

## Uma Visão Geral sobre Técnicas para Construção de Algoritmos para a Geração da Tesselação de Delaunay e do Diagrama de Voronoi

*An Overview of Techniques for Building Algorithms for the Generation of the Delaunay's Tessellation and Voronoi Diagram*

Jéssica Renata Nogueira<sup>1</sup>, Sanderson L. Gonzaga de Oliveira<sup>2</sup>

### RESUMO

Neste trabalho, são abordadas, resumidamente, as técnicas utilizadas no projeto de algoritmos para a geração da tesselação de Delaunay ou do diagrama de Voronoi. Essas malhas possuem aplicações em áreas diversas, em que computação científica, visualização e sistemas de informações geográficas são exemplos. Várias técnicas já foram utilizadas na construção de algoritmos para a geração da tesselação de Delaunay e do diagrama de Voronoi. Escolher a técnica mais adequada pode permitir a criação de algoritmos eficientes em tempo de execução, na utilização de memória, que facilitem a implementação e/ou o paralelismo, entre outras vantagens. Neste trabalho são descritas as abordagens por divisão e conquista, inserção incremental, mudança global de arestas, por linha de varredura (*sweep-line*), por *lifting map*, por *gift wrapping* e a técnica de varredura por círculo.

**Palavras-chave:** Tesselação de Delaunay, Diagrama de Voronoi, Geração de malhas, Projeto de algoritmos.

### ABSTRACT

This paper overviews the techniques used for the design of algorithms for the generation of the Delaunay tessellation and Voronoi diagram. These meshes have applications in several areas, where scientific computing, scientific visualization, and geographic information systems are examples. Practitioners have been using several techniques in the construction of algorithms for the generation of the Delaunay tessellation and Voronoi diagram. Choosing the most appropriate technique can allow the creation of low-cost algorithms (in time and space) that facilitate implementation and parallelism, among other advantages. This paper considers the following techniques: divide and conquer, incremental algorithms, the Lawson flip algorithm, sweep-line algorithms, lifting map, gift wrapping, and sweep-circle algorithms.

**Keywords:** Mesh generation. Delaunay tessellation. Voronoi Diagram. Computer-aided design, engineering, and manufacturing, Computational geometry and topology.

<sup>1</sup> Doutoranda em Biotecnologia e professora do Instituto Federal de Ciência e Tecnologia do Sul de Minas Gerais – Campus Passos.

E-mail:

jessica.nogueira@ifsuldeminas.edu.br

<sup>2</sup> Doutor em Ciência da Computação e Professor da Universidade Federal de Lavras - UFLA.

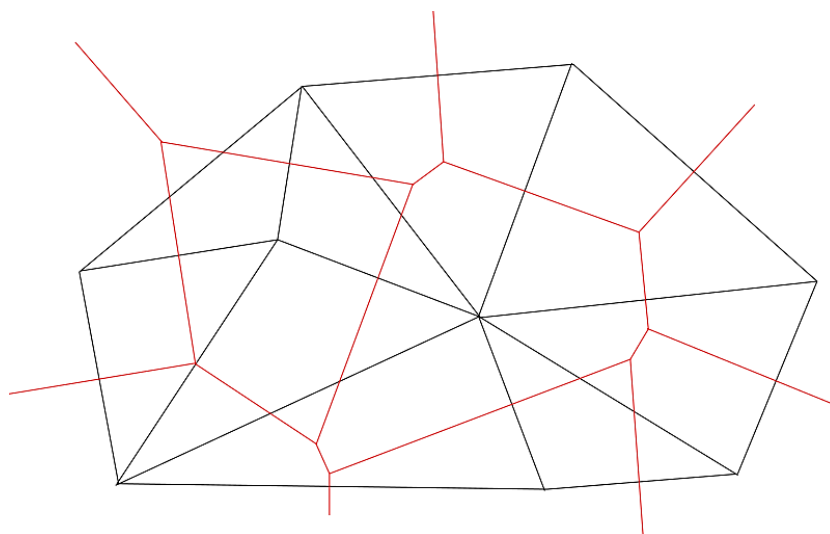
## 1. INTRODUÇÃO

As tesselações de Delaunay e de Voronoi são empregadas na resolução de problemas presentes em áreas diversas, tais como problemas físicos e de engenharia. Essas malhas podem ser utilizadas, por exemplo, na modelagem molecular, de sólidos e de terrenos, e em vídeo *games* (Gonzaga de Oliveira *et al.*, 2014). A tesselação de Delaunay tem sido utilizada em simulações numéricas em uma grande variedade de disciplinas, em que geofísica, método dos elementos finitos e cristalografia são exemplos.

Uma tesselação de Delaunay (*TD*) para um conjunto  $S$  de vértices deve atender à condição de Delaunay, que é:  $TD(S)$  é uma tesselação de Delaunay se nenhum circuncírculo (circunferência, em três dimensões) que passa pelos triângulos (tetraedros) da malha contém pontos de outros triângulos (tetraedros) em seu interior. Exceto em casos degenerados, em que quatro (cinco, em três dimensões) ou mais pontos são cocirculares (coesféricos), a tesselação de Delaunay gerada para um conjunto de pontos é única.

O diagrama de Voronoi, definido como uma estrutura dual da tesselação de Delaunay, é um tipo específico de decomposição do espaço, cujos perímetros dos polígonos de Voronoi são determinados de acordo com a distância euclidiana a pontos geradores da malha.

