

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GUSTAVO TIMMERMANS PIRES DA SILVA

**UM ALGORITMO PARA GERAÇÃO DE ESTRUTURAS DE SUPORTE EM
ÁRVORE PARA MANUFATURA ADITIVA**

CURITIBA

2022

GUSTAVO TIMMERMANS PIRES DA SILVA

**UM ALGORITMO PARA GERAÇÃO DE ESTRUTURAS DE SUPORTE EM
ÁRVORE PARA MANUFATURA ADITIVA**

**An algorithm for the generation of tree support structures for additive
manufacturing**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção
do título de Bacharel em Engenharia de
Computação do Curso de Bacharelado em
Engenharia de Computação da Universidade
Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Ricardo Dutra da Silva

CURITIBA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

GUSTAVO TIMMERMANS PIRES DA SILVA

**UM ALGORITMO PARA GERAÇÃO DE ESTRUTURAS DE SUPORTE EM
ÁRVORE PARA MANUFATURA ADITIVA**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção
do título de Bacharel em Engenharia de
Computação do Curso de Bacharelado em
Engenharia de Computação da Universidade
Tecnológica Federal do Paraná.

Data de aprovação: 30/junho/2022

Ricardo Dutra da Silva
Doutorado
Universidade Tecnológica Federal do Paraná

Neri Volpato
Doutorado
Universidade Tecnológica Federal do Paraná

Rodrigo Minetto
Doutorado
Universidade Tecnológica Federal do Paraná

CURITIBA
2022

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. Ricardo Dutra da Silva, pela grande paciência em todo momento desta trajetória, pela força de incentivo quando parecia que não se podia enxergar uma luz no fim do túnel, pelo entusiasmo em relação ao projeto que conseguiu disputar com o meu, pela disponibilidade de seu tempo, não apenas para questões, mas também para conceptualizações que permitiram extrair o máximo deste projeto. O resultado deste trabalho não teria sido possível sem sua orientação. Muito obrigado pela parceria.

RESUMO

A manufatura aditiva consiste em construir objetos a partir da deposição de camadas sucessivas de material. Dependendo da tecnologia de manufatura, regiões de camadas que não possuem sustentação direta em uma camada inferior podem sofrer deformação e, por este motivo, a construção de estruturas adicionais de suporte é essencial para fabricar corretamente objetos. No entanto, as estruturas de suporte podem aumentar muito a quantidade de material e incorrer no aumento dos tempos de processamento e fabricação. Esta monografia descreve um método para computação de estruturas de suporte em formato de árvores, visando diminuir a quantidade de material necessário para fabricação. A construção de árvores é baseada em procurar conexões entre pontos de sustentação para atingir um menor custo de material. A procura por conexões pode ser bastante custosa considerando uma busca exaustiva. Por este motivo, o método desenvolvido utiliza uma busca em largura por pares de pontos de sustentação mais próximos, com o objetivo de diminuir o tempo computacional e obter também a redução da quantidade de material. Experimentos foram realizados para diferentes modelos tridimensionais para mostrar a eficácia do método proposto.

Palavras-chave: estruturas de suporte; estrutura de suporte em árvore; manufatura aditiva; ângulo de autossuporte; suporte pós-fatiamento.

ABSTRACT

Additive manufacturing consists of building objects from the deposition of successive layers of material. Depending on the manufacturing technology, layer regions that do not have direct support on a lower layer can experience deformation, and for this reason, building additional support structures is essential to manufacture an object correctly. However, support structures can greatly increase the amount material and incur in increasing processing and fabrication times. This monograph describes a method for computing support structures in the form of trees, aiming to reduce the amount of material needed for fabrication. Tree construction is based on looking for connections between support points to achieve lower material cost. The search for connections can be quite expensive considering an exhaustive search, for this reason, the developed method uses a breadth first search to look for pairs of closest support points, with the objective of reducing the computational time and also obtaining the reduction of the amount of material. Experiments were carried out for different three-dimensional models to show the effectiveness of the proposed method.

Keywords: support structures; tree support structure; additive manufacturing; self-support angle; post-slicing support.

LISTA DE FIGURAS

Figura 1 – Exemplo de criação de estruturas de suporte.	11
Figura 2 – Diferentes representações de um modelo para análise de suporte	12
Figura 3 – Exemplo de estrutura de suporte em forma de árvore.	13
Figura 4 – Polígono formado por uma sequência cíclica de vértices.	14
Figura 5 – Operações sobre polígonos.	15
Figura 6 – Posição de um ponto relativa a um polígono.	16
Figura 7 – Processo de fatiamento: um modelo é seccionado por um conjunto de planos para formar fatias de polígonos.	19
Figura 8 – Resultado final do fatiamento: o conjunto de fatias do topo E_1 até a base E_N é usado como entrada para os algoritmos de computação de suporte.	19
Figura 9 – Diferentes casos de sustentação de uma fatia considerando a fatia inferior adjacente.	20
Figura 10 – Possível separação das estruturas de suporte devido a imprecisões nas adjacências. Ao invés de ser obtida uma única região, são obtidas três regiões (representadas em verde, alaranjado e azul).	21
Figura 11 – Processo de filtragem.	21
Figura 12 – Ângulo de autossuporte: para determinadas inclinações não há necessidade de estrutura auxiliar de suporte.	22
Figura 13 – Remoção de regiões suspensas considerando o ângulo de autossuporte.	23
Figura 14 – Remoção de regiões de suporte utilizando o ângulo de autossuporte. Caso com suporte próximos de uma borda de polígonos.	24
Figura 15 – Exemplo de sequência de etapas no processo de construção de uma árvore.	25
Figura 16 – Transformação das áreas de suporte em pontos iniciais para a construção das estruturas de árvores.	27
Figura 17 – Definição de cone de sustentação considerando o ângulo de autossuporte.	28
Figura 18 – Junção de um nó à base: o ponto de intersecção da projeção vertical de um ponto P , com a base do modelo ou com um polígono de uma fatia intermediária, corresponde a uma junção mais próxima.	28
Figura 19 – Junção V entre nós em uma mesma camada.	29
Figura 20 – Junção V entre nós em diferentes camadas.	30

Figura 21 – Junção de extensão: nó interior ao cone de um ponto que produz a conexão mais próxima.	31
Figura 22 – Agrupamento de pontos em células.	31
Figura 23 – Exemplo de busca em largura onde o nó mais próximo está em uma célula vizinha.	32
Figura 24 – Busca em largura por um nó mais próximo.	33
Figura 25 – Uma junção V encontrada limita a busca por outras junções. As camadas consideradas devem estar a uma distância vertical menor ou igual ao raio do círculo definido pela junção V.	34
Figura 26 – Pelo ângulo de autossuporte é possível determinar um limite para camadas de busca.	34
Figura 27 – Se a projeção de um ponto atinge um objeto que se comporta como base, o limite de busca para camadas é determinado pela camada da mesma base.	35
Figura 28 – Cone definido pelo ângulo de autossuporte que limita a fronteira de busca nas camadas.	35
Figura 29 – Composição de todos os limites parciais no limite final de busca.	36
Figura 30 – Atualização dos limites de busca.	36
Figura 31 – Validação de ramo por interpolação dos nós.	37
Figura 32 – Subdivisão de arestas e criação de polígonos em nós para formar regiões a serem impressas.	38
Figura 33 – Moldelos 3D utilizados nos experimentos.	40
Figura 34 – Comparação de estratégias de suporte: (a) suporte em toda região suspensa; (b) suporte apenas em inclinações que não atendem o ângulo de autossuporte; e (c) suporte usando estrutura de árvore.	40
Figura 35 – Efeito do ângulo mínimo para os ramos na árvore resultante.	41
Figura 36 – Geração de suporte em árvore do modelo Turbina.	41
Figura 37 – Geração de suporte em árvore do modelo Túnel.	42
Figura 38 – Geração de suporte em árvore do modelo Cilindro Dentado.	42
Figura 39 – Geração de suporte em árvore do modelo Dinossauro.	43

LISTA DE TABELAS

Tabela 1 – Tempos para a geração de suporte comparando estratégias.	43
Tabela 2 – Complexidade do modelo após fatiamento e número de nós gerados.	44
Tabela 3 – Número de camadas avaliadas por nó.	44
Tabela 4 – Número de candidatos avaliados para melhor nó.	45
Tabela 5 – Número de células percorridas por nó.	45
Tabela 6 – Total de nós analisados para produção de um nó.	46
Tabela 7 – Volume relativo de material para a geração de suporte comparando estratégias.	46

SUMÁRIO

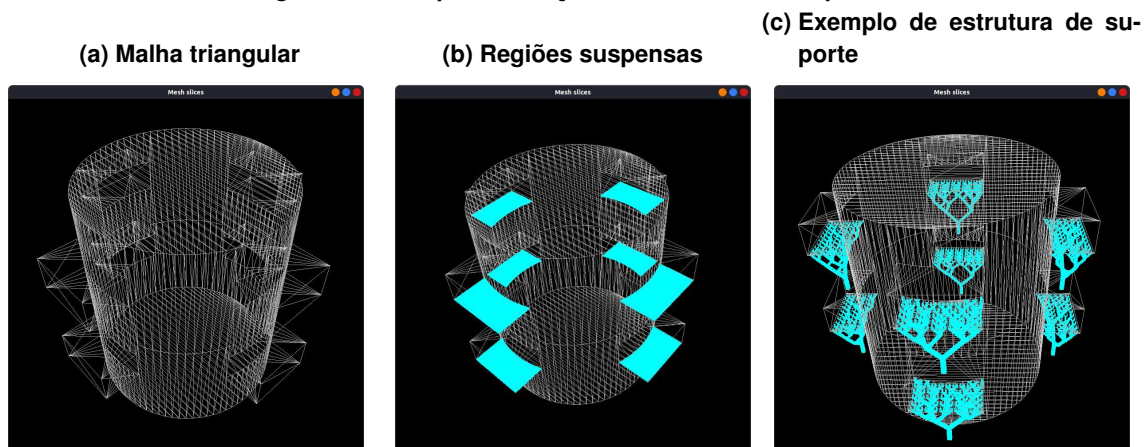
1	INTRODUÇÃO	11
2	REVISÃO DE LITERATURA	14
2.1	Polígonos e operações	14
2.2	Trabalhos relacionados	16
3	METODOLOGIA	18
3.1	Fatiamento e notação	18
3.2	Algoritmo básico para suporte	19
3.3	Algoritmo considerando autossuporte	22
3.4	Algoritmo baseado em árvore	24
3.4.1	Pontos de sustentação	26
3.4.2	Tipos de nós de junção	27
3.4.2.1	<u>Junção à base</u>	27
3.4.2.2	<u>Junção em V</u>	29
3.4.2.3	<u>Junção de extensão</u>	30
3.4.3	Busca pela junção mais próxima	31
3.4.4	Validação de nós de junção	36
3.4.5	Construção do suporte a partir do grafo	37
4	RESULTADOS	39
4.1	Exemplos de árvores	39
4.2	Análise de tempo do algoritmo	42
4.3	Análise de volume dos suportes	46
5	CONSIDERAÇÕES FINAIS	48
	REFERÊNCIAS	49

1 INTRODUÇÃO

A manufatura aditiva é um processo de fabricação de objetos tridimensionais através da deposição de material em camadas. O processo de planejamento da manufatura aditiva consiste em diferentes etapas que envolvem: orientação, posicionamento e escala de um modelo; computação de estruturas de suporte; fatiamento de malhas triangulares, planejamento de trajetórias e de preenchimento (VOLPATO; CARVALHO, 2017).

Esta monografia considera a etapa de computação de regiões de suporte. A entrada do processo de planejamento é um modelo tridimensional, comumente representado por uma malha de triângulos (Figura 1a). Alguns modelos podem possuir regiões suspensas que, por não terem sustentação no próprio modelo, dificultam ou mesmo impossibilitam a fabricação do modelo de entrada diretamente (Figura 1b). Nessas regiões suspensas, é necessária a criação de estruturas auxiliares de sustentação para permitir a fabricação correta de um modelo. A Figura 1c apresenta um exemplo de estrutura de suporte em forma de árvore para as mesmas regiões suspensas com o objetivo de reduzir o volume necessário.

Figura 1 – Exemplo de criação de estruturas de suporte.

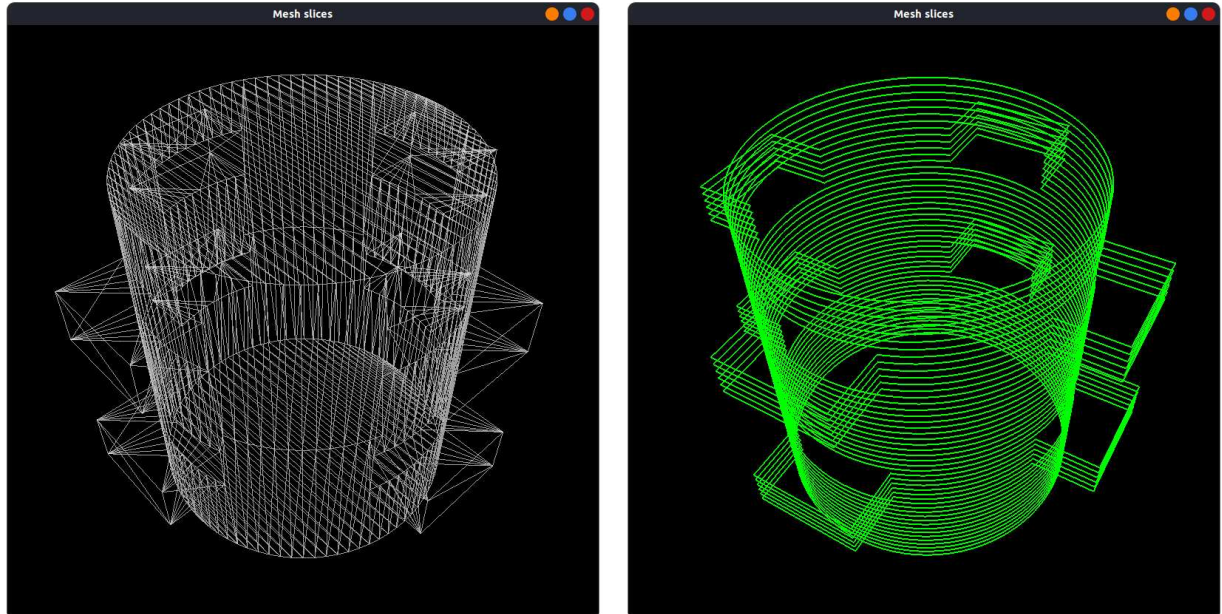


Fonte: Autoria própria.

