_i}\}$.

A partir da proposição 10 podemos concluir quatro fatos importantes sobre FQACSE's:

- A distância no sentido de Hausdorff do bordo é limitada por α_i em cada nível, garantindo a convergência da aproximação da região sólida por QACS's ;
- A convergência da sequência dada pela FQACS's é de boas aproximações e, portanto, tem convergência de normais;
- Em cada nível de amostragem seus respectivos QACS's tem compatibilidade com a informação topológica, i.e., se $s \in \text{int}(R)$ então $s \in \text{int}(C_\alpha(R))$.
- Na prática, o conjunto os triângulos têm razão de aspecto menor que 8.

7.3.2 Cadenciando uma Família Quasi-Escalonada

Na seção 6.1 expomos as dificuldades de cadenciar uma família escalonada. Para a nossa surpresa, devido à maior generalidade de uma família quasi-escalonada isso é sempre possível. O principal resultado sobre cadenciamento de uma FQE é o teorema 2. Primeiramente vamos definir quando uma FQE é cadenciada.

Definição 35. Seja $F = \{S_{\alpha_i\beta_i}\}_{i \in \{1,2,\dots,n\}}$ uma família Quasi-Escalonada de R . Dizemos que F é Quasi-Cadenciada se $S_{\alpha_{i+1}\beta_{i+1}} = S_{\alpha_i\beta_i} \cup \{s_i\}$, $\beta_{i+1} < \beta_i$, $\alpha_{i+1} < \alpha_i$ com $\beta_{i+1} = d(S_{\alpha_i\beta_i}, s_i)$.

Teorema 2. Seja $\mathcal{F} = \{S_{\alpha_i\beta_i}\}_{i \in \{1,2,\dots,n\}}$ uma família Quasi-Escalonada de R . Então existe uma família \mathcal{F}_c que cadencia \mathcal{F} .

Demonstração. Vamos mostrar duas maneiras de cadenciar uma Família Quasi-Escalonada. Na primeira maneira ordenamos diretamente inserindo pontos. Na segunda ordenamos inversamente removendo pontos.

1ª) seja $D = S_{\alpha_{i+1}\beta_{i+1}} - S_{\alpha_i\beta_i} = \{s_1, s_2, \dots, s_k\}$, daí geramos então uma ordenação $D_\sigma = \{s_{\sigma(1)}, s_{\sigma(2)}, \dots, s_{\sigma(k)}\}$ tal que pela unicidade da distância $\exists! s_{\sigma(l)} \in D$ vale:

$$\max\{d(s, A_{l-1}) \mid s \in D_{l-1}\} = \beta_{i_l}$$

onde, $D_l = D - \{s_{\sigma(1)}, s_{\sigma(2)}, \dots, s_{\sigma(l)}\}$ e $A_l = S_{\alpha_i\beta_i} \cup \{s_{\sigma(1)}, s_{\sigma(2)}, \dots, s_{\sigma(l)}\}$.

Aqui vamos fazer duas observações. Na primeira, se considerarmos $DT(A_l)$. O valor de α_{i_l} pode ser escolhido como o menor real positivo que satisfaz a condição de cobertura:

$$\alpha_{i_l} = \inf\{\alpha \in \mathbb{R} \mid R \subset \cup_{s \in A_l} B_\alpha(s)\}.$$

2ª) Dado $S_{\alpha_i\beta_i} \cup \{\sigma_1, \sigma_2, \dots, \sigma_k\} = S_{\alpha_{i+1}\beta_{i+1}}$, o algoritmo de cadenciamento de \mathcal{F} é inicializado a partir da triangulação de Delaunay de $S_{\alpha_{i+1}\beta_{i+1}}$. Fazemos ordenação de maneira inversa, utilizando mais uma vez o algoritmo de remoção de pontos descrito em Devillers [18].