Considere como um conjunto de  $n$  pontos geradores no espaço euclidiano, para  $S \subseteq \mathbb{R}^d$ . O politopo de Voronoi  $P_i$  de um ponto gerador  $p_i$  é o conjunto de todos os pontos no  $\mathbb{R}^d$  que são mais próximos ao ponto gerador  $p_i$  do que a qualquer ponto gerador em  $S$ . Formalmente, para uma dimensão  $d \geq 2$ , cada politopo é um conjunto de pontos  $P_i = \{p \in \mathbb{R}^d : (\exists p_i \in S)(\forall p_j \in S) \|p - p_i\| \leq \|p - p_j\|\}$ , com  $i \neq j \wedge 1 \leq i, j \leq n$  (Gonzaga de Oliveira *et al.*, 2014; Gonzaga de Oliveira, 2015). Na Figura 1, pode ser observada uma partição do diagrama de Voronoi e a triangulação de Delaunay correspondente.



**Figura 1:** Exemplo de triangulação de Delaunay e diagrama de Voronoi correspondente.

Este artigo é uma versão revisada de artigo apresentado no XX Encontro Nacional de Modelagem Computacional (Nogueira e Gonzaga De Oliveira, 2017). Na seção 2, são descritos os métodos empregados para a realização deste trabalho. Na seção 3, são apresentadas técnicas utilizadas no projeto de algoritmos e exemplos de algoritmos utilizando essas técnicas para a geração das malhas de Delaunay e de Voronoi. As técnicas apresentadas são as abordagens incremental, por divisão e conquista, por mudança global de arestas, por linha de varredura (*sweepline*), por *lifting map*, por *gift wrapping* e a técnica de varredura por círculo (*sweep-circle*). Finalmente, as discussões e considerações finais são apresentadas, respectivamente, nas seções 4 e 5.

## 2. MÉTODOS

Neste trabalho, é mostrada uma revisão das técnicas já utilizadas na criação de algoritmos para a geração das malhas de Delaunay e Voronoi. A natureza desta pesquisa é básica.

Para realização deste trabalho, foram realizadas buscas em bases de artigos (Scielo, Scopus e Google Acadêmico) e em livros sobre o assunto, como o livro de Preparata e Shamos (1985), Cheng *et al.* (2013) e Gonzaga de Oliveira (2015). Para exemplificação, algoritmos que utilizam as técnicas abordadas são citados. Exceto quando não identificados, esses exemplos são citados para abordagens sequenciais e paralelas, e empregados na construção do diagrama de Voronoi e na tesselação de

Delaunay. Como não houve envolvimento de seres humanos, não houve necessidade de submissão ao Comitê de Ética em Pesquisa.

### 3. RESULTADOS

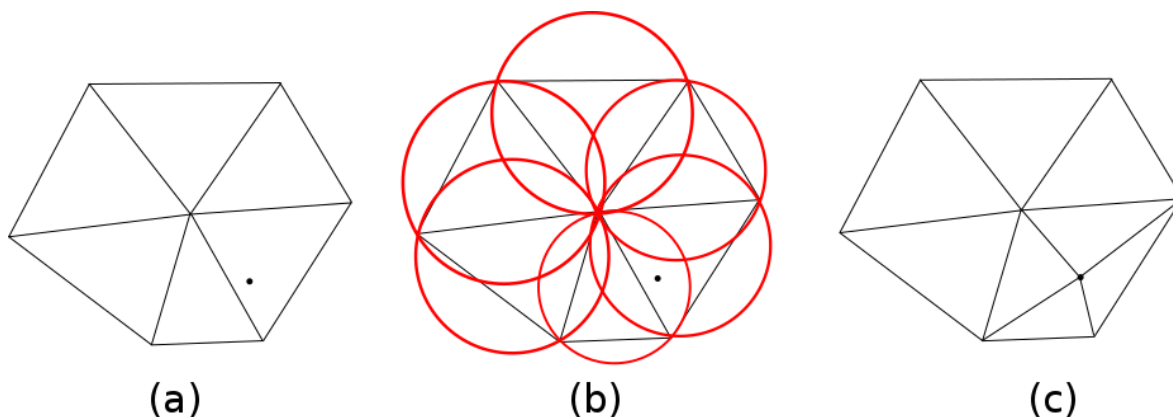
Nas subseções a seguir, são descritas as técnicas identificadas na literatura para a geração das malhas de Delaunay ou de Voronoi. Nas subseções 3.1, 3.2 e 3.3, são descritas, respectivamente, as abordagens incrementais, por divisão e conquista e por mudança global de arestas. Nas subseções 3.4, 3.5, 3.6 e 3.7 são apresentadas as técnicas de linha de varredura (*sweep-line*), *lift mapping*, *gift wrapping* e varredura por círculo (*sweep-circle*), respectivamente.

#### 3.1 ALGORITMOS INCREMENTAIS

Nesta seção, são abordados algoritmos incrementais para a geração das malhas de Delaunay e de Voronoi. Dado um conjunto inicial de pontos, nessa classe de algoritmos, um novo ponto é inserido no domínio a cada iteração e a malha é refeita.