A identificação das regiões de suporte pode ser realizada através da malha triangular de um modelo (Figura 2a) ou por polígonos resultantes do fatiamento do modelo (Figura 2b) (VOLPATO; CARVALHO, 2017). Métodos baseados na malha consideram as normais de triângulos para identificar se o triângulo pertence a uma região suspensa e criar estruturas auxiliares de suporte. Nos métodos baseados em polígonos provenientes do fatiamento, as regiões de suporte são identificadas através de operações entre polígonos de camadas adjacentes (CHALASANI; JONES; ROSCOE, 1995).

A criação de estruturas de suporte tem algumas implicações, como o aumento do tempo de processamento para o planejamento do processo e o aumento do tempo e da quantidade de material necessários para fabricação (THOMAS, 2009). Assim, melhorias no processo de geração de suportes devem ser consideradas. Por exemplo, para minimizar a quantidade de

Figura 2 – Diferentes representações de um modelo para análise de suporte
(a) Malha triangular **(b) Modelo fatiado**



Fonte: Autoria própria.

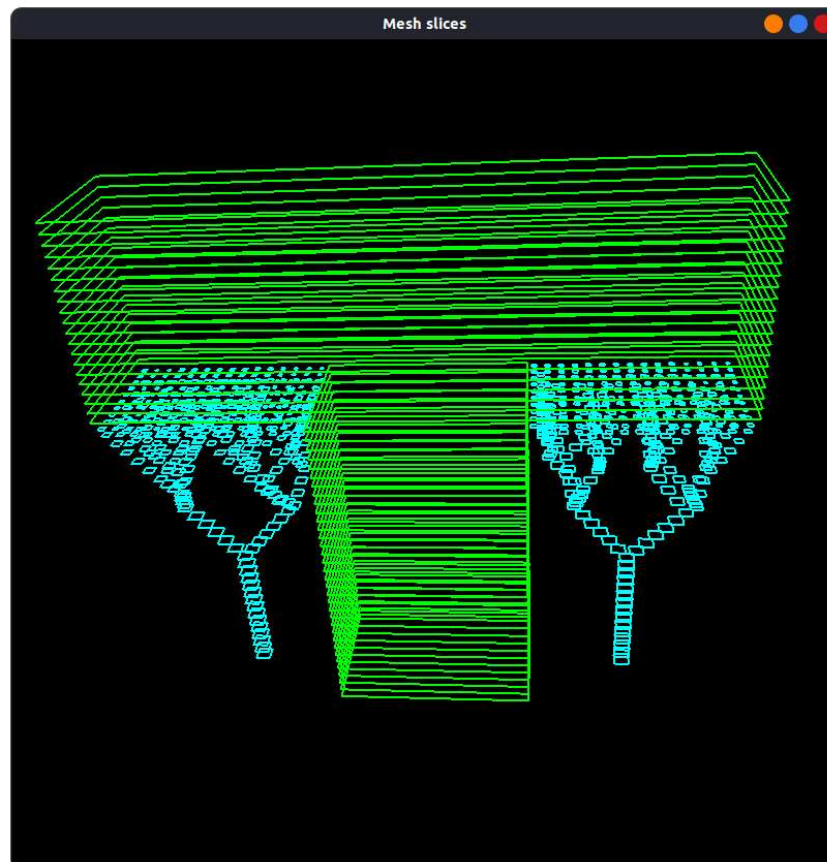
material, é possível explorar a orientação do modelo e as formas das estruturas de suporte (JIANG; XU; STRINGER, 2018).

Esta monografia considera a criação de estruturas de suporte baseadas em formatos de árvores para diminuir a quantidade de material necessário para impressão (Figura 3). O método apresentado para a criação de estruturas de árvores é baseado nas fatias de um modelo tridimensional de entrada. Sobre regiões suspensas das fatias, são amostrados pontos de sustentação que devem ser unidos por conexões formando ramos de árvores. No entanto, a criação de uma árvore que minimiza o volume de suporte está relacionada com o Problema da Árvore Mínima de Steiner (GILBERT; POLLAK, 1968), um problema que é da classe NP-difícil (VANEK; GALICIA; BENES, 2014).

Considerando a dificuldade do problema, o algoritmo apresentado usa uma heurística de busca por pontos mais próximos, ou seja, dado um ponto qualquer, um ponto mais próximo é buscado para formar uma junção. Uma estratégia de busca é proposta de forma a evitar uma busca exaustiva pelo ponto mais próximo e tornar o algoritmo de busca mais rápido. Para isso, o espaço de busca é subdividido em células e uma busca em largura é usada para procurar pontos mais próximos em células vizinhas.

Além do algoritmo para a criação de estruturas de árvores, são descritos também dois algoritmos básicos para a criação de estruturas de suportes: um algoritmo que considera todo volume abaixo de uma região suspensa como suporte e um algoritmo que remove deste volume as regiões autossuportadas (regiões suspensas mas em inclinações que podem ser diretamente sustentadas em camadas inferiores do próprio objeto). Experimentos foram realizados para analisar a quantidade de suporte necessário ao comparar as três estratégias, demons-

Figura 3 – Exemplo de estrutura de suporte em forma de árvore.



Fonte: Autoria própria.

trando os benefícios da estratégia baseada em árvore. Também foram realizados experimentos para comparar os tempos entre os métodos e para analisar características relacionadas com a complexidade da busca desenvolvida para o método de árvores. Embora o tempo do método baseado em árvores seja maior que os tempos para os dois outros métodos, para muitos modelos testados, o tempo não foi significativamente maior.

No Capítulo 2 são apresentados conceitos relacionados à metodologia desenvolvida e uma revisão de trabalhos relacionados. O Capítulo 3 descreve os três métodos considerados para a criação de estruturas de suporte. Resultados experimentais são mostrados no Capítulo 4. O Capítulo 5 expõe a conclusão do trabalho e perspectivas de desenvolvimentos futuros.

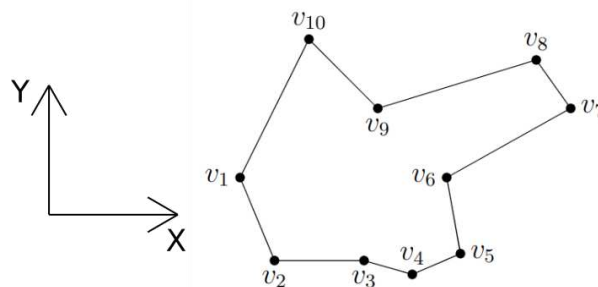
2 REVISÃO DE LITERATURA

Neste capítulo, inicialmente são revisados conceitos fundamentais para o entendimento dos algoritmos desenvolvidos neste trabalho (Seção 2.1). Na sequência, a Seção 2.2 apresenta uma revisão de trabalhos relacionados para a produção de estruturas de suporte.

2.1 Polígonos e operações

Neste trabalho, um polígono P é definido por uma sequência cíclica de n pontos no plano, $P = v_1, v_2, \dots, v_n$, de forma que vértices subsequentes (v_i, v_{i+1}) formam uma aresta do polígono e v_1 sucede v_n , quando $i = n$, formando a aresta (v_n, v_1) . Também é considerado que polígonos são simples, não havendo intersecção entre arestas além do ponto coincidente entre arestas adjacentes. A sequência de pontos de um polígono ainda é orientada em sentido anti-horário, de forma que o interior do polígono sempre está à esquerda de qualquer aresta orientada (v_i, v_{i+1}) . A Figura 4 ilustra uma sequência de vértices formando um polígono.

Figura 4 – Polígono formado por uma sequência cíclica de vértices.



Fonte: Autoria própria.

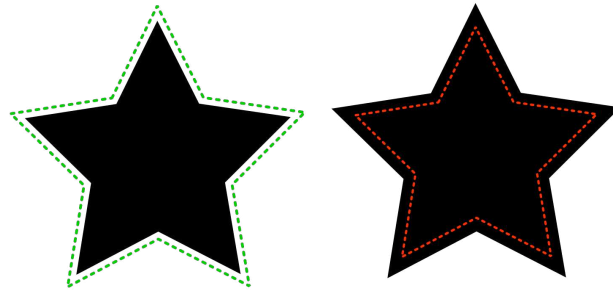
No desenvolvimento do método apresentado nesta monografia surgem operações importantes sobre polígonos, como *offset*, intersecção, união e subtração. A Figura 5 ilustra cada uma dessas operações. Para algoritmos que realizam cada uma dessas operações, o leitor pode verificar a literatura relacionada (SCHNEIDER; EBERLY, 2002; CHEN; MCMAINS, 2005).

A operação de *offset* de um polígono (Figuras 5a e 5b) resulta em um novo polígono com as arestas deslocadas, produzindo um polígono maior (dilatação) ou menor do que o original (erosão) (CHEN; MCMAINS, 2005).

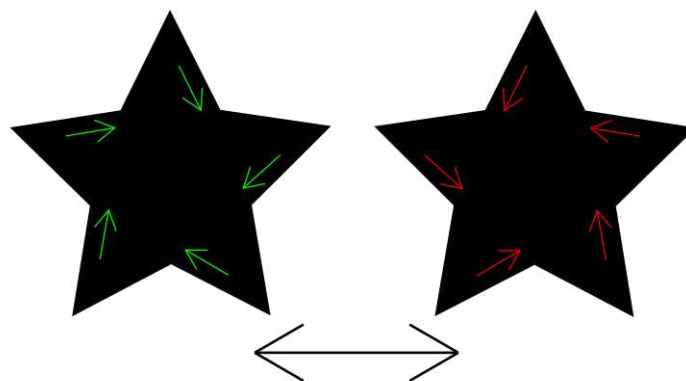
A negação \bar{P} de um polígono P corresponde à inversão de suas regiões internas e externas (Figura 5c). Considerando a representação de P como uma sequência de pontos com arestas orientadas, a negação pode ser realizada invertendo a ordem da sequência (SCHNEIDER; EBERLY, 2002), logo, se $P = v_1, v_2, \dots, v_n$ então $\bar{P} = v_n, v_{n-1}, \dots, v_1$.

A intersecção $P \cap Q$ de dois polígonos corresponde a um ou mais polígonos que delimitam as regiões comuns aos dois polígonos intersectados (Figura 5d). Este tipo de operação é

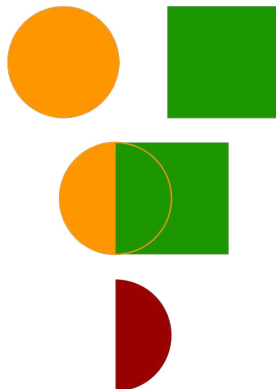
Figura 5 – Operações sobre polígonos.
 (a) Dilatação (b) Erosão



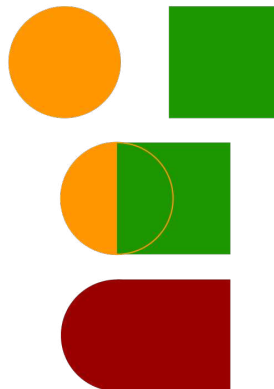
(c) Negação



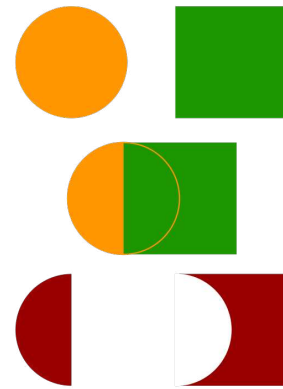
(d) Intersecção



(e) União



(f) Subtração



Fonte: Autoria própria.

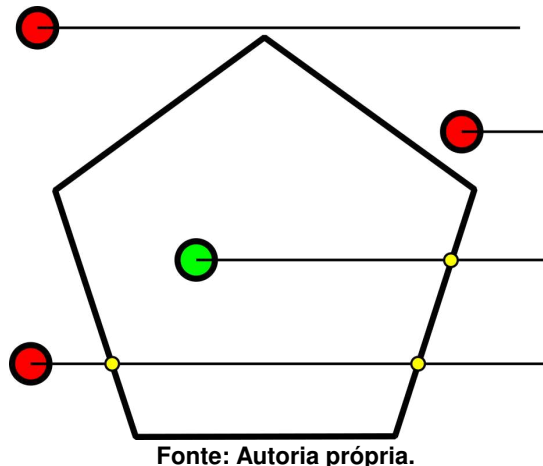
baseada em algoritmos de recorte, comumente usados em Computação Gráfica (O'ROURKE, 1994).

A Figura 5e ilustra a união de dois polígonos, que corresponde à união das regiões delimitadas pelos polígonos iniciais. A computação da união pode ser feita a partir das operações e negação e intersecção conforme mostra a regra de De Morgan, $P \cup Q = \overline{\overline{P} \cap \overline{Q}}$ (SCHNEIDER; EBERLY, 2002).