Construímos inicialmente uma fila de prioridades L_e que contém todas as arestas da triangulação cuja primeira da fila é a de menor tamanho. Sejam $lv(w)$ o nível do ponto w e $\text{link}(w)$ o conjunto formado pelas arestas vizinhas a w em uma malha. Seja e a aresta que está no topo da fila de prioridades

L_e . O ponto a ser removido é um dos vértices de e . Sejam p e q tais vértices. Então se $lv(p) > lv(q)$ removemos p conforme descrito em [18]. Analogamente, se $lv(q) > lv(p)$ então removemos q . Em caso de igualdade entre os níveis, optamos por remover aquele que estiver mais próximo de seus vértices vizinhos, desconsiderando a aresta e . Nossa implementação a respeito de “medida de proximidade dos vizinhos” para um vértice qualquer é o comprimento da maior aresta que está no seu link. Quanto maior esse comprimento, maior a distância. Assim, se $lv(p) = lv(q)$ considere e_p e e_q arestas tais que $e_p = \max\{m(w) | w \in \text{link}(p)\}$ e $e_q = \max\{m(w) | w \in \text{link}(q)\}$, onde $m(w)$ é o comprimento de w . Se $m(e_p) > m(e_q)$ então removemos q caso contrário, removemos p . A fila L_e é atualizada e o algoritmo prossegue da mesma forma. Observe que o valor de β corresponde ao comprimento da aresta que está no topo de L_e . Veja o algoritmo 6.

Algorithm 6: Ordenação Inversa

Entrada: $S_{\alpha_{i+1}, \beta_{i+1}}$.
Saída : Ordenação de $S_{\alpha_{i+1}, \beta_{i+1}} - S_{\alpha_i, \beta_i}$.

$A \leftarrow S_{\alpha_{i+1}, \beta_{i+1}} - S_{\alpha_i, \beta_i}$;
 $M \leftarrow DT(S_{\alpha_{i+1}, \beta_{i+1}})$;
 Crie fila de prioridades L_e ;

while $A \neq \emptyset$ **do**

- $e \leftarrow$ topo de L_e ;
- if** $lv(p) > lv(q)$ **then**
 - remova p de $DT(S_{\alpha_{i+1}, \beta_{i+1}})$;
 - $A \leftarrow A - \{p\}$;
- if** $lv(q) > lv(p)$ **then**
 - remova q de $DT(S_{\alpha_{i+1}, \beta_{i+1}})$;
 - $A \leftarrow A - \{q\}$;
- if** $lv(q) = lv(p)$ **then**
 - $e_p = \max\{m(w) \mid w \in \text{link}(p)\}$;
 - $e_q = \max\{m(w) \mid w \in \text{link}(q)\}$;
 - if** $e_p > e_q$ **then**
 - remova q de $DT(S_{\alpha_{i+1}, \beta_{i+1}})$;
 - $A \leftarrow A - \{q\}$;
 - else**
 - remova p de $DT(S_{\alpha_{i+1}, \beta_{i+1}})$;
 - $A \leftarrow A - \{p\}$;

Atualize L_e ;

Em ambos os casos, Assim geramos subníveis “intermediários” entre os níveis de uma Família Quasi-Escalonada a fim de que se torne uma Família Quasi-Cadenciada. \square

7.4 Retornando à Filtragem

Equivalente ao Teorema 1 para qualquer FQACS não necessariamente cadenciada, temos o seguinte teorema:

Teorema 3. Para qualquer função $g : \mathbb{R} \rightarrow \mathbb{R}$ uma função monótona não decrescente e qualquer FQACSE $QC_{g(\alpha)}\mathcal{F} = \{(\tilde{S}_{\alpha_i\beta_i}, QC_{\alpha_i}(\tilde{S}_{\alpha_i\beta_i}))\}_{i \in \{1,2,\dots,n\}}$, existe um algoritmo por refinamento e um algoritmo por simplificação que geram filtragens por operadores topológicos de $QC_{\alpha}\mathcal{F}$ cujas operações topológicas de refinamento são inversas às operações topológicas de simplificação.

Demonstração. Primeiramente aplicamos o teorema 2 para cadenciar os níveis intermediários de F em uma nova família F_c . O algoritmo é o mesmo do teorema 1 com a modificação da função `test_triangle_ref` para o refinamento e da função `test_triangle_spl` para a simplificação. Nessas funções os testes de validação para que uma face seja incluída ou removida são dados pelo parâmetro $g(\alpha_i)$ e pelo item 2 da definição 29. \square

A função g permite um certo grau de controle aos algoritmos na construção da filtragem por operadores topológicos. Como visto anteriormente, se a função $g = \infty$ a filtragem equivale à triangulação de Delaunay.