Em geral, algoritmos incrementais utilizam o algoritmo de Bowyer-Watson para a geração da tesselação de Delaunay ou do diagrama de Voronoi. Esse algoritmo é chamado de algoritmo de Bowyer-Watson porque ambos os autores apresentaram a mesma ideia. Entretanto, o algoritmo de Bowyer (1981) foi proposto para a geração do diagrama de Voronoi e o algoritmo de Watson (1981) foi proposto para geração da tesselação de Delaunay. Ambos os artigos foram publicados em sequência em uma mesma edição, de um mesmo *journal*.

Um exemplo da execução do algoritmo incremental de Bowyer-Watson para a geração da triangulação de Delaunay pode ser observado na Figura 2. Na Figura 2(a), é mostrada uma triangulação de Delaunay inicial. Na Figura 2(b), ocorre a inserção de um ponto na malha. Nessa figura, pode-se observar os triângulos da malha e seus circuncírculos correspondentes. Note também que circuncírculos que contêm pontos em seu interior estão em vermelho. Na Figura 2(c), observa-se a triangulação de Delaunay após a remoção de arestas de triângulos cujo circuncírculo contém o novo ponto e a criação de novos triângulos válidos para a triangulação de Delaunay.



**Figura 2:** Exemplos de passos necessários para a construção da triangulação de Delaunay utilizando o algoritmo incremental de Bowyer-Watson.

Exemplos de algoritmos incrementais, cujas implementações são sequenciais, podem ser encontrados em Xi e Zuo (2013), para a construção da triangulação de Delaunay, e em Xu *et al.*(2012), para a construção do diagrama de Voronoi. Para um algoritmo incremental que utiliza paralelismo para a construção da triangulação de Delaunay, veja Lee *et al.* (1997). Algoritmos incrementais são populares e de fácil implementação (Gonzaga de Oliveira *et al.*, 2014; Gonzaga de Oliveira e Nogueira, 2018; Gonzaga de Oliveira *et al.*, 2018).

Em algoritmos incrementais sequenciais, a ordem de inserção dos pontos influencia diretamente no custo de execução do algoritmo. Por esse motivo, foram propostas diversas sequências de inserção de pontos para definir a ordem em que os pontos são inseridos no domínio. Amenta *et al.* (2003) apresentaram o algoritmo incremental *Biased Randomized Insertion Order* (BRIO), em que os pontos são inseridos de maneira parcialmente aleatória no domínio. Com a proposta do esquema BRIO, os pesquisadores começaram a dar maior atenção à ordem em que os pontos são inseridos no domínio. Isso se, principalmente, a ganhos por causa de acertos na busca de dados em memória *cache*.

Liu e Snoeyink (2005) apresentaram um algoritmo incremental que insere os pontos na tesselação de Delaunay na ordem da curva de Hilbert. Esse algoritmo apresentou custo de execução menor que o custo de execução do algoritmo BRIO.

Liu *et al.* (2013) apresentaram um algoritmo incremental em que os pontos são inseridos no domínio de acordo com o percurso em largura na *cut-longest-edge kd-tree*. Esse algoritmo apresentou melhores custos de execução que o algoritmo incremental proposto por Liu e Snoeyink (2005). Gonzaga de Oliveira e Nogueira (2018) avaliaram

diversas sequências de inserção de pontos em algoritmos incrementais em várias distribuições de pontos bidimensionais e tridimensionais, e a sequência dada pela *cut-longest-edge kd-tree* (Liu *et al.*, 2013) mostrou-se superior às demais sequências avaliadas. Foram avaliados apenas algoritmos determinísticos, isto é, sem aleatoriedade. Gonzaga de Oliveira *et al.* (2018) avaliaram três sequências de inserção na tesselação de Delaunay. Foram avaliados algoritmos incrementais para a geração da tesselação de Delaunay com a sequência dada pela *cut-longest-edge kd-tree* (Liu *et al.*, 2013) e duas sequências dadas pelo esquema híbrido BRIO–Hilbert order em conjunto com esquemas de divisão de sub-domínio (*spatial middle* e *spatial median*) empregados na versão 4.11 da biblioteca *Computational Geometry Algorithms Library* (CGAL, 2018). Foram utilizadas oito distribuições de pontos em domínios tridimensionais e também modelos de testes tridimensionais. Conclui-se que o algoritmo incremental com sequência de inserção dada pela ordem híbrida BRIO–Hilbert combinada com a política *spatial middle* empregada na última versão da CGAL pode ser considerada como o atual estado da arte na geração da tesselação de Delaunay tridimensional.

O algoritmo de Liu *et al.* (2013) foi superado pelo algoritmo incremental de Lo (2013), que propôs a utilização do esquema *multigrid* para a geração da triangulação de Delaunay. Detalhes nesse contexto também são encontrados em Gonzaga de Oliveira *et al.* (2014). O algoritmo de Lo (2013) foi superado pela proposta de Tianyun (2016), que combina o uso do *multigrid* com a ordem da curva de Hilbert para a inserção dos pontos.