A diferença de dois polígonos corresponde à região do primeiro polígono que não é comum ao segundo polígono (Figura 5f). Com a união, é possível computar a diferença usando a negação e intersecção: $P \cap \overline{Q}$ (SCHNEIDER; EBERLY, 2002).

Uma operação importante para o desenvolvimento do método, apresentado no Capítulo 3, consiste em computar a intersecção de um segmento de reta com o modelo 3D. No caso do método descrito neste trabalho, esta operação se reduz à computação da posição relativa de um ponto com um polígono, se o ponto é interno ou externo ao polígono. A Figura 6 mostra exemplos de pontos internos e externos ao polígono. A operação pode ser implementada considerando o número de intersecções de uma reta semi-infinita partindo do ponto testado e seguindo em uma direção arbitrária, por exemplo, a do vetor $(1,0)$ (FIGUEIREDO; CARVALHO, 1991; O'ROURKE, 1994; SCHNEIDER; EBERLY, 2002). Um número de intersecções ímpar indica que o ponto é interno ao polígono, enquanto um número par indica que ele é externo.

Figura 6 – Posição de um ponto relativa a um polígono.



2.2 Trabalhos relacionados

Yu-an, Yong e Jian-zhong (2015) descrevem um método básico de construção de suporte, tanto interno quanto externo, baseado em dados das fatias de um modelo, usando operações morfológicas. O ângulo de autossuporte é empregado para determinar se uma área de um polígono parcialmente suspenso necessita realmente de suporte pois, dependendo da inclinação, camadas inferiores conseguem sustentar camadas superiores mesmo sem adição de estruturas adicionais de suporte. Para isso, pontos ao longo do perímetro externo da área suspensa são gerados e é calculada a menor distância até o polígono oferecendo suporte, na camada abaixo. Se todas as distâncias estiverem dentro de um limite estabelecido através do ângulo de autossuporte, não há necessidade de suporte, caso contrário, se faz necessário o suporte em toda a região. O algoritmo consiste em posicionar círculos de raio igual ao limite determinado pelo ângulo de autossuporte em cada um dos pontos selecionados no perímetro e

depois computar um envelope dos círculos. Caso o envelope possua alguma interseção com o polígono na fatia abaixo, a área é dada como necessitando de suporte.

Vanek, Galicia e Benes (2014) utilizam uma abordagem geométrica para otimizar a quantidade de material necessário através da construção de árvores de suporte e outras otimizações como a rotação do modelo. Áreas não-suportadas são identificadas pelo ângulo normal dos triângulos da malha do modelo. Comparando a normal do triângulo e a normal da superfície, é possível aplicar o conceito do ângulo de autossuporte. Após determinar as áreas que requerem suporte, pontos são posicionados uniformemente espaçados usando um algoritmo de rasterização. Uma vez tendo todos os pontos a serem sustentados, para cada um deles, um cone é computado para definir o volume onde pontos podem ser colocados para que sejam capazes de suportar o ponto original. Após essa preparação, os pontos são percorridos sequencialmente, ordenados por sua respectiva camada, de cima a baixo, em busca de outro ponto que forma a interseção válida mais próxima, usando uma busca gulosa. Assim, o suporte é progressivamente gerado. O algoritmo possibilita a interseção da estrutura de suporte com o próprio modelo, não obrigando que o suporte vá até a base da estrutura, possibilitando reduzir ainda mais a quantidade de material necessário. A solução utiliza a GPU tanto para a computação dos cones quanto para a intersecção destes com a malha do modelo, sendo esta última a parte mais custosa do processo. Os resultados obtidos foram comparados às soluções empregadas pelos softwares comerciais MakerwareTM e MeshmixerTM. Em todas instâncias apresentadas, o algoritmo descrito teve melhores resultados tanto em tempo quanto em material necessário.

Jiang, Xu e Stringer (2019) discutem o processo completo ideal para minimizar o volume de suporte necessário e apresentam uma nova abordagem de otimização que, em conjunto com o ângulo de autossuporte, é baseada em um parâmetro denominado *Longest Printable Bridge Length* (LPBL). O LPBL representa a máxima distância em uma direção que pode ser impressa sem precisar de suporte imediatamente abaixo, efetivamente construindo uma ponte. Assim sendo, o método se baseia em posicionar suportes separados por uma certa distância ao mesmo tempo que a direção da impressão é feita de maneira perpendicular aos suportes. Os autores exploram os parâmetros físicos que podem influenciar os resultados da impressão, especialmente os que afetam o LPBL, como temperatura do filamento, velocidade de impressão e velocidade do ventilador de resfriamento do material. Ao otimizar o planejamento de percurso de impressão, a fim de maximizar o LPBL, é possível melhorar ainda mais a economia de material. Quando comparado aos métodos tradicionais de construção de suporte, baseados em formação de pilares verticais até a base, a abordagem conferiu melhoria significativa nos modelos comparados, tanto para suportes internos quanto externos.

3 METODOLOGIA

Neste capítulo são discutidos os métodos utilizados na construção das estruturas de suporte. Na Seção 3.1 é apresentada a etapa de fatiamento de um modelo 3D e a notação utilizada nas seções subsequentes. Um método básico de construção de suporte, utilizando operações booleanas sobre polígonos, é apresentado na Seção 3.2. A Seção 3.3 apresenta a noção de autossuporte, que pode ser usada para remoção de algumas áreas de suporte desnecessárias. O conceito de autossuporte é também explorado para a formulação do algoritmo de suporte em árvore apresentado na Seção 3.4.

3.1 Fatiamento e notação

No processo de planejamento de impressão, a entrada comumente consiste em um modelo 3D definido por uma malha de triângulos. Esse modelo é fatiado por planos paralelos à base do modelo com diferentes alturas, ou seja, diferentes valores no eixo Z . O processo de fatiamento gera polígonos equivalentes a cada um dos perímetros presentes no modelo na altura dos planos. Uma fatia (camada) é definida pela coleção dos polígonos resultantes após o processo de fatiamento. A Figura 7 mostra um exemplo do processo: para cada plano dado, a intersecção com o modelo é computada para definir os polígonos de uma fatia. O algoritmo para fatiamento utilizado neste trabalho foi proposto por Minetto *et al.* (2017)¹.

O modelo de entrada para os métodos de geração de suporte, descritos neste trabalho, é representado por uma coleção de N fatias

$$E = E_1, E_2, \dots, E_N \quad (1)$$

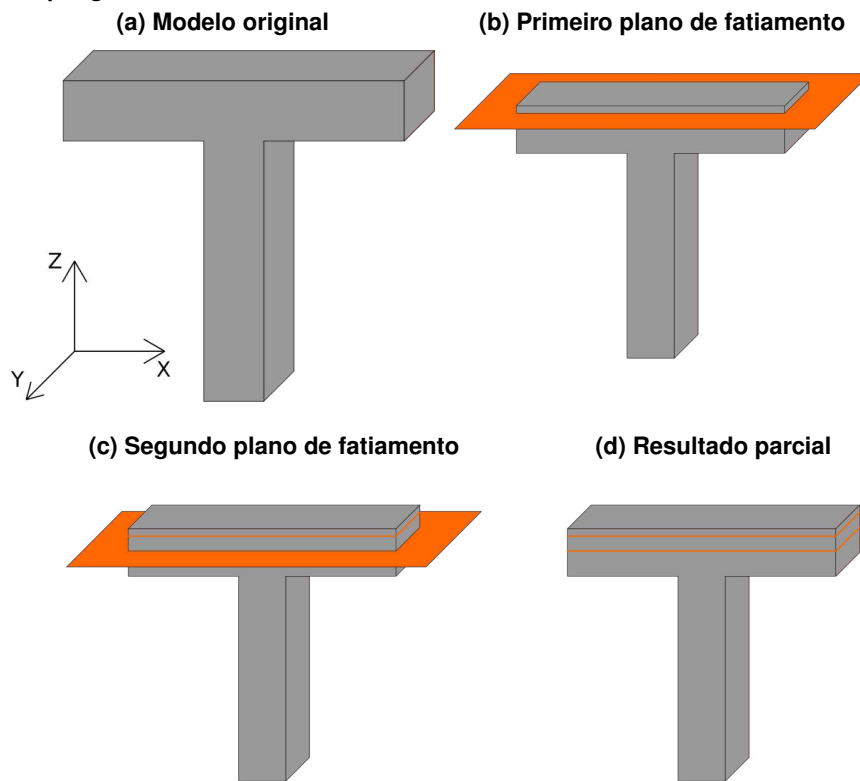
de forma que E_1 é a fatia do topo do modelo e E_N é a fatia da base (ver Figura 8). Cada fatia E_i é definida pela coordenada Z do plano que corresponde à altura da fatia e, também, por um conjunto de L_i polígonos:

$$E_i = P_i^1, P_i^2, \dots, P_i^{L_i}. \quad (2)$$

O modelo de entrada E não é alterado pelas estratégias de geração de suporte. Um modelo de saída $S = S_1, S_2, \dots, S_N$, contendo também N fatias, é criado para representar o suporte gerado.

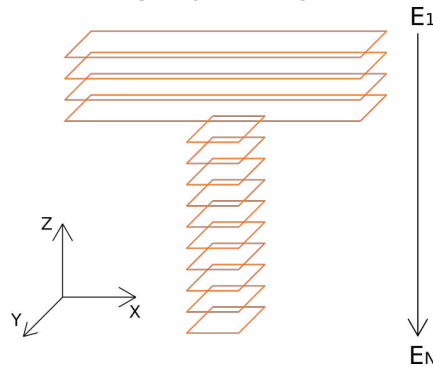
¹ Códigos para o algoritmo podem ser encontrados em <https://github.com/rminetto/slicing>, acessado em 5 de julho de 2022.

Figura 7 – Processo de fatiamento: um modelo é seccionado por um conjunto de planos para formar fatias de polígonos.



Fonte: Autoria própria.

Figura 8 – Resultado final do fatiamento: o conjunto de fatias do topo E_1 até a base E_N é usado como entrada para os algoritmos de computação de suporte.



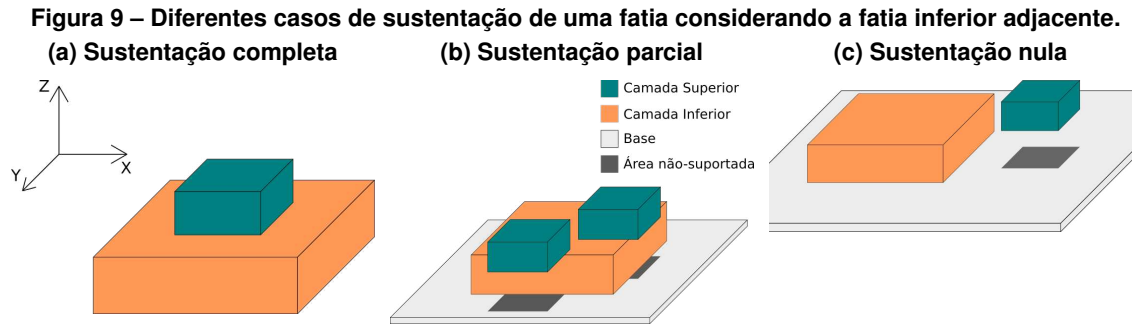
Fonte: Autoria própria.

3.2 Algoritmo básico para suporte

O algoritmo básico para geração de suporte se caracteriza por definir uma estrutura de suporte para qualquer região suspensa. Uma região suspensa, R_i , pode ser definida como toda área de uma camada E_i que não possui interseção com nenhuma das áreas definidas na camada imediatamente inferior E_{i+1} :

$$R_i \cap E_{i+1} = \emptyset. \quad (3)$$

A Figura 9 mostra exemplos de configurações entre duas fatias adjacentes, superior (verde) e inferior (alaranjado), de forma que a fatia superior pode ser totalmente suportada (Figura 9a), parcialmente suportada (Figura 9b) ou não suportada (Figura 9c). Para as regiões suspensas, indicadas pelas regiões sombreadas nas figuras, devem ser definidas estruturas de suporte. O suporte básico consiste na projeção ortogonal dessas regiões, em direção à base, até que seja encontrado um polígono capaz de sustentar a região ou até alcançar a própria base do modelo de entrada, a fatia E_N .



Fonte: Autoria própria.

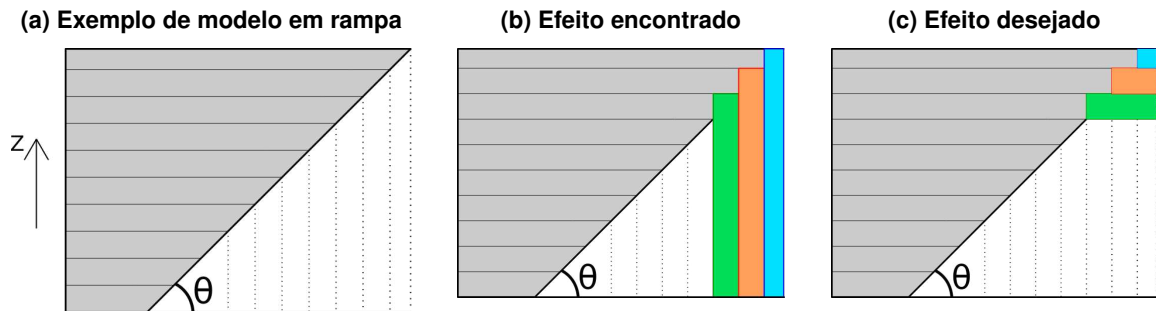
Para a computação de suportes, a partir da camada do topo, uma camada E_i é sempre comparada à camada imediatamente abaixo E_{i+1} , até que a base do modelo, E_N , seja alcançada. Como descreve a Equação 4, para todo polígono da camada de índice i , tanto de E_i quanto de S_i , é realizada uma operação booleana de diferença com a união dos polígonos da camada imediatamente inferior E_{i+1} . Se a diferença não for vazia, então existe região suspensa. Como a camada mais ao topo não tem o que sustentar, S_1 é vazio.

$$S_{i+1} = \bigcup_{P \in \{E_i \cup S_i\}} \left\{ P - \bigcup_{Q \in E_{i+1}} Q \right\} \quad (4)$$

Na prática, as coordenadas dos vértices dos polígonos de entrada podem ter certa imprecisão, por exemplo, em uma parede supostamente alinhada do modelo 3D, os polígonos de fatias adjacentes podem não estar perfeitamente alinhados. Por isso, é possível ocorrer casos com resultado diferente do que é esperado na teoria, como uma região contígua ser composta por múltiplos polígonos ao invés de apenas um. A Figura 10b representa um desses casos, em que uma única região de suporte acaba sendo subdividida em três.