8 Conclusão

Nesse artigo introduzimos um *framework* teórico e prático para multiresolução com controle de mudança de topologia. Até o nosso conhecimento não existem trabalhos abordando tal problema.

Nosso objeto gráfico de estudo é uma região sólida. Fizemos uma análise da geometria do bordo e da topologia através de amostragens por discos de Poisson (ADP's e (α, β) -ADP's) reconstruídas por *alfa* complexos sólidos (e seus variantes) para cada nível de resolução. Com isso pudemos intercalar operações atômicas (estelares e de *handle*) entre os níveis de resolução. A essa sequência de operações denominamos de *filtragem por operações topológicas*.

Na figura 11 temos de a) a d) quatro níveis de resolução de uma região sólida formada por um retângulo com dois buracos que são círculos de raios distintos. Os níveis foram amostrados conforme a proposição 10. Como esperado, à medida que a aumentamos a resolução os buracos vão aparecendo, bem como seus respectivos bordos são mais detalhados. De e) a h) estão representados níveis intermediários entre os níveis c) e d). Os pontos foram ordenados segundo o algoritmo 6.

O principal resultado do artigo traz um teorema (em duas versões) que diz que é possível gerar duas filtragens por operadores topológicos tanto por simplificação como por refinamento de forma independente com a propriedade das operações serem inversas. Este teorema permitirá gerar a representação de uma *malha em resolução variável* [12] que seja flexível de tal sorte que funções de adaptação poderão gerar malhas adaptativas partindo tanto do nível mais simplificado como do nível mais refinado, a depender da aplicação. É o que faremos em um dos nossos trabalhos futuros que são os seguintes:

- Gerar uma estrutura hierárquica entre níveis de resolução para compor uma malha em resolução variável;
- Criar funções de adaptação;
- Gerar aplicações;

- Generalizar todos os resultados para dimensão três.

Os três primeiros são resultados mais imediatos que estamos trabalhando atualmente.

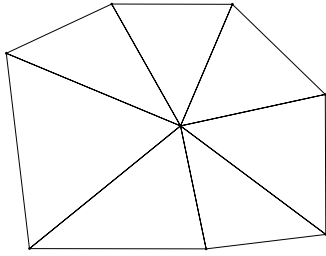
Agradecimentos

Agradecemos ao CNPq por todo apoio financeiro e ao IMPA pela estrutura e motivação. Agradecemos a Dimas Martinez pela revisão e sugestões.

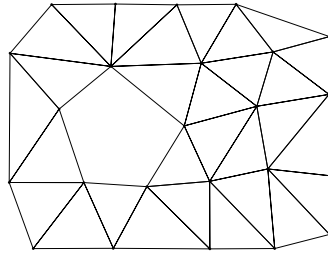
Referências

- [1] J. Gomes, B. Costa, L. Darsa e L. Velho. *Graphical Objects*. The Visual Computer, 12(6): 269–282, 1996.
- [2] N. Amenta, M. Bern e D. Eppstein. *The Crust and the β -Skeleton: Combinatorial Curve Reconstruction*. Graphical Models and Image Processing, 60/2:2, 125–135, 1998.
- [3] F. Bernardini e C. Bajaj. *Sampling and Reconstructing Manifolds Using Alpha-Shapes*. Proc. of the Ninth Canadian Conference on Computational Geometry, 193–198, Agosto, 1997.
- [4] F. Bernardini, C. Bajaj, J. Chen and D. Schikore. *Automatic Reconstruction of 3D CAD Models from Digital Scans*. Int. J. on Comp. Geom. and Appl., 9(4–5):327–370, Agosto–Outubro 1999.
- [5] C. E. Shannon. *Communication in the presence of noise*. Proc. Institute of Radio Engineers, 37(1): 10–21, 1949.
- [6] E. Medeiros, L. Velho e H. Lopes. *Restricted BPA: Applying Ball-Pivoting on the Plane*. Proc. do XVII Simpósio Brasileiro em Computação Gráfica e Processamento de Imagens (SIBGRAPI), Curitiba-Brasil, 2004.
- [7] E. Medeiros, T. Lewiner, H. Lopes, and L. Velho. *An Aspect Ratio Upper Bound in 2D Solid Alpha Complexes of Poisson Disc Samplings*. Technical Report 02-07, IMPA - VISGRAF Laboratory, 2007.
- [8] E. Medeiros, T. Lewiner, H. Lopes, and L. Velho. *Reconstructing Poisson Disc Samplings of Solid Objects with Topological Guarantees*. Technical Report 03-07, IMPA - VISGRAF Laboratory, 2007.
- [9] L. Kobbelt. *$\sqrt{3}$ -Subivision*. Proc. do SIGGRAPH, 2000.
- [10] L. Velho e D. Zorin. *4–8 Subdivision*. Computer-Aided Geometric Design, 18(5):397–427, 2001.
- [11] L. Velho, H. Lopes, E. Medeiros, T. Lewiner e G. Tavares. *Topological Mesh Operators*. Technical Report - VISGRAF Laboratory TR-2005-03, IMPA, Outubro 2005.
- [12] E. Puppo. *Variable Resolution Triangulations*. Computational Geometry, 11(3–4): 219–238, 1998.