### 3.2 ALGORITMOS POR DIVISÃO E CONQUISTA

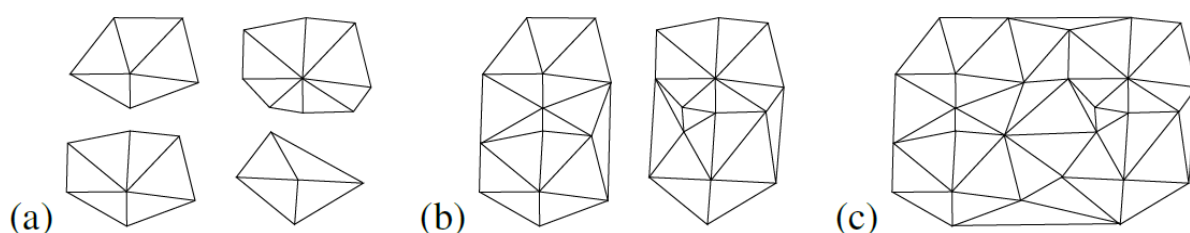
Nesta seção, são abordados algoritmos por divisão e conquista para a geração das malhas de Voronoi e de Delaunay. A técnica de divisão e conquista tem sido amplamente utilizada no projeto de algoritmos para a construção dessas estruturas. A utilização de algoritmos recursivos é tradicional e os algoritmos dessa classe são, geralmente, eficientes (Gonzaga de Oliveira, 2011). Além disso, as implementações de algoritmos por divisão e conquista são elegantes.

Em algoritmos por divisão e conquista para a geração da tesselação de Delaunay ou do diagrama de Voronoi, inicialmente, o conjunto com todos os pontos do domínio é dividido em subconjuntos com números menores de pontos. Após a divisão dos subconjuntos, ocorre a fase de conquista, em que chamadas recursivas ocorrem até que cada subproblema contenha apenas um número pequeno de pontos de forma que a



malha pode ser formada. No retorno da recursão, as malhas obtidas nas resoluções dos subproblemas são combinadas, até que se obtenha a solução final para o problema (Gonzaga de Oliveira, 2011). Nessa solução final, todos os pontos do domínio estão inseridos na malha de Delaunay ou de Voronoi.

Na Figura 3, observa-se um exemplo de construção da triangulação de Delaunay por divisão e conquista. Nessa figura, para cada subconjunto, é construída uma triangulação de Delaunay válida. Na Figura 3(c), é apresentada a triangulação de Delaunay final desse exemplo.



**Figura 3:** Construção da triangulação de Delaunay por divisão e conquista. Nas Figuras (a), (b) e (c), observa-se a combinação das triangulações de Delaunay.

Embora existam algoritmos que possam ocorrer, sob restrições, em tempo linear, Preparata e Shamos (1985) provaram que o tempo ótimo para a geração da triangulação de Delaunay ou para a geração do diagrama de Voronoi, para um conjunto de  $n$  pontos, é  $O(n \lg n)$ . O primeiro algoritmo ótimo para gerar o diagrama de Voronoi com  $n$  pontos geradores no plano foi proposto por Shamos e Hoey (1975) e foi projetado por divisão e conquista. Lee e Schachter (1980) apresentaram uma variação do algoritmo de Shamos e Hoey (1975) para a construção da triangulação de Delaunay por essa mesma técnica, cujo custo também é  $O(n \lg n)$ . Segundo Gonzaga de Oliveira *et al.* (2014), os primeiros algoritmos sequenciais que apresentaram custo esperado linear para a geração do diagrama de Voronoi ou da tesselação de Delaunay foram projetados por divisão e conquista. Esse algoritmo foi proposto por Bentley *et al.* (1978) para a geração do diagrama de Voronoi.

Um exemplo de algoritmo sequencial projetado por essa técnica é o algoritmo de Bohler *et al.* (2014), para a construção do diagrama de Voronoi. Exemplos de algoritmos que utilizam paralelismo projetados por essa técnica incluem o algoritmo de Wu *et al.* (2014), para a geração da triangulação de Delaunay, e de Dong *et al.* (2009), para a geração do diagrama de Voronoi.

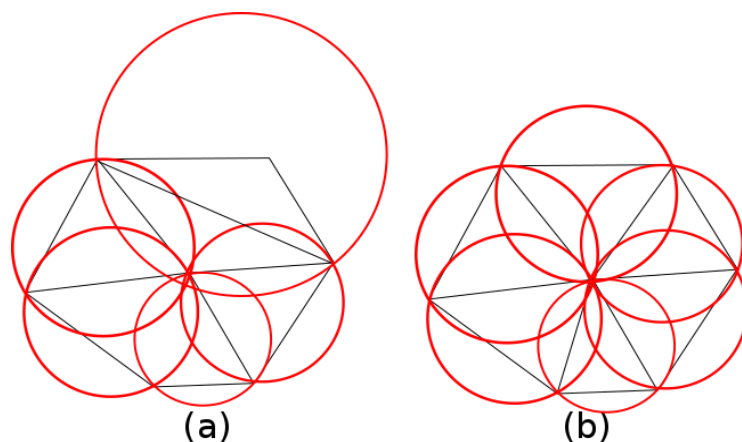
Também, existem algoritmos propostos para que a fase de intercalação dos subconjuntos possa ser realizada de maneira eficiente. Chen (2010) propôs um algoritmo paralelo em que descreve como realizar a fase de intercalação dos subconjuntos para a geração da tetraedrização de Delaunay. Isso porque, em algoritmos recursivos, pode-se realizar tarefas de maneira repetida. Além disso, nesses algoritmos, realiza-se a chamada de pilha do sistema operacional, o que pode aumentar o consumo de memória alto.