Para corrigir essas condições, ao final da geração de cada camada S_i do modelo de saída, os polígonos que a compõem são minimamente dilatados por um fator F_{D1} e então passam por um processo de união. Em seguida os polígonos são erodidos por um fator F_E , onde $F_E > F_{D1}$ para que polígonos insignificantes sejam removidos. Finalmente, é realizada uma dilatação por um fator F_{D2} para restaurar o tamanho original dos polígonos. Os fatores citados devem seguir a seguinte relação:

Figura 10 – Possível separação das estruturas de suporte devido a imprecisões nas adjacências. Ao invés de ser obtida uma única região, são obtidas três regiões (representadas em verde, alaranjado e azul).



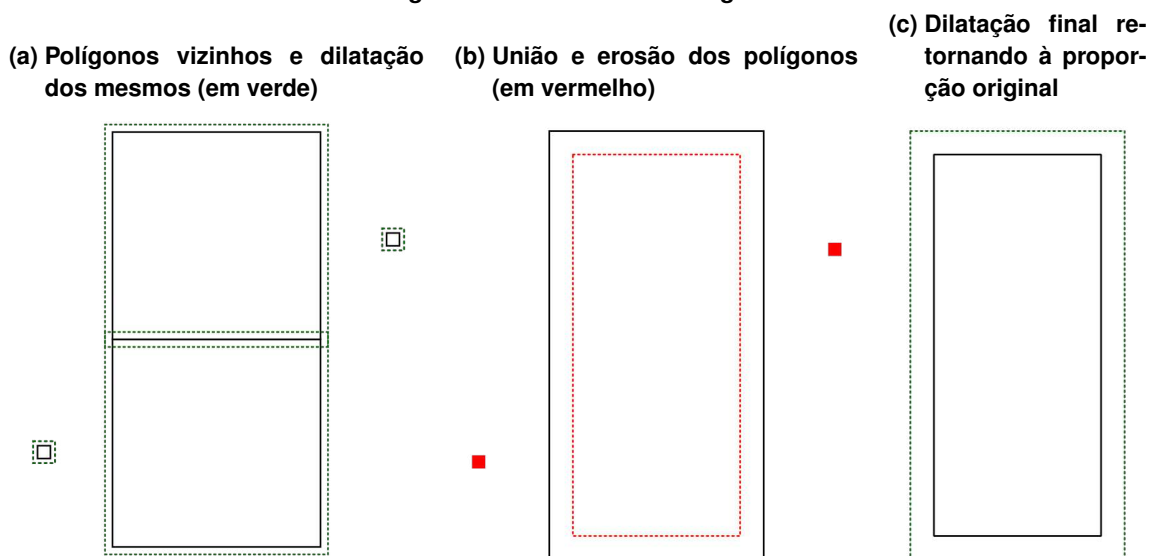
Fonte: Autoria própria.

$$F_{D1} - F_E + F_{D2} = 0. \quad (5)$$

Como a erosão e a dilatação são operações morfológicas inversas, ou seja, dilatar um polígono por um fator F_D e depois o erodir pelo mesmo fator não gera nenhuma mudança na área ou perímetro do mesmo, a sequência descrita acima não produz uma área de suporte diferente da inicial.

Ao primeiramente dilatar minimamente os polígonos, aqueles que antes estavam apenas se tocando passam a definir uma única forma após uma operação de união. A erosão que ocorre em sequência remove quaisquer polígonos com área insignificante para a análise de sustentação. A dilatação final retorna os polígonos à dimensão inicial. A Figura 11 exemplifica as etapas do processo.

Figura 11 – Processo de filtragem.

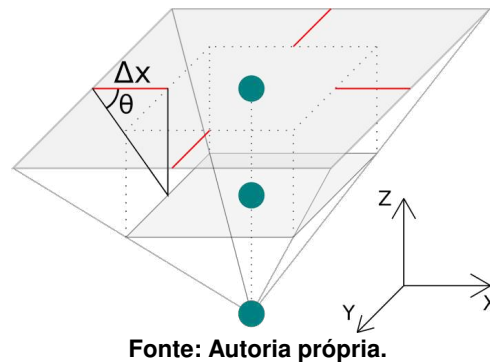


Fonte: Autoria própria.

3.3 Algoritmo considerando autossuporte

Na prática, não é necessário que uma estrutura seja totalmente perpendicular à superfície para que possa ser sustentada sem a adição de estruturas de suporte. O ângulo de autossuporte define a maior inclinação que o próprio modelo é capaz de sustentar. Considerando a Figura 12, se o ângulo de uma região da peça com uma linha horizontal for maior ou igual a θ , então uma camada superior é sustentada pela camada inferior, sem a necessidade de estrutura adicional de suporte. O ângulo de autossuporte varia de impressora para impressora e depende também de diversos parâmetros como velocidade de impressão, tipo de filamento, temperatura e espessura de cada camada do modelo (JIANG; STRINGER; XU, 2019). Na literatura, é comumente empregado o valor de 45° para esse ângulo.

Figura 12 – Ângulo de autossuporte: para determinadas inclinações não há necessidade de estrutura auxiliar de suporte.



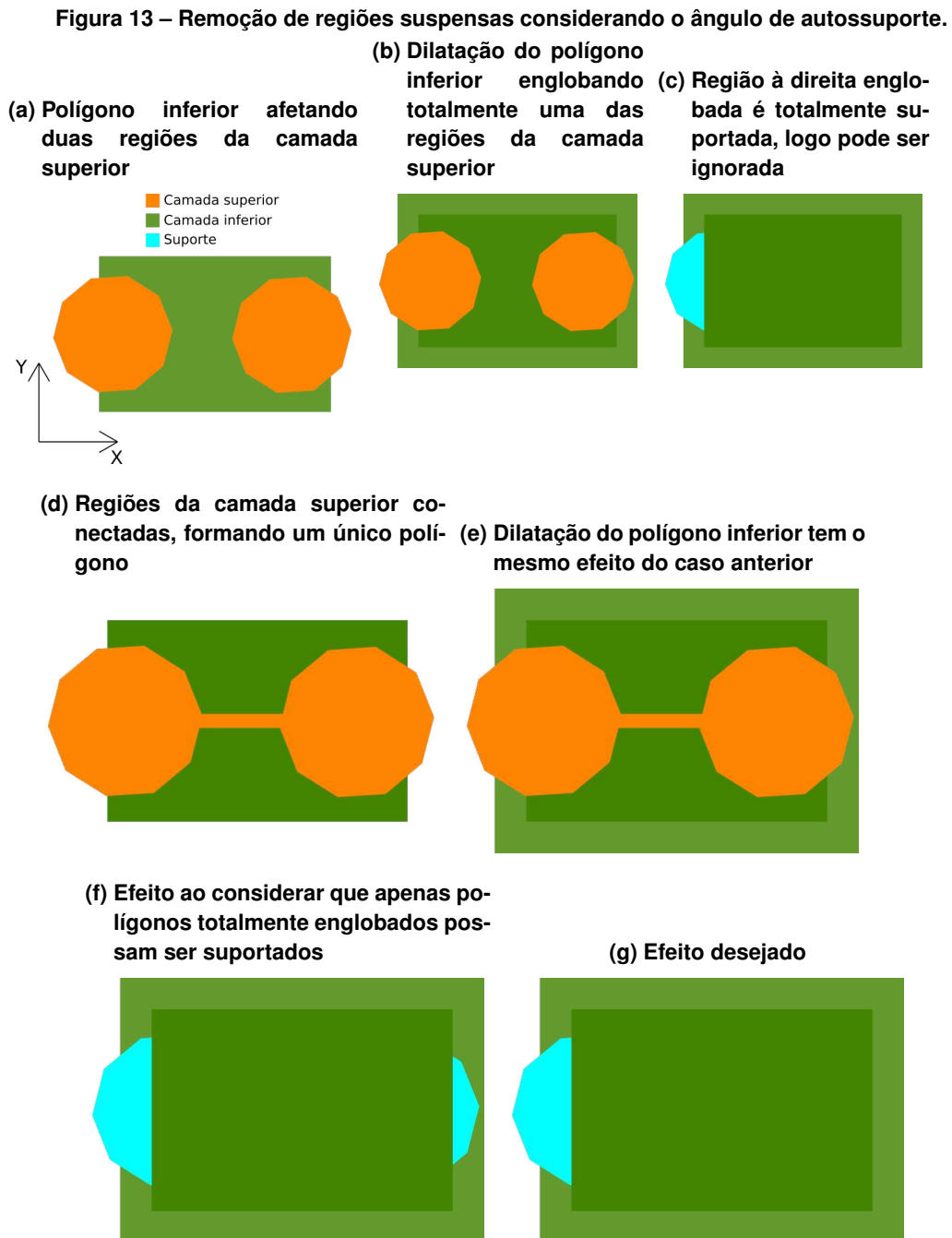
A estratégia de suporte descrita na Seção 3.2 pode ser vista como um caso especial quando θ é 90° . Para identificar a sustentação que uma camada tem em uma camada superior i , usando o ângulo de autossuporte, é usada uma dilatação da camada inferior $i + 1$ por um fator Δ_x . Este fator é calculado por uma simples regra trigonométrica:

$$\Delta_x = \frac{D_{i,i+1}}{\tan \theta} \quad (6)$$

tal que $D_{i,i+1}$ é a distância entre a camada i e a camada $i + 1$.

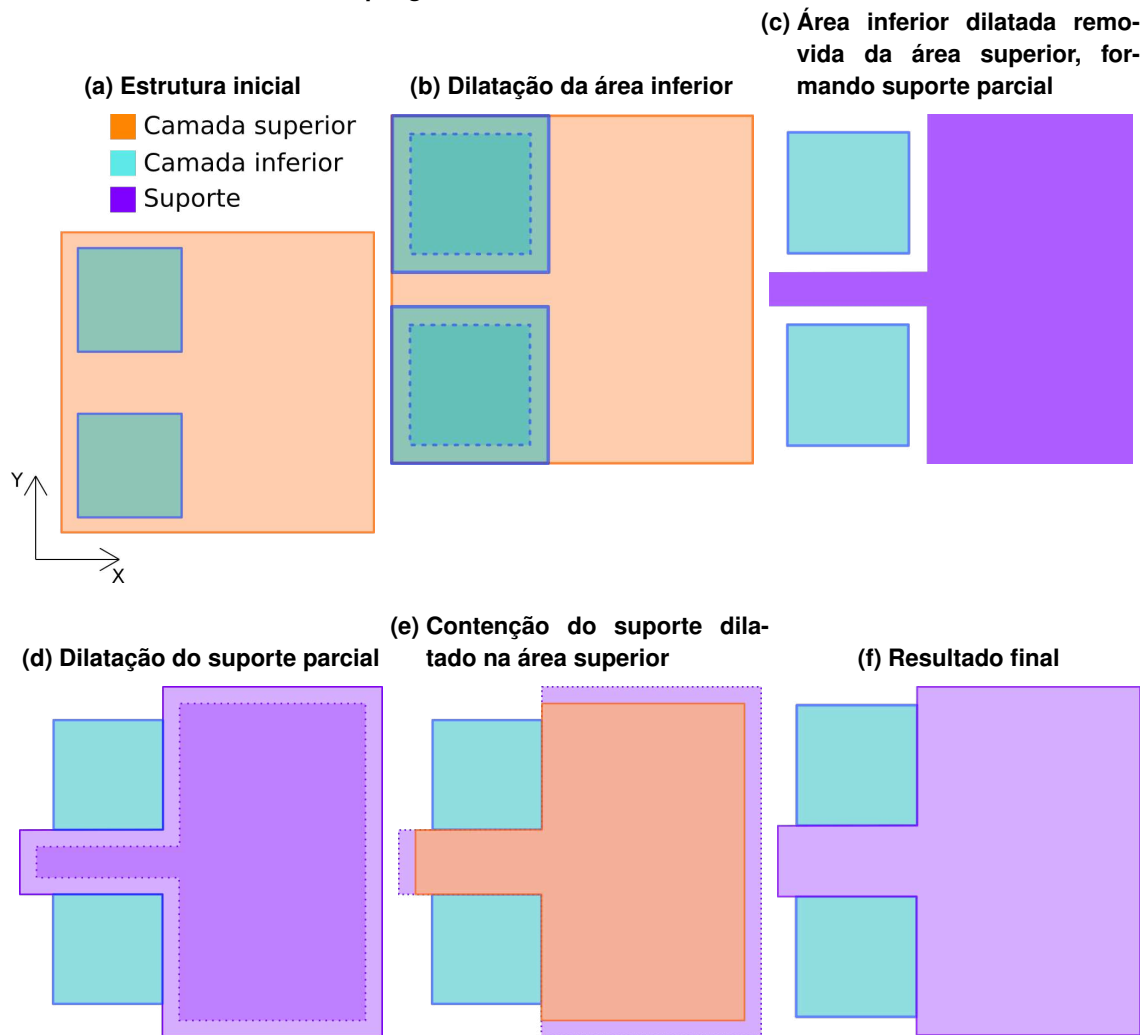
Quando um polígono da camada i é totalmente englobado pela camada dilatada abaixo, pode-se concluir que ele é perfeitamente sustentado pelo modelo original, não precisando de estruturas de suporte (ver Figuras 13a a 13c). Quando um polígono não fica totalmente englobado, ainda é possível que parte dele seja corretamente sustentada. As Figuras 13d a 13g mostram um exemplo assim. As regiões dessas figuras possuem uma conexão entre elas formando um único polígono (Figura 13d). Caso fosse aplicado a mesma regra, necessitando que um polígono esteja totalmente englobado para que seja sustentado, ambas as regiões acabam necessitando de suporte (Figura 13f). Porém, como foi observado (Figura 13c), a região à direita é autossuportável. Assim sendo, é de interesse manipular essa regra para que seja aplicada sobre “regiões totalmente englobadas”; não simplesmente “polígonos totalmente en-

globados”. Com isso, é possível chegar ao mesmo resultado de quando havia dois polígonos separados (Figura 13g).