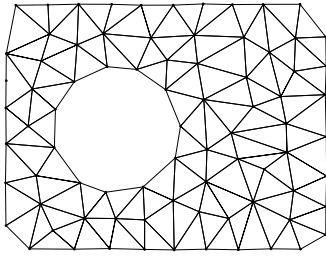
- [13] H. Hoppe. *Progressive meshes*. Proc. do SIGGRAPH, 1996.
- [14] H. Edelsbrunner, D. Letscher e A. Zomorodian. *Topological Persistence and Simplification*. Discrete Comput. Geom., 28: 511–533, 2002.
- [15] H. Edelsbrunner. *The Union of Balls and its Dual Shape*. Discrete Comput. Geom. 13: 415–44, 1995.
- [16] H. Edelsbrunner e E. P. Mucke. *Three-Dimensional Alpha Shapes*. ACM Trans. Graphics 13: 43–72, 1994.
- [17] M. de Berg, M. van Kreveld, M. Overmars e O. Schwarzkopf. *Computational Geometry*, Springer-Verlag, Berlin, 1997.
- [18] O. Devillers. *On Deletion in Delaunay Triangulation*. Internat. J. Comput. Geom. Appl., 12:193-205, 2002.
- [19] R. L. Cook. *Stochastic Sampling in Computer Graphics*. ACM Transactions on Graphics, 5(1):51-72, Janeiro, 1986.
- [20] S. Hiller and O. Deussen. *Tiled Blue Noise Samples*. Proceedings of VISION, MODELING, AND VISUALIZATION, IOS Press, pp. 265–271, 2001.
- [21] C. L. Lawson. *Software for C1 Surface Interpolation*. Mathematical Software III (John R. Rice, editor), Academic Press, pp. 161-194., New York, 1977.



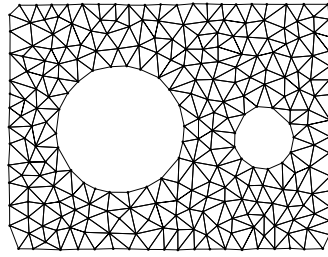
(a)



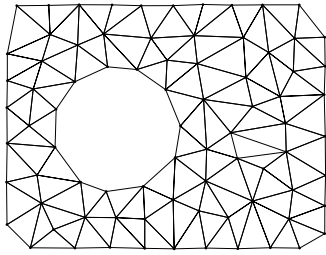
(b)



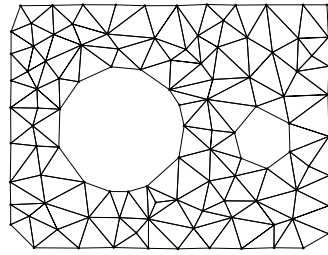
(c)



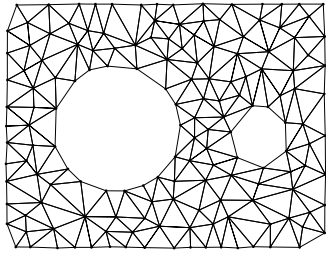
(d)



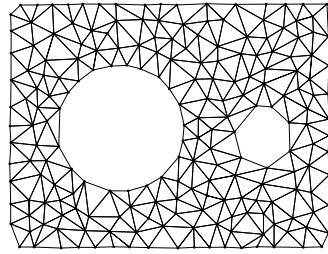
(e)



(f)



(g)



(h)

Figura 11:
35