### 3.3 ALGORITMOS POR MUDANÇA GLOBAL DE ARESTAS

Em um algoritmo por mudança global de arestas, é dada uma tesselação inicial qualquer. Para a geração dessa tesselação, são realizados sucessivos testes de circuncírculo (ou circunferência, em estruturas tridimensionais).

Em malhas bidimensionais, arestas inválidas para a triangulação de Delaunay são excluídas e arestas válidas são criadas. Essas arestas devem ser válidas para a triangulação de Delaunay e são formadas pela inclusão de adjacências de vértices do triângulo excluído. Após a inserção de um novo triângulo na malha, outras arestas podem se tornar inválidas na triangulação de Delaunay. Por esse motivo, arestas de triângulos adjacentes aos triângulos criados devem ser verificados novamente, mesmo que já tenham sido previamente verificados.

Um exemplo de algoritmo por mudança global de arestas pode ser observado na Figura 4. Na Figura 4(a), é apresentada uma triangulação inicial, em que circuncírculos que contêm pontos de outros triângulos estão em vermelho. Após a troca de arestas inválidas, na Figura 4(b), pode ser observada a triangulação de Delaunay gerada.



**Figura 4:** Construção da triangulação de Delaunay por mudança global de arestas.



O primeiro algoritmo por mudança global de arestas para gerar a triangulação de Delaunay foi proposto por Lawson (1977). Para estruturas  $d$ -dimensionais, um algoritmo utilizando essa técnica foi apresentado por Edelsbrunner e Shah (1996). Cao *et al.* (2014) apresentaram um algoritmo paralelo para a construção da tetraedrização de Delaunay utilizando essa técnica.

Navarro *et al.* (2011) e Cao *et al.* (2014) apresentaram algoritmos paralelos para a construção da triangulação de Delaunay e para a tetraedrização de Delaunay, respectivamente, por mudança global de arestas. Em testes, Liu e Snoeyink (2005) observaram que, no algoritmo por mudança global de arestas, foi necessário um número maior de exclusões de tetraedros após a inserção de um novo ponto quando comparado com o algoritmo de Bowyer-Watson. De acordo com Liu e Snoeyink (2005), isso ocorre porque muitos tetraedros criados pela troca de faces inválidas para a malha, após a inserção de um novo ponto, são quase que imediatamente destruídos por outras trocas de faces. Dessa maneira, o algoritmo incremental de Bowyer-Watson é mais eficiente quando comparado ao algoritmo por mudança global de arestas. Porém, o algoritmo de Bowyer-Watson é menos robusto que o algoritmo de Lawson (1977) por mudança global de arestas. Segundo Shewchuk (1997), falhas são raras no algoritmo de Lawson (1977) em comparação ao número de falhas que ocorrem no algoritmo de Bowyer-Watson.

### 3.4 ALGORITMOS POR LINHA DE VARREDURA (SWEEPLINE)

Em algoritmos por linha de varredura, ocorre, inicialmente, uma ordenação dos pontos. Essa ordenação ocorre, geralmente, para o eixo das ordenadas ( $y$ ) no plano. Logo após, utiliza-se uma linha de varredura para percorrer o domínio. Um ponto é inserido na malha quando a linha de varredura passa pelo ponto. A varredura é realizada até que todos os pontos sejam inseridos no domínio.

O algoritmo de Fortune (1986) foi o primeiro por linha de varredura para a geração do diagrama de Voronoi. Esse algoritmo é  $O(n \lg n)$ , para um conjunto de  $n$  pontos no plano.

Exemplos de algoritmos sequenciais por linha de varredura incluem os algoritmos apresentados por Jin *et al.* (2006), para a geração do diagrama de Voronoi, e o algoritmo de Zalik (2005), para a construção da triangulação de Delaunay. Não foram encontrados algoritmos por linha de varredura utilizando paralelismo. Como notaram Goodrich e Ó'Dúnlaing (1985), a técnica por linha de varredura parece ser inerentemente sequencial.

### 3.5 ALGORITMOS POR *LIFT MAPPING*

Na classe de algoritmos por *lifting map*, os pontos que estão em uma estrutura de dimensão  $d$  são projetados para uma estrutura de dimensão  $d+1$ . Logo após essa projeção, calcula-se o fecho convexo para essa estrutura  $(d+1)$ -dimensional gerada. Basicamente, o fecho convexo é a menor região convexa a ser gerada para um conjunto de pontos. Construir o fecho convexo de uma estrutura  $(d+1)$ -dimensional é equivalente à computação da malha de Delaunay ou de Voronoi  $d$ -dimensional.

O próximo passo é a projeção do fecho convexo de dimensão  $d+1$  para uma dimensão  $d$ . Nessa projeção dos pontos de uma estrutura  $d+1$  dimensional para uma dimensão  $d$ , é gerada a malha de Delaunay ou de Voronoi. Por esse motivo, o custo computacional dos algoritmos por *lifting map* é proporcional ao custo computacional do algoritmo para a construção do fecho convexo em dimensão  $d+1$  utilizado. O primeiro algoritmo por *lifting map* foi proposto por Brown (1979) para a construção do diagrama de Voronoi. Outro exemplo de algoritmo por *lifting map* para a construção da triangulação de Delaunay é o algoritmo de Barber *et al.* (1996).