A Figura 14 descreve passo-a-passo o processo utilizado para permitir que os resultados desejados sejam alcançados. Após a remoção da área que pode ser suportada da camada superior, como na Figura 14c, é possível observar que as bordas à esquerda da camada superior foram determinadas como autossuportáveis, não necessitando de suporte, podendo reduzir a área que de fato necessita de suporte.

Figura 14 – Remoção de regiões de suporte utilizando o ângulo de autossuporte. Caso com suporte próximos de uma borda de polígonos.



Fonte: Autoria própria.

Para chegar a esse resultado, as seguintes operações foram realizadas para definir as áreas que necessitam de suporte.

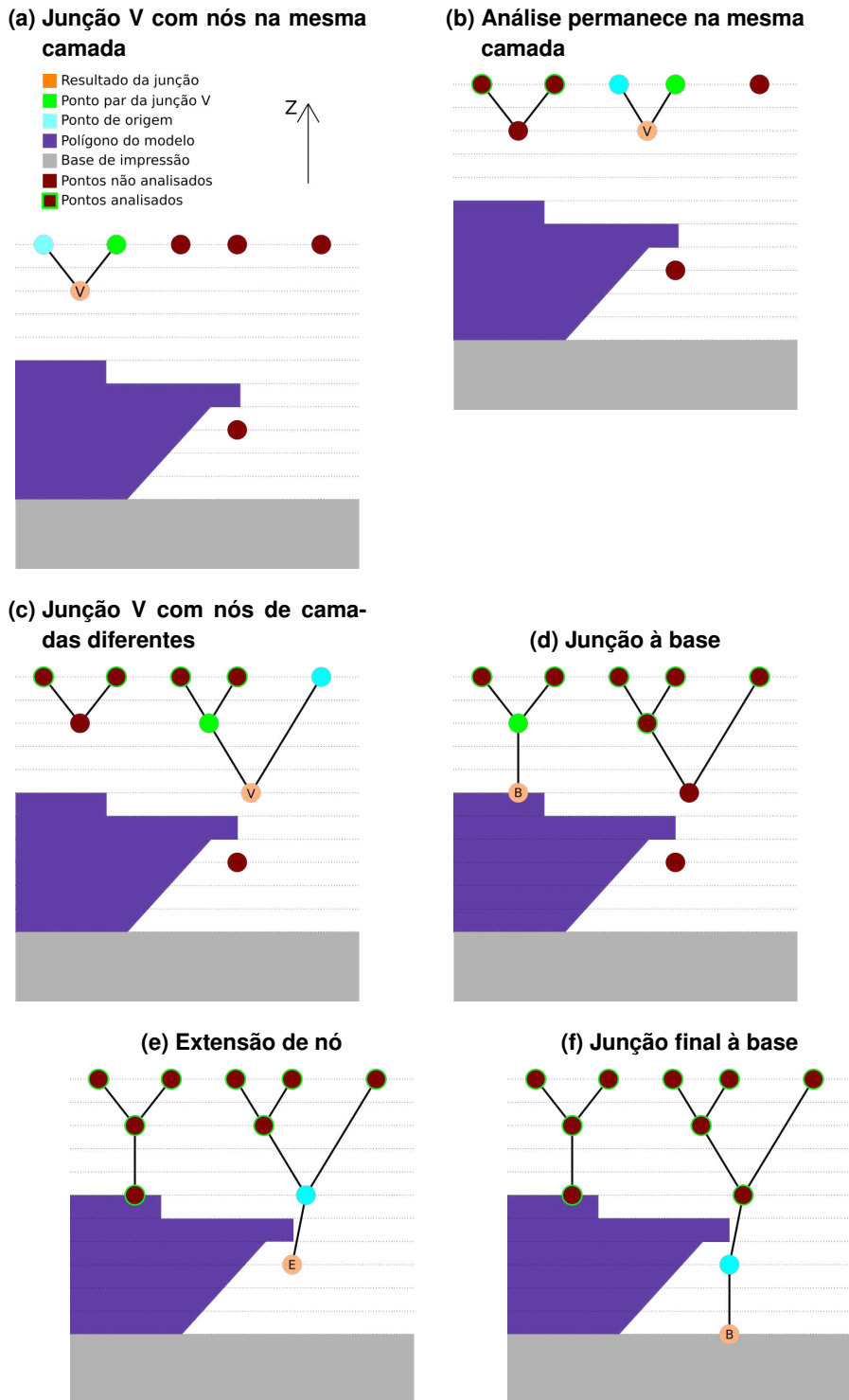
$$S_{i+1} = \text{offset}(\Delta x, (E_i - \text{offset}(\Delta x, E_{i+1}))) \cap E_i \quad (7)$$

3.4 Algoritmo baseado em árvore

A estrutura de suporte em forma de árvore pode ser usada como uma otimização para diminuir a quantidade de material necessário à impressão dos suportes. A partir de um conjunto inicial de pontos de sustentação, a estratégia conecta progressivamente os pontos em ramos de uma árvore (grafo). A conexão é baseada na ideia de regiões que são autossuportadas, como descrito na Seção 3.3. O uso de material é minimizado quando a árvore ocupa o menor volume possível, ou seja, quando a soma de todos os comprimentos das arestas da árvore

é a menor possível. No entanto, esse objetivo caracteriza o Problema de Árvore Mínima de Steiner (GILBERT; POLLAK, 1968), um problema que é ao menos da classe NP-difícil (VANEK; GALICIA; BENES, 2014).

Figura 15 – Exemplo de sequência de etapas no processo de construção de uma árvore.



Fonte: Autoria própria.

Considerando a dificuldade do problema, foi adotado um algoritmo guloso baseado em junções mais próximas: para cada ponto (nó) ainda desconectado é procurado um ponto de

junção com menor distância para formar uma subárvore. A Figura 15 exemplifica o processo de criação de árvores. Inicialmente, um conjunto de pontos é amostrado sobre regiões sem sustentação (Seção 3.4.1). Partindo das camadas superiores, para cada ponto é encontrado (ou criado) um ponto de junção mais próximo. O processo continua para as camadas inferiores, considerando tanto os pontos de sustentação, criados no início do processo, como pontos de junção, criados a medida que o processamento avança. A busca de uma junção para um nó é finalizada se atingir a camada de base (Figura 15f), se parte do próprio objeto é atingida pela árvore (Figura 15d) ou se a busca por um candidato melhor do que um já encontrado não é possível.

Nas próximas seções, as etapas relacionadas com a criação das árvores são explicadas em detalhes. O suporte em árvore utiliza as ideias das estratégias anteriores (Seções 3.2 e 3.3) para detectar regiões suspensas, que requerem suporte, e preencher essas regiões com pontos de sustentação: estes pontos formam as folhas da árvore. O conceito de autossuporte dirige a região de busca por junções de forma a criar ramos autossustentados. Nas Seções 3.4.1 e 3.4.2 são descritos os tipos de pontos (nós) usados na criação das árvores. Os pontos de sustentação (Seção 3.4.1) são os pontos iniciais, distribuídos em regiões que necessitam de suporte. Os nós de junção (Seção 3.4.2) são os nós produzidos durante o processo a partir de conexões entre outros pontos de sustentação ou de junção, podendo ser de três tipos: junção à base (Figuras 15d e 15f), junção em V (Figuras 15a-15c) e junção de extensão (Figura 15e). Nas Seções 3.4.3 e 3.4.4 são descritos, respectivamente, o processo de busca e validação de junções. Por fim, a Seção 3.4.5 descreve a conversão da árvore (grafo) para um conjunto de polígonos a serem impressos em cada camada.

3.4.1 Pontos de sustentação

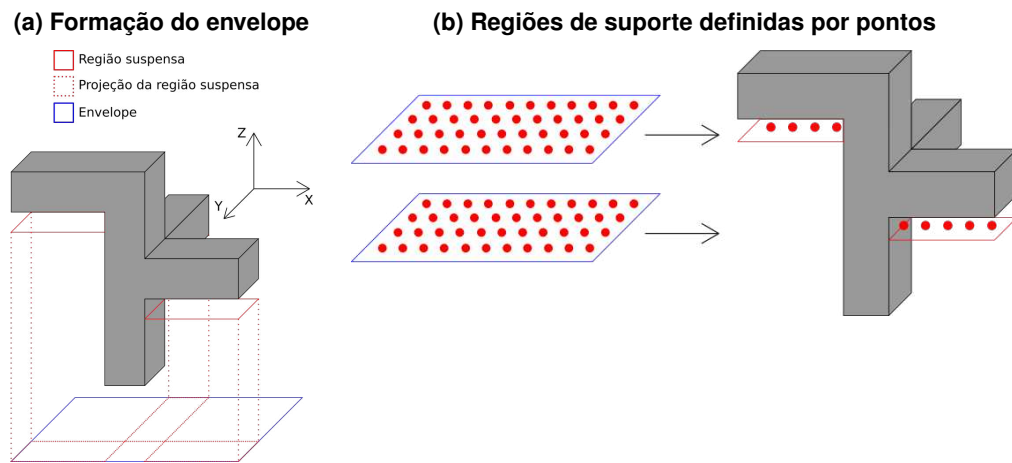
Conforme discutido na Seção 3.2, é possível identificar regiões suspensas pela diferença entre camadas adjacentes. No entanto, na construção da árvore, não é preciso considerar as extensões das regiões de suporte identificadas para fatias subsequentes. Modificando a Equação 4, as regiões suspensas seriam portanto dadas por:

$$S_{i+1} = \bigcup_{P \in E_i} \left\{ P - \bigcup_{Q \in E_{i+1}} Q \right\}. \quad (8)$$

Como essas regiões representam a camada de suporte mais próxima à camada a ser sustentada, elas irão conter as folhas das árvores. A Figura 16b mostra um exemplo de regiões suspensas e conjuntos de pontos distribuídos nestas regiões, formando o início das árvores que serão construídas.

Uma vez identificadas as regiões suspensas, é realizada uma projeção ortogonal até a base do modelo de todas essas regiões para então computar um envelope 2D - o menor

Figura 16 – Transformação das áreas de suporte em pontos iniciais para a construção das estruturas de árvores.



Fonte: Autoria própria.

retângulo que envolve as regiões (ver Figura 16a). Esse envelope é preenchido com pontos uniformemente espaçados no plano e, depois, é realizada a interseção desses pontos com as regiões suspensas, para atribuir a cada região os pontos que serão usados nas árvores como folhas (Figura 16b).

3.4.2 Tipos de nós de junção

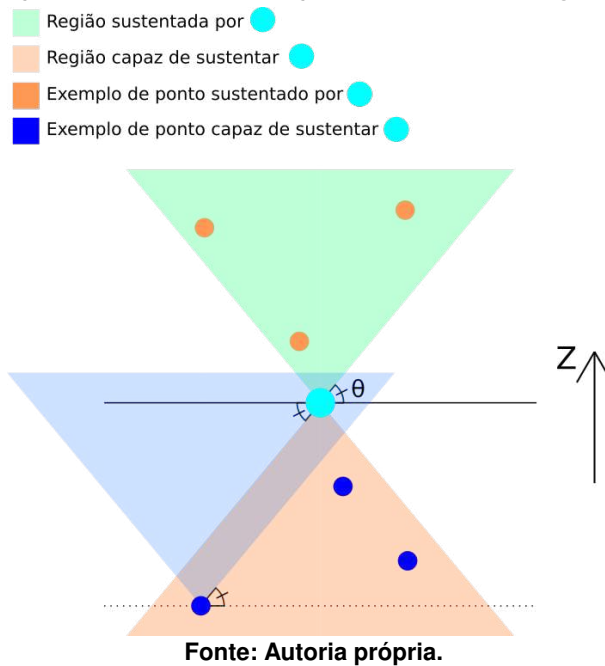
Há três diferentes maneiras para a formação de junções durante a construção da árvore. Considerando um ponto P , para o qual é procurada uma junção, e um ângulo θ de autossuporte, é possível definir uma região cônica, abaixo do ponto, de forma que pontos nesta região, em camadas inferiores, são capazes de suportar o ponto P . Portanto, a região cônica define onde pontos de junção devem estar situados para criar uma aresta de conexão com o ponto P . A Figura 17 ilustra essa região.

Para cada ponto P , o algoritmo de criação da árvore pode avaliar diversos pontos candidatos a fim de formar a melhor junção dentro da região cônica. Como citado anteriormente, a junção escolhida será aquela mais próxima. Todas as formas de junções podem ser consideradas para qualquer um dos pontos. Dentre as junções candidatas, existe a possibilidade de a junção não ser válida. Uma junção é válida se a aresta formada por ela e o ponto P não intersecta com qualquer parte do objeto e não é inferior (no eixo Z) à fatia de base (fatia E_n).