A fase de construção do fecho convexo em dimensão  $d+1$  pode ser paralelizada. A aplicação do algoritmo proposto por Day (1991), por exemplo, após a conversão do fecho convexo tridimensional gerado, geraria uma malha de Delaunay bidimensional. As implementações de algoritmos paralelos para a construção do diagrama de Voronoi ou da triangulação de Delaunay referem-se a algoritmos paralelos para a construção do fecho convexo do conjunto de pontos inicial.

### 3.6 ALGORITMOS POR *GIFT WRAPPING*

Em algoritmos por *gift wrapping* para a geração da triangulação de Delaunay, primeiramente, constrói-se um triângulo inicial, que atenda à propriedade de Delaunay. Para a construção do triângulo inicial, busca-se um ponto  $a$ , cuja coordenada seja mínima no eixo das ordenadas ( $y$ ), no plano. Um segundo ponto, que chamaremos de  $b$ , deve ser procurado no domínio, de forma que todos os pontos do domínio estejam à esquerda do segmento  $\overline{ab}$ . Finalmente, busca-se por um ponto  $c$ , de forma que  $c$  esteja à esquerda do segmento  $\overline{ab}$ . Com isso, um triângulo inicial  $\triangle abc$  é formado. A cada passo, novos triângulos de Delaunay válidos são encontrados.

Um novo triângulo de Delaunay válido é gerado, buscando-se um ponto  $p$  não inserido na malha que esteja à esquerda de uma aresta  $\overline{de}$  pertencente a um triângulo descoberto previamente e que esteja na borda da triangulação. O ponto  $p$  é unido aos pontos finais da aresta  $\overline{de}$  para formar novos triângulos válidos para a malha de Delaunay. Esse passo ocorre até que todos os pontos estejam inseridos na triangulação de Delaunay. Esse processo pode ser estendido para estruturas  $d$ -dimensionais, utilizando-se  $d+1$  pontos para a construção do politopo inicial.

De acordo com Cheng *et al.* (2013), o primeiro algoritmo proposto para a geração da triangulação de Delaunay foi proposto por Fredericket *et al.* (1970). Esse algoritmo foi projetado por *gift wrapping*. Outro exemplo de algoritmo empregando a técnica de *gift wrapping* é o algoritmo de Dwyer (1989) para a construção do diagrama de Voronoi. Também, de acordo com Cheng *et al.* (2013), os algoritmos por *gift wrapping* são lentos.

### 3.7 ALGORITMOS POR VARREDURA POR CÍRCULO (SWEEP-CIRCLE)

Na técnica de varredura por círculo, utiliza-se um círculo que é aumentado a partir de um ponto fixo no domínio. Em um algoritmo por varredura por círculo, o domínio é percorrido utilizando-se um círculo  $C$ . O círculo  $C$  tem o centro fixado em um ponto  $p$  do fecho convexo do conjunto inicial de pontos. Para que ocorra um percurso no plano, ocorre um aumento do diâmetro do círculo  $C$ . Para isso, inicialmente, os pontos são ordenados de acordo com suas distâncias ao ponto  $p$ . Dessa maneira, um ponto é inserido na malha quando o círculo  $C$  passa pelo ponto. Essa varredura é finalizada quando todos os pontos do domínio estiverem na tesselação de Delaunay ou no diagrama de Voronoi.

O primeiro algoritmo para a geração do diagrama de Voronoi utilizando a técnica de varredura por círculo foi desenvolvido por Dehne e Klein (1987) e foi uma variação da técnica de linha de varredura. Esse algoritmo apresenta o comportamento assintótico  $O(n \lg n)$  (Dehne e Klein, 1987).

O algoritmo sequencial de Biniáz e Dastghaibyfar (2012), por exemplo, combina as técnicas de varredura por círculo e de mudança global de arestas para a geração da triangulação de Delaunay. Xin *et al.* (2013), por exemplo, combina as técnicas de varredura por círculo e de mudança global de arestas para a geração da triangulação de Delaunay. Xin *et al.* (2013) apresentaram também um algoritmo em paralelo projetado pela a técnica de varredura por círculo para a construção do diagrama de Voronoi.

## 4. DISCUSSÃO

A escolha da técnica adequada para a construção das malhas de Delaunay ou de Voronoi é uma fase importante para a implementação dessas estruturas. Fatores que, em geral, devem ser considerados no projeto de algoritmos para essas malhas são o baixo tempo de execução, a facilidade de paralelização, a baixa ocupação de memória e a facilidade de implementação e de manutenção do código. Também, combinações podem ser utilizadas para a construção de novos algoritmos para a geração da tesselação de Delaunay ou do diagrama de Voronoi.

Em geral, algoritmos incrementais são os mais utilizados, uma vez que são eficientes e de fácil implementação. Nessa classe, estão os algoritmos no estado da arte para a geração da tesselação de Delaunay em implementações sequenciais. Também, algoritmos incrementais são implementados na CGAL (2018), uma biblioteca com rotinas no estado da arte em geometria computacional. Em relação a algoritmos paralelos, devido às características inerentes da técnica, algoritmos por divisão e conquista são mais populares para a geração dessas malhas.

## 5. CONSIDERAÇÕES FINAIS

Neste trabalho, foi apresentada uma visão geral das técnicas utilizadas no projeto de algoritmos para a geração da tesselação de Delaunay e do diagrama de Voronoi. As técnicas de divisão e conquista, de inserção incremental, de mudança global de arestas, por linha de varredura (*sweep line*), por *lifting map*, por *gift wrapping* e por varredura por círculo foram abordadas. Entre os algoritmos sequenciais estudados, a maior quantidade de algoritmos propostos para geração das malhas de Delaunay e de Voronoi são incrementais. Em particular, algoritmos no estado da arte para a geração da triangulação de Delaunay são incrementais.

Como continuidade na investigação neste contexto, pretendemos avaliar se um algoritmo incremental com inserção de pontos dada na ordem BRIO (Amenta *et al.*, 2003), em conjunto com a sequência *cut-longest-edge kd-tree* (Liu *et al.*, 2013) pode gerar custos de execução menor na geração da tesselação de Delaunay que com a utilização da política *spatial middle* empregada na última versão da biblioteca CGAL (2018). O algoritmo de Tianyun (2016) é o provável estado da arte na geração da triangulação de

Delaunay. Também, pretendemos implementar versões paralelas de algoritmos para a geração da tesselação de Delaunay.

## REFERÊNCIAS

- AMENTA, N.; CHOI, S.; ROTE, G. **Incremental constructions con BRIO**. Proceedings 19th annual symposium on Computational geometry, ACM, San Diego, 2003, p. 211-219.
- BARBER, C. B.; DOBKIN, D. P. and HUHDANPA, H. The quickhull algorithm for convex hulls. **ACM Transactions on Mathematical Software**, n. 22, vol. 4, 1996, p. 469-483.
- BENTLEY, J. L.; WEIDE, B. W.; YAO, A. C. **Optimal expected-time algorithms for closest point problems**. 16th Proceedings Allerton, Monticello, 1978, p. 843-851.
- BINIAZ, A.; DASTGHAIBYFARD, G. A faster circle-sweep Delaunay triangulation algorithm. **Advances in Engineering Software**, n. 43, vol. 1, 2012, p. 1-13.
- BOHLER, C.; LIU, C. H.; PAPADOPOULOU, E.; ZAVERSHYNSKYI, M. A randomized divide and conquer algorithm for higher-order abstract Voronoi diagrams. **Lecture Notes in Computer Science**, 8889, 2014, p. 27-37.
- BOWYER, A. Computing Dirichlet tessellations. **Computer Journal**, 24, 1981, p. 162-166.
- BROWN, K. Q. Voronoi diagrams from convex hulls. **Information processing letters**, n. 9, vol. 5, 1979, p. 223-228.
- CAO, T.-T.; NANJAPPA, A.; GAO, M.; TAN, T.-S. **A GPU accelerated algorithm for 3D Delaunay triangulation**. 18th SIGGRAPH, San Francisco, 2014, p. 47-54.
- CHEN, M. B. **The merge phase of parallel divide-and-conquer scheme for 3D Delaunay triangulation**, Proceedings International Symposium on Parallel and Distributed Processing with Applications, Taipei, 2010, p. 224-230.
- CHENG, S. W.; DEY, T. K.; SHEWCHUK, J. R. **Delaunay Mesh Generation**. Boca Raton: CRC Computer and Information Science Series, CRC, 2013.
- DAY, A. M. Parallel implementation of 3D convex-hull algorithm, **Computer-Aided Design**, n. 23, vol. 3, 1991, p. 177-188.
- DEHNE, F.; KLEIN, R. **A sweepcircle algorithm for Voronoi diagrams**, in Proceedings WG'87, Springer Verlag, Erlangen, vol. 314, 1987, p. 59-70.
- DONG, L.; WU, Y.; ZHOU, S. **Constructing the Voronoi diagram of planar point set in parallel**, in CiSE 2009, Wuhan, 2009.
- DWYER, R. A. **Higher-dimensional Voronoi diagrams in linear expected time**, Proceedings 5th annual symposium on Computational geometry, ACM, NY, 1989, p. 326-333.

EDELSBRUNNER, H.; SHAH, N. R. Incremental topological flipping works for regular triangulations. **Algorithmica**, vol. 15, 1996, p. 223-241.

FORTUNE, S. **A sweepline algorithm for Voronoi diagrams**, Proceedings of the 2nd annual symposium on Computational geometry, ACM, NY, 1986, p. 313-322.

FREDERICK, C. O.; WONG, Y. C.; EDGE, F. W. Two-dimensional automatic mesh generation for structural analysis, **International Journal for Numerical Methods in Engineering**, n. 2, vol. 1, 1970, p. 133-144.

GONZAGA DE OLIVEIRA, S. L. **Algoritmos e seus fundamentos**. Lavras: Ed. UFLA, 2011.

GONZAGA DE OLIVEIRA, S. L. **Introdução à geração de malhas triangulares**, Notas em Matemática Aplicada, São Carlos: Sociedade Brasileira de Matemática Aplicada e Computacional - SBMAC, 2015.

GONZAGA DE OLIVEIRA, S. L.; NOGUEIRA, J. R.; TAVARES, J. M. R. S. A systematic review of algorithms with linear-time behaviour to generate Delaunay and Voronoi tessellations. **Computer Modeling in Engineering and Sciences**, n. 100, vol. 1, 2014, p. 31-57.