3.4.2.1 Junção à base

A junção à base corresponde ao “tronco” da árvore. Esse tipo de junção sempre é um dos candidatos e pode ser a melhor alternativa ou quando é a junção mais próxima ou quando não há nenhuma outra junção válida. O processo consiste em criar um nó considerando a projeção

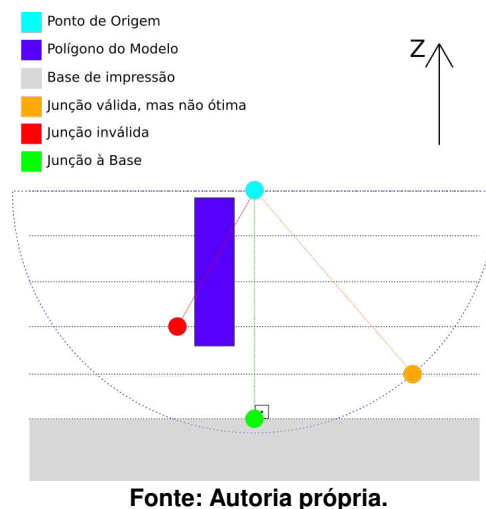
Figura 17 – Definição de cone de sustentação considerando o ângulo de autossuporte.



vertical do ponto P até que qualquer intersecção com polígonos do modelo 3D ocorra ou até que a camada da base seja atingida. Considerando que uma intersecção com o modelo ocorra na camada i , o ponto de junção possuirá as mesmas coordenadas que o ponto P no plano XY e valor no eixo Z correspondente ao da camada $i - 1$. Caso não haja intersecções com polígonos do modelo, a junção é criada na camada que determina a base do próprio modelo.

A junção do tipo base é considerada uma junção final, produzindo um nó raiz, ou seja, o nó gerado como base não participa de futuras análises de formação da árvore. A Figura 18 ilustra um exemplo de junção à base.

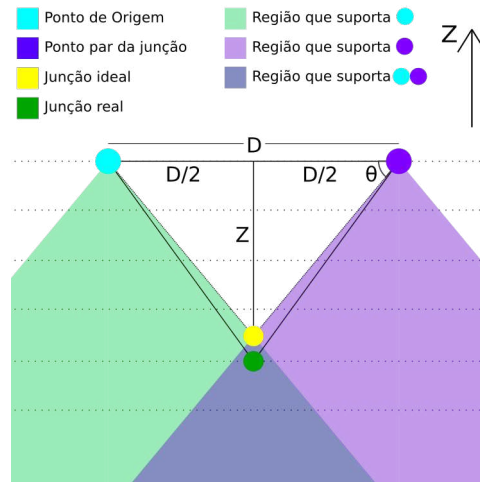
Figura 18 – Junção de um nó à base: o ponto de intersecção da projeção vertical de um ponto P , com a base do modelo ou com um polígono de uma fatia intermediária, corresponde a uma junção mais próxima.



3.4.2.2 Junção em V

A junção em V corresponde à conexão de dois pontos em um. O ponto de junção está situado na interseção entre os cones de sustentação dos dois pontos considerados, garantindo que a junção seja a mais próxima entre os pontos ao mesmo tempo que seja capaz de sustentar os pontos. Na Figura 19 é mostrado um exemplo de junção em V.

Figura 19 – Junção V entre nós em uma mesma camada.



Fonte: Autoria própria.

Quando os nós considerados pertencem a uma mesma camada, como na Figura 19, as coordenadas do ponto de junção são encontradas com base nas Equações 9 e 10. O ponto médio, P_M , determina as coordenadas em X e Y , enquanto a coordenada Z do nó de junção é determinada com base no ângulo de autossuporte θ . Estas relações são ilustradas na Figura 19.

$$P_M = \frac{P_1 + P_2}{2} \quad (9)$$

$$\Delta Z = \frac{|P_1 - P_2|}{2} \cdot \tan \theta \quad (10)$$

Quando os nós P_1 e P_2 , considerados para a junção em V, não estão na mesma camada, como na Figura 20, inicialmente é encontrado um ponto P'_1 . O ponto P'_1 é o ponto pertencente ao cone de P_1 mais próximo de P_2 na mesma camada que P_2 . Para encontrar P'_1 , usa-se a equação abaixo

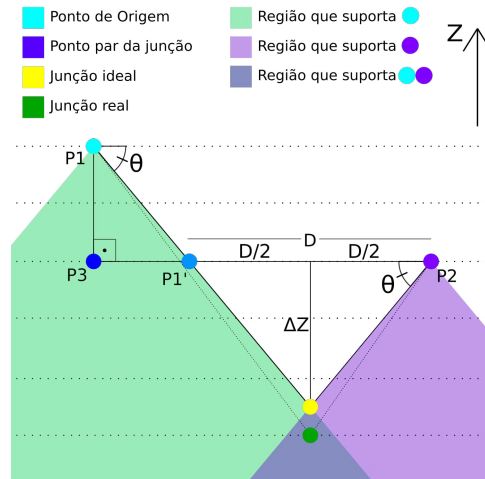
$$P'_1 = P_3 + \frac{P_2 - P_3}{\|P_2 - P_3\|} \cdot \frac{\Delta Z_{P_1, P_3}}{\tan \theta} \quad (11)$$

sendo que P_3 é projeção vertical de P_1 sobre a camada que contém P_2 e $\Delta Z_{P_1, P_3}$ é a altura do segmento dado por P_1 e P_3 . Tendo P'_1 , as coordenadas do ponto de junção podem ser encontradas pelas seguintes equações, de forma análoga ao caso quando os pontos estão na mesma camada, descrito anteriormente.

$$P_M = \frac{P'_1 + P_2}{2} \quad (12)$$

$$\Delta Z = \frac{|P'_1 - P_2|}{2} \cdot \tan \theta \quad (13)$$

Figura 20 – Junção V entre nós em diferentes camadas.



Fonte: Autoria própria.

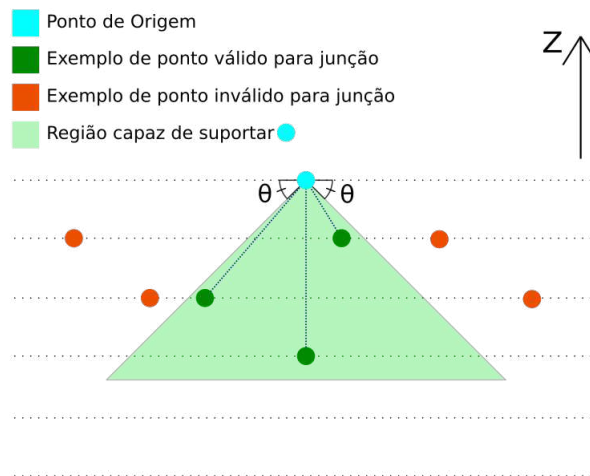
Uma vez que a análise para criação das junções é feita em um modelo discretizado de camadas, comumente a coordenada da junção ideal não terá a mesma coordenada Z que uma das camadas existentes. Devido a isso, a camada imediatamente inferior ao ponto ideal é utilizada como provedora da coordenada Z . Como resultado, o ângulo entre as arestas e a superfície é levemente maior que o de autossuporte. Mesmo assim, o ponto da junção continua estando dentro dos cones de suporte de ambos os pontos considerados. Tanto na Figura 19 quanto na Figura 20, os pontos ideais de junção (representados pelos nós de cor amarela) estão entre camadas, portanto, as junções são projetadas na camada inferior (nós em verde-escuro).

Esse tipo de junção não é final como no caso de junção à base. Nesse caso, o nó gerado é adicionado à camada em que foi computado e deve ser considerado para a criação de outras junções durante as próximas etapas de construção da árvore.

3.4.2.3 Junção de extensão

Uma junção de extensão é caracterizada por um nó já existente no interior do cone de um ponto P (Figura 21). Nesse tipo de junção, uma aresta na árvore é criada entre ela e o nó P . Esse é o único tipo de junção que não gera novos pontos. Mesmo assim, o nó achado continua fazendo parte das análises futuras de construção da árvore.

Figura 21 – Junção de extensão: nó interior ao cone de um ponto que produz a conexão mais próxima.

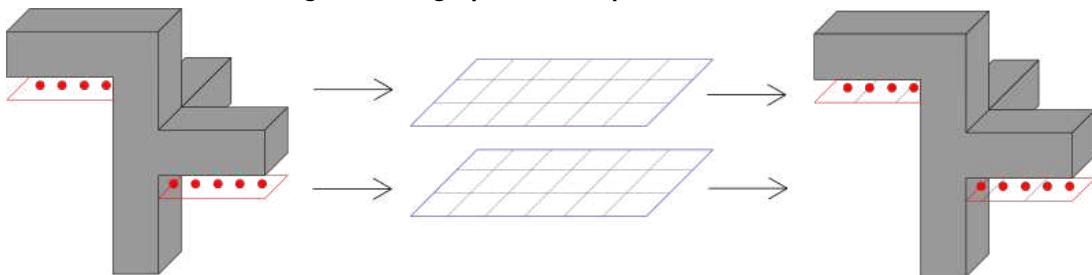


Fonte: Autoria própria.

3.4.3 Busca pela junção mais próxima

O objetivo do processo de busca é encontrar qual o nó que, junto com um nó P , irá produzir a junção mais próxima. Invariavelmente, a junção mais próxima de um ponto P envolve também os pontos mais próximos. A fim de evitar uma comparação exaustiva de distâncias entre todos os outros pontos e acelerar a busca pelo vizinho mais próximo, cada fatia é subdividida em células retangulares que passam a conter os pontos cujas coordenadas estão dentro da região descrita pela célula. Assim sendo, é possível realizar uma busca em largura pelo ponto mais próximo. A Figura 22 mostra um exemplo da criação da grade para os pontos iniciais de sustentação. Conforme a construção da árvore procede, nós de junção são também adicionados às células nas camadas em que estão localizados.

Figura 22 – Agrupamento de pontos em células.

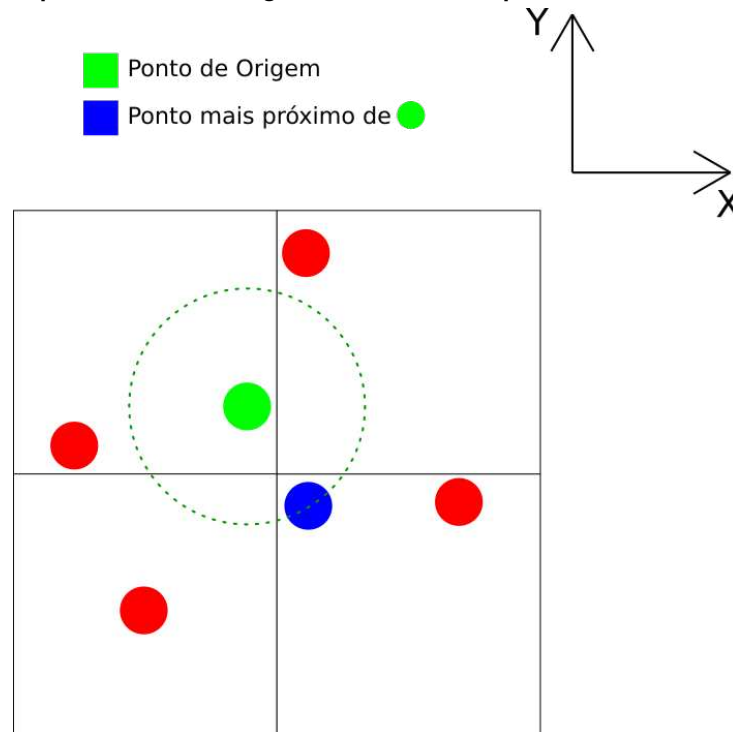


Fonte: Autoria própria.

A busca começa na mesma camada que o nó P sendo avaliado. Tendo a célula do nó P como ponto de partida, primeiro é verificado se há algum outro nó na mesma célula. Se houver, este é um forte candidato para o nó mais próximo. Porém, pode haver casos em que há um nó mais próximo em células vizinhas, como exemplificado na Figura 23. Por isso,

uma vez encontrado um candidato, independente do momento no processo de busca, é sempre necessário continuar, estendendo a fronteira de busca por mais um nível.

Figura 23 – Exemplo de busca em largura onde o nó mais próximo está em uma célula vizinha.

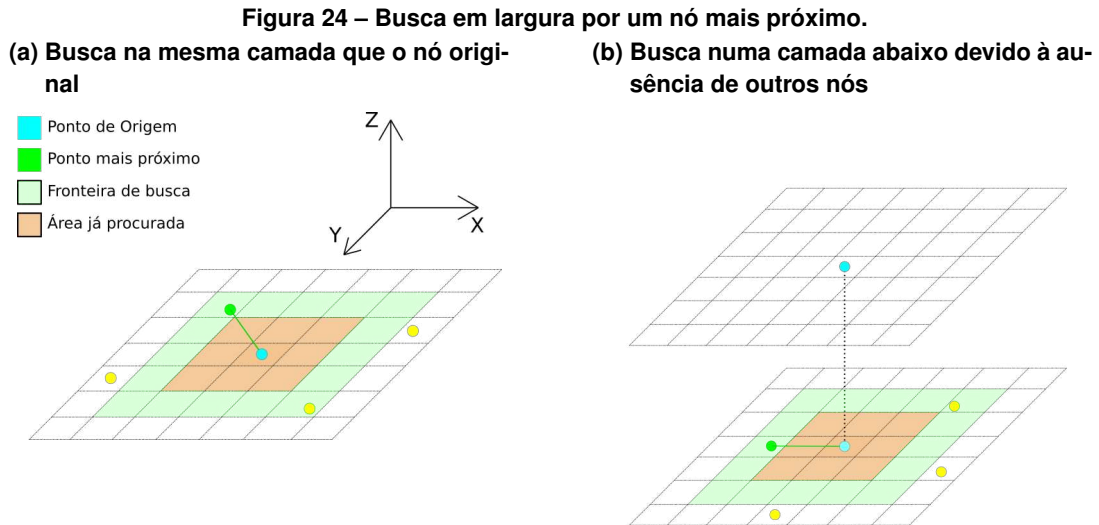


Fonte: Autoria própria.