GONZAGA DE OLIVEIRA, S. L.; NOGUEIRA, J. R. An evaluation of point-insertion sequences for incremental Delaunay tessellations. **Computational & Applied Mathematics**, v. 37, 2018, p. 641-674.

GONZAGA DE OLIVEIRA, S. L.; ROBAINA, D. T.; BRANDÃO, D. N.; KISCHINHEVSKY, M.; OLIVEIRA, G. **An experimental assessment of three point-insertion sequences for 3-D incremental Delaunay tessellations (to appear)**. In: The International Conference on Computational Science (ICCS), Wuxi. Lecture Notes in Computer Science. Berlin: Springer International Publishing, 2018.

GOODRICH, M. T.; O'DÚNLAIN, C. Fast parallel algorithm for Voronoi diagrams. Courant Institute of Mathematical Sciences, New York, Technical report CSD-TR 538, 1985.

JIN, L.; KIM, D.; MU, L.; KIM, D.-S.; HU, S.-M. A sweepline algorithm for Euclidean Voronoi diagram of circles, **Computer-Aided Design**, vol. 38, 2006, p. 260-272.

LAWSON, C. L. Software for surface interpolation, Technical report, National Aeronautics and Space Administration, JPL Publication, Washington, 1977, p. 77-30.

LEE, S.; PARK, C.-I.; PARK, C.-M. **An improved parallel algorithm for Delaunay triangulation on distributed memory parallel computers**, Proceedings Advances in Parallel and Distributed Computing, 1997, p. 131-138.

LEE, D. T.; SCHACHTER, B. J. **Two algorithms for constructing a Delaunay triangulation**, International Proceedings 16th Annual Journal of Computer and Information Sciences, n. 9, vol. 3, 1980, p. 219-242.



LIU, Y.; SNOEYINK, J. **A comparison of five implementations of 3D Delaunay Tessellation**, in Combinatorial and Computational Geometry. J.E. Goodman and J. Pach and E. Welzl (ed.), MSRI, Cambridge, vol. 52, 2005, p. 439 - 458.

LIU, J.-F.; YAN, J.-H.; LO, S. H. A new insertion sequence for incremental Delaunay triangulation. **Acta Mechanica Sinica**, n. 29, vol. 1, 2013, p. 99-109.

LO, S. H. Delaunay triangulation of non-uniform point distributions by means of multi-grid insertion. **Finite Elements in Analysis and Design**, vol. 63, 2013, p. 8-22.

NAVARRO, C. A.; HITSCHFELD-KAHLER, N.; SCHEIHING, E. **A parallel GPU-based algorithm for Delaunay edge-flips**, 27th EuroCG 2011, Morschach, 2011, p. 28-30.

NOGUEIRA, J. R.; GONZAGA DE OLIVEIRA, S. L. **Técnicas Aplicadas em Projetos de Algoritmos para a Geração da Tesselção de Delaunay e do Diagrama de Voronoi**. In: Anais do XX Encontro Nacional de Modelagem Computacional. Nova Friburgo: IPRJ-UERJ, 2017.

PREPARATA, F. P.; SHAMOS, M. I. **Computational Geometry - An Introduction**, Springer-Verlag, New York, 1985.

SHAMOS, M. I.; HOEY, D. **Closest-point problems**, in Proceedings 16th Annual Symposium on Foundations of Computer Science, Washington, 1975, p. 151-162.

SHEWCHUK, J. R. **Delaunay Refinement Mesh Generation**. PhD thesis, School of Computer Science, Carnegie Mellon University, 1997.

THE CGAL PROJECT. **The CGAL Project, CGAL Editorial Board, Edition 4.11**. Disponível em <<http://doc.cgal.org/4.11/Manual/packages.html>>. Acesso em: 28 de julho de 2018.

TIANYUN, S.; WEN, W.; ZHIHAN, L.; WEI, W.; XINFANG, L. Rapid Delaunay triangulation for randomly distributed point cloud data using adaptive Hilbert curve. **Computers and Graphics**. vol. 54, 2016, 65-74.

WATSON, D. F. Computing the  $n$ -dimensional Delaunay tessellation with application to Voronoi polytopes. **Computer Journal**, n. 24, vol. 2, 1981, p. 167-172.

XI, J. H.; ZUO, S. C. An Improved Algorithm Based on Incremental Insertion in Delaunay Triangulation. **Applied Mechanics and Materials**, n. 397, vol. 400, 2013, p. 1691-1694.

XIN, S.-Q.; WANG, X.; XIA, J.; MUELLER-WITTIG, W.; WANG, G.-J.; HE, Y. Parallel computing 2D Voronoi diagrams using untransformed sweepcircles, **Computer-Aided Design**, n. 45, vol. 2, 2013, p. 483 - 493.

XU, P.; CHEN, Z.; DENG, X.; JIANPING, Y. An efficient implementation of incremental construction Voronoi region. **International Journal of Advancements in Computing Technology**, n. 4, vol. 2, 2012, p. 230-237.



WU, W.; RUI, Y.; SU, F.; CHENG, L.; WANG, J. Novel parallel algorithm for constructing Delaunay triangulation based on a twofold-divide-and-conquer scheme, **GIScience and Remote Sensing**, vol. 51, 2014, p. 537-554.

ZALIK, B. An efficient sweep-line Delaunay triangulation algorithm, **Computer-Aided Design**, n. 37, vol. 10, 2005, p. 1027-1038.