De forma geral, a busca inicia na célula do ponto P e nas oito células ao seu redor. Caso não sejam encontrados pontos nestas células, a busca procede para as células adjacentes às últimas já verificadas, estendendo a fronteira de busca, e assim sucessivamente. A Figura 24a mostra um exemplo de como as células são avaliadas durante a busca. Caso não haja nenhum outro nó na mesma camada que P , a busca continua nas camadas inferiores (Figura 24b), até que seja encontrado um nó ou até que uma camada de base seja atingida.

Em alguns casos, é possível que sejam formadas áreas no espaço de busca com células que não contêm nenhum ponto, separando, por exemplo, regiões com alta densidade de pontos, como se fossem ilhas. Nesse tipo de situação, é possível que a busca em largura tenha que percorrer muitos níveis de busca, excedendo, possivelmente, o próprio número de pontos que podem ser comparados na camada. A fim de tentar remediar esse efeito, ao invés da busca em largura, foi adotada uma busca exaustiva avaliando todos os candidatos quando a quantidade de nós na camada é menor do que um dado limite τ . Assim, ao invés de avaliar um número de células que cresce quadraticamente com o nível de busca, passa-se a avaliar um número arbitrário de nós.

Caso a busca não encontre nenhum nó, atingindo uma camada de base, a junção produzida é uma junção de base (Seção 3.4.2.1). Caso tenha sido de fato encontrado um nó, primeiramente é verificado se o mesmo é interno ao volume que define o cone de sustentação, produzindo assim uma junção de extensão (Seção 3.4.2.3) e finalizando a busca. Do contrário,



Fonte: Autoria própria.

se o nó está fora do cone de sustentação, o nó define uma junção em V (Seção 3.4.2.2). Nesse último caso, é necessário continuar a busca, pois é possível que ainda haja outros candidatos a uma junção mais próxima.

Para que um ponto seja considerado como estando dentro do cone de sustentação do ponto P , é utilizada a Equação 11 para identificar o limite do cone na mesma camada que o nó encontrado. Caso a distância entre a projeção ortogonal do ponto P e o nó encontrado seja menor que a distância entre essa mesma projeção e o ponto do limite do cone, o nó estará dentro do volume do cone.

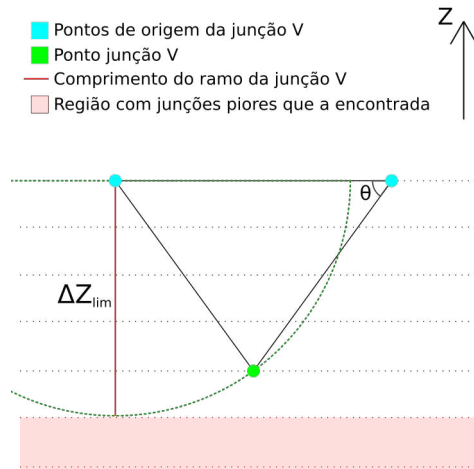
Ao encontrar uma junção em V , é possível definir um limite de busca para outras junções. Conforme ilustra a Figura 25, qualquer outra junção mais próxima deve estar no interior do círculo com raio igual à distância entre o nó P e a junção V encontrada. Assim sendo, pode-se considerar um limite ΔZ_{lim} (igual ao raio do círculo) de busca em camadas inferiores. As camadas ilustradas pela região em vermelho na Figura 25 só poderão produzir junções mais distantes.

Também é possível definir um limite de busca para as camadas que devem ser consideradas com base no ângulo θ . Para isso, utiliza-se a intersecção formada pela projeção vertical de P e a reta definida pela aresta da junção V com o par de P que gerou a própria junção (ver Figura 26). Qualquer ponto que esteja abaixo da intersecção não será capaz de produzir uma junção melhor, devido à limitação do ângulo de autossuporte.

Um terceiro limite para a busca pode ser obtido considerando a projeção de P até uma base (intersecção com um polígono do modelo ou a própria camada de base). Como é ilustrado na Figura 27, qualquer junção abaixo da base será uma junção mais distante.

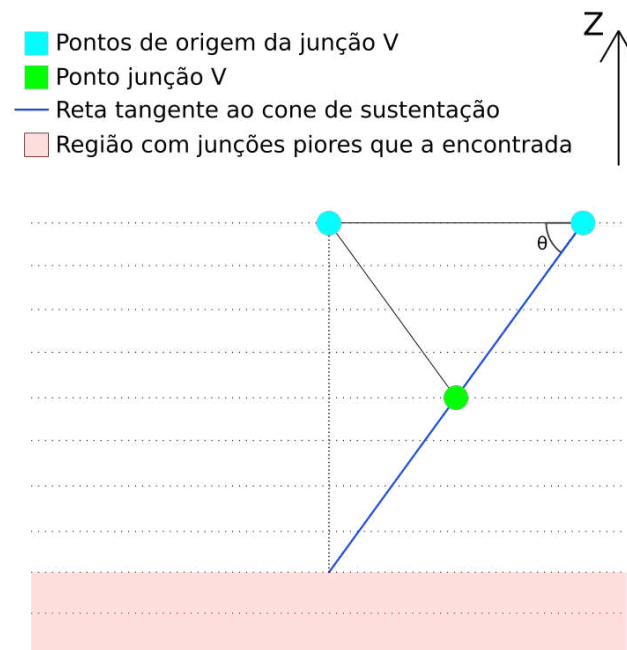
Como há três limites de busca para as camadas a serem consideradas, o limite a ser usado será o mais restritivo entre os três.

Figura 25 – Uma junção V encontrada limita a busca por outras junções. As camadas consideradas devem estar a uma distância vertical menor ou igual ao raio do círculo definido pela junção V.



Fonte: Autoria própria.

Figura 26 – Pelo ângulo de autossuporte é possível determinar um limite para camadas de busca.



Fonte: Autoria própria.

Além do limite de busca definido para camadas a serem consideradas, há também um limite para a busca em largura em cada camada. Este limite também é dado pelo ângulo de autossuporte. A reta formada pela junção V e o ponto par de P , o outro ponto responsável pela junção V, define também um cone que limita onde outras melhores junções V podem estar (Figura 28). Qualquer junção fora deste cone pode ser imediatamente descartada, pois é incapaz de produzir uma junção V melhor que a já encontrada.

A composição de todos estes limites determina a região onde deve ser realizada a busca por outras junções melhores do que uma junção V já encontrada (Figura 29).

Figura 27 – Se a projeção de um ponto atinge um objeto que se comporta como base, o limite de busca para camadas é determinado pela camada da mesma base.

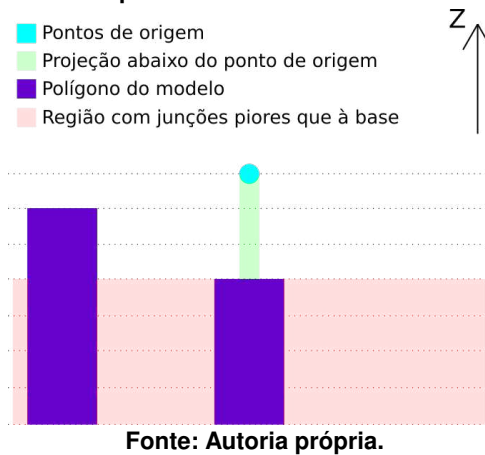
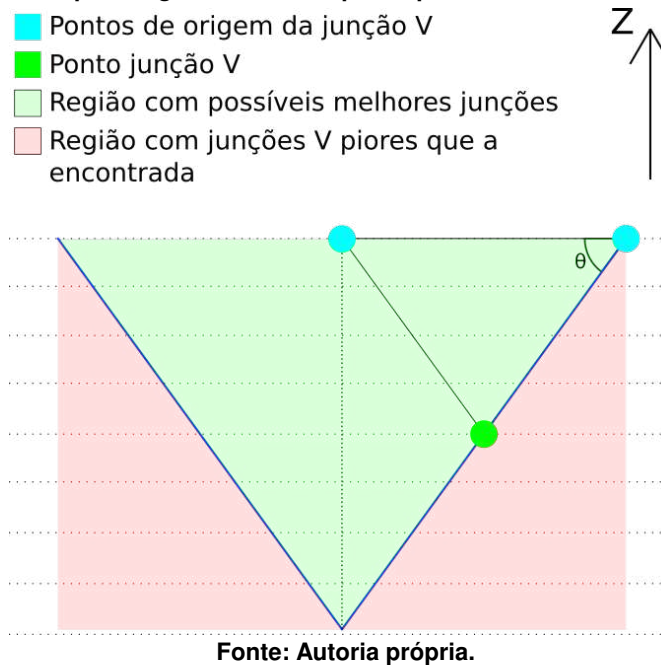


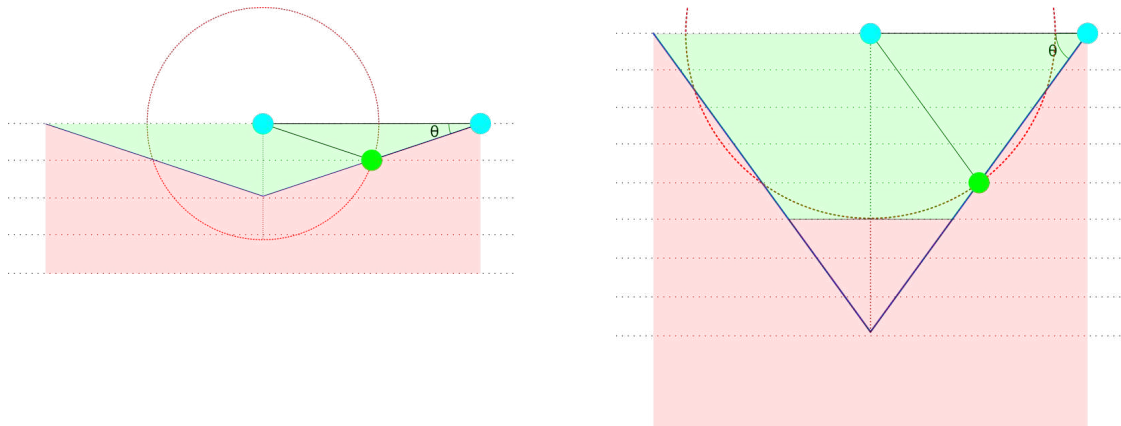
Figura 28 – Cone definido pelo ângulo de autossuporte que limita a fronteira de busca nas camadas.



Conforme novos candidatos de junção são encontrados, os limites de busca podem ser atualizados. Após uma junção V inicial (Figura 30a), novas junções V mais próximas podem ser encontradas (Figura 30b), limitando mais ainda o espaço de busca. A Figura 30c mostra o caso em que uma junção de extensão é encontrada. Neste caso a busca pode ser finalizada, pois a junção de extensão minimiza a distância até o ponto P .

Durante o processo de busca, é desconhecido se as junções encontradas são realmente válidas. Por isso, até que o limite de busca seja alcançado, as junções candidatas, cada vez melhores, são mantidas em uma pilha. Dessa maneira, caso uma possível junção seja invalidada, a próxima melhor é avaliada, até que seja encontrada uma junção válida. Caso as opções se esgotem, uma junção à base é criada.

Figura 29 – Composição de todos os limites parciais no limite final de busca.
 (a) Área de busca quando ângulo de autossuporte menor que 45°
 (b) Área de busca quando ângulo de autossuporte maior que 45°

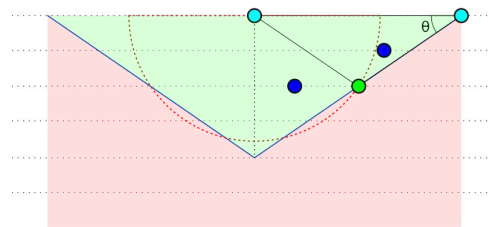


Fonte: Autoria própria.

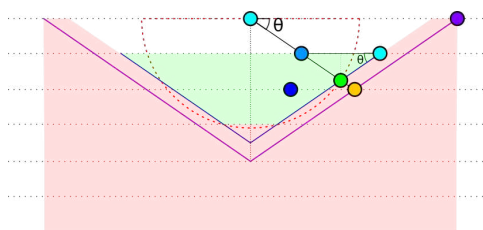
Figura 30 – Atualização dos limites de busca.

(a) Junção V inicial

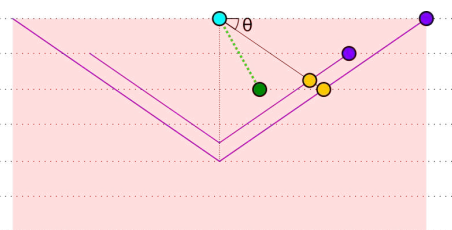
- Pontos de origem da junção V
- Junção V atual
- Junção V anterior
- Pontos não avaliados
- Ponto associado à junção V anterior
- Junção final (extensão)
- Região com possíveis junções melhores
- Região com junções V piores que a encontrada



(b) Uma junção V melhor encontrada



(c) Fim da busca ao encontrar uma junção de extensão



Fonte: Autoria própria.

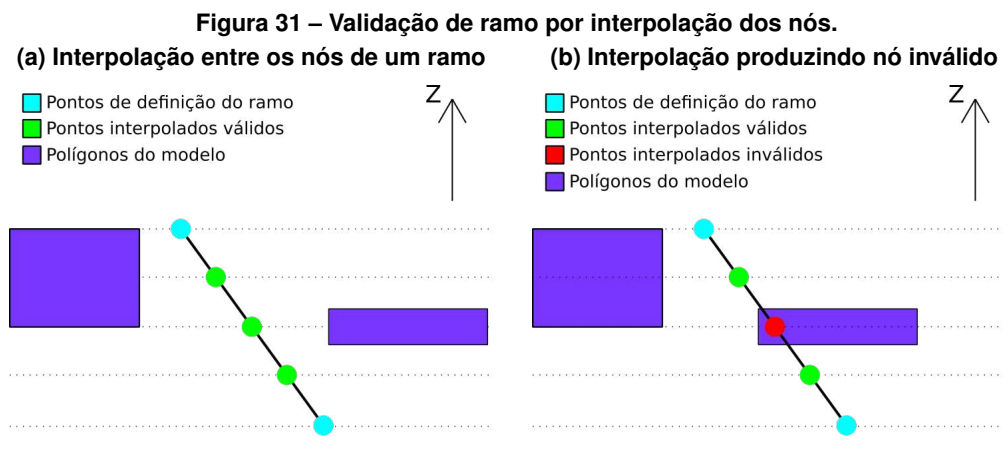
3.4.4 Validação de nós de junção

O processo de validação de um nó de junção consiste em avaliar se os segmentos de linhas (ramos) entre o nós participantes da junção produzem uma interseção com alguma parte do modelo de entrada ou com a base do mesmo. Como o modelo já está discretizado em camadas, os pontos do segmento que precisam ser avaliados podem ser determinados usando

a coordenada no eixo Z das camadas que cortam o segmento. Considerando um segmento dado pelo ponto $P = (x_0, y_0, z_0)$ e pela junção $P_J = (x_J, y_J, z_J)$, as coordenadas (x', y', z') do ponto do segmento, em uma camada definida na altura z , são obtidas fazendo $z' = z$ e calculando as interpolações lineares dadas pela equações abaixo com $t = \frac{z-z_0}{z_J-z_0}$. A Figura 31 mostra exemplos de interpolação produzindo uma junção válida (Figura 31a) e uma junção inválida (Figura 31b). A identificação de uma intersecção pode ser realizada computando a posição de um ponto relativa a um polígono (Seção 2.1), se o ponto é interno a algum polígono, então ele é inválido.

$$x' = (1 - t)x_0 + x_J t \quad (14)$$

$$y' = (1 - t)y_0 + y_J t \quad (15)$$



Fonte: Autoria própria.

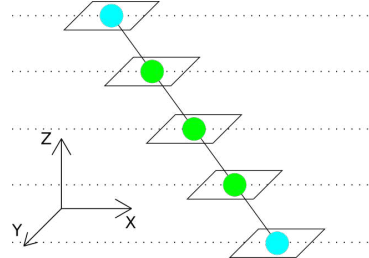
Além de verificar se há intersecção com parte do modelo, é preciso ainda verificar se a junção não está localizada abaixo da camada de base do modelo. Caso isso ocorra, uma junção de base é preferível, pois produz um ramo mais curto. Assim sendo, a junção candidata é invalidada.

3.4.5 Construção do suporte a partir do grafo

Ao fim da construção do grafo, os nós de cada um dos ramos da árvore computada não necessariamente estão todos em camadas adjacentes. É possível que os nós que definem um ramo estejam separados por múltiplas camadas. Uma interpolação linear é então usada para subdividir os ramos criando pontos em cada uma das camadas intermediárias. No lugar de cada um dos pontos gerados são então criados polígonos para de fato definir a região de impressão da árvore. A Figura 32 mostra a interpolação de nós (verdes) entre dois nós que formavam uma

aresta (azuis). Os polígonos na figura representam a formação de uma área para deposição de material durante a impressão.

Figura 32 – Subdivisão de arestas e criação de polígonos em nós para formar regiões a serem impressas.



Fonte: Autoria própria.

Os polígonos podem ser parametrizados usando as informações da própria estrutura da árvore. Por exemplo, conforme a profundidade de um nó, é possível criar polígonos maiores para formar a base das árvores e polígonos menores para ramos mais próximos às folhas. Para fins de demonstração, neste trabalho foi determinado que a área dos polígonos é diretamente proporcional à profundidade do nó amostrado. Como as raízes das árvores estão nas bases e as folhas no topo, a estrutura gerada se assemelha a uma árvore na natureza, com um tronco largo e galhos finos. Outras estratégias são também facilmente adaptáveis, por exemplo, Vanek, Galicia e Benes (2014) utilizam uma heurística baseada no ângulo e comprimento do ramo para determinar o diâmetro do polígono.

4 RESULTADOS

Neste capítulo são apresentados experimentos para demonstrar os suportes resultantes do algoritmo desenvolvido. Na Seção 4.1 são mostrados alguns exemplos de árvores de suporte resultantes para diferentes modelos. Na Seção 4.2 é realizada uma comparação dos tempos dos algoritmos e, também, é apresentada uma análise de características das árvores e das buscas do algoritmo para criação de árvores. Por fim, uma comparação dos volumes de suporte é apresentada na Seção 4.3.

Os algoritmos foram implementados usando Python, versão 3.8.10, como linguagem de programação. As operações booleanas sobre polígonos e a manipulação de objetos planares fizeram uso da biblioteca Shapely (SHAPELY, 2021). Os experimentos foram realizados em um computador com processador Ryzen 3600, de 8+8 GB 3600MHz RAM e sistema operacional Ubuntu 20.04.3 LTS. Os modelos utilizados nos experimentos são mostrados na Figura 33 e foram fatiados usando a distância de 0.254 milímetros entre camadas. A distância entre os pontos de sustentação (folhas) usados no suporte em árvore é de 1 milímetro e o limite τ para uma busca exaustiva (Seção 3.4.3) considerado nos experimentos foi 50. Exceto pelo experimento apresentando o efeito da configuração do ângulo mínimo para os ramos da árvore (apresentado na próxima seção), as análises consideram um ângulo mínimo em relação à base para os ramos de 45 graus.

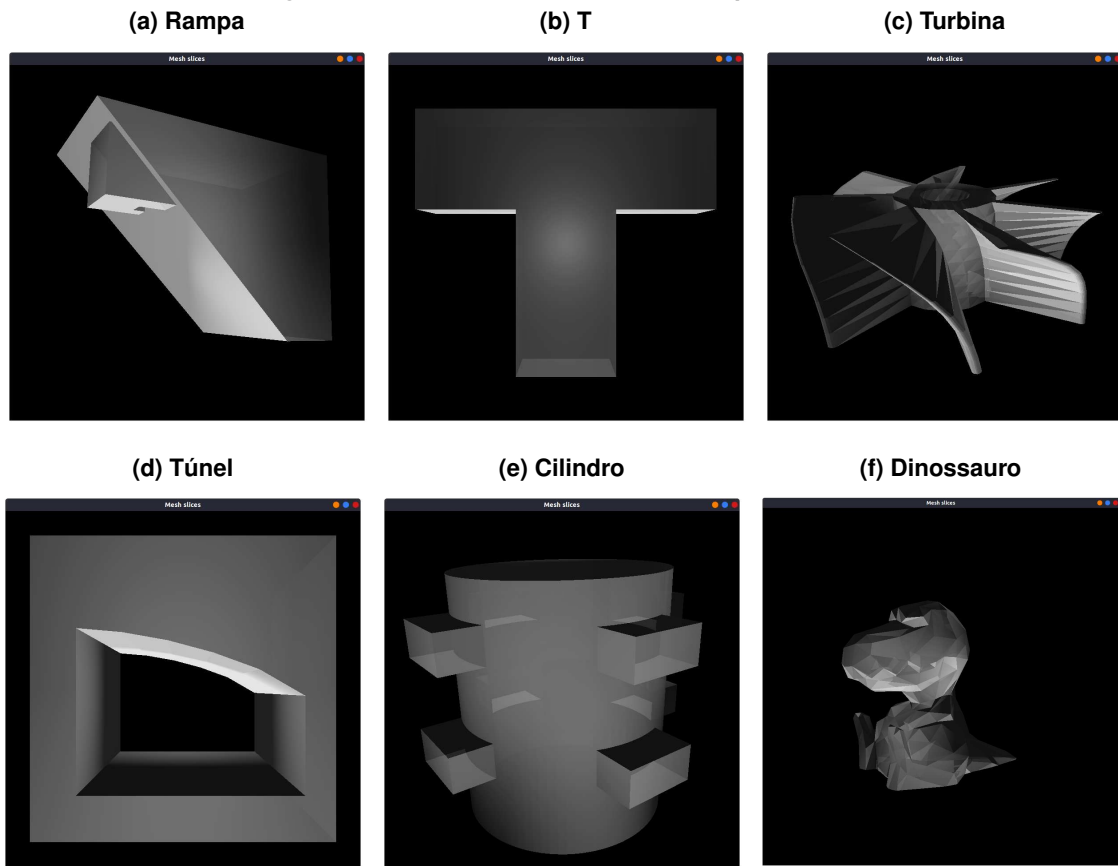
4.1 Exemplos de árvores

A Figura 34 mostra exemplos de suportes para o modelo “Rampa”, que é composto por uma região inclinada de 45 graus com uma pequena região suspensa no topo da inclinação. Na Figura 34a, usando a estratégia de suporte básico, toda área suspensa requer suporte; na Figura 34b, considerando o ângulo de autossuporte, toda região inclinada é dada como não necessitando suporte, sobrando apenas um pilar de suporte. Finalmente, na Figura 34c, o volume do pilar é otimizado utilizando uma estrutura de árvore como suporte.

O ângulo de autossuporte influencia tanto na detecção de áreas necessitando suporte quanto na formação da estrutura da árvore. Mesmo assim, o ângulo usado na detecção de áreas de fato suspensas e o ângulo preferível para os ramos das árvores podem ser escolhidos separadamente. A Figura 35 mostra os resultados da geração de árvores para o modelo “T” considerando três ângulos distintos: 30°, 45° e 60°. Nota-se que quanto menor o ângulo, mais rapidamente o tronco da árvore é formado, resultando em maior economia de material. Nas análises subsequentes, o ângulo mínimo para os ramos escolhido é sempre 45°.

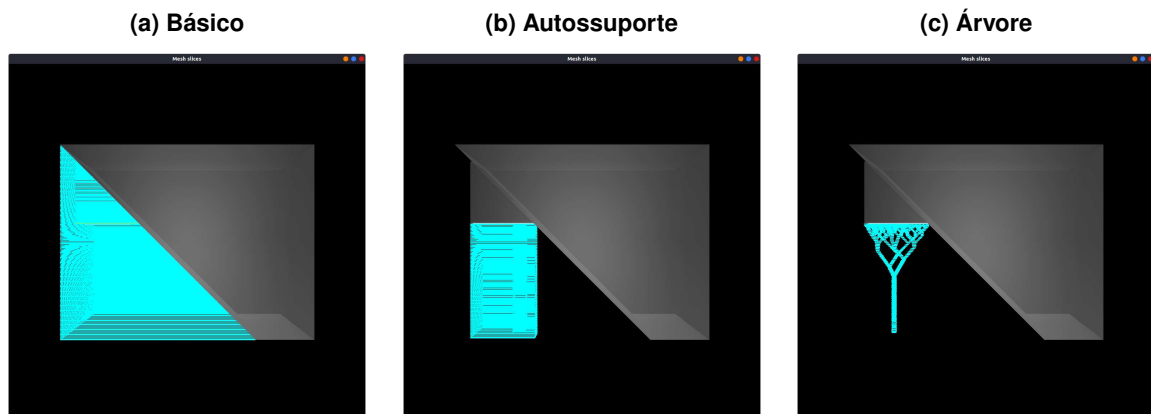
O modelo “Turbina” é composto por hélices que se curvam com maior intensidade conforme a altura da camada em relação à base do modelo aumenta. A Figura 36 mostra a formação das árvores. Nestas árvores, as folhas não estão todas na mesma camada, como nos modelos anteriores.

Figura 33 – Modelos 3D utilizados nos experimentos.



Fonte: Autoria própria.

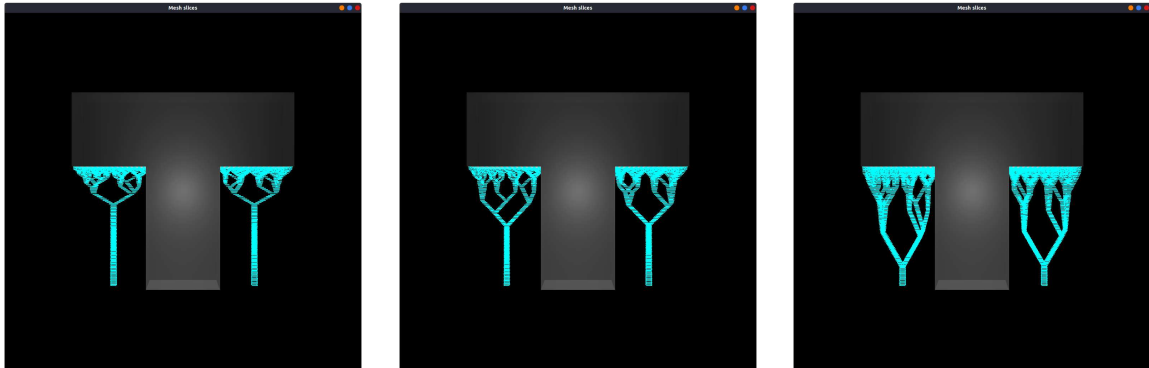
Figura 34 – Comparação de estratégias de suporte: (a) suporte em toda região suspensa; (b) suporte apenas em inclinações que não atendem o ângulo de autossuporte; e (c) suporte usando estrutura de árvore.



Fonte: Autoria própria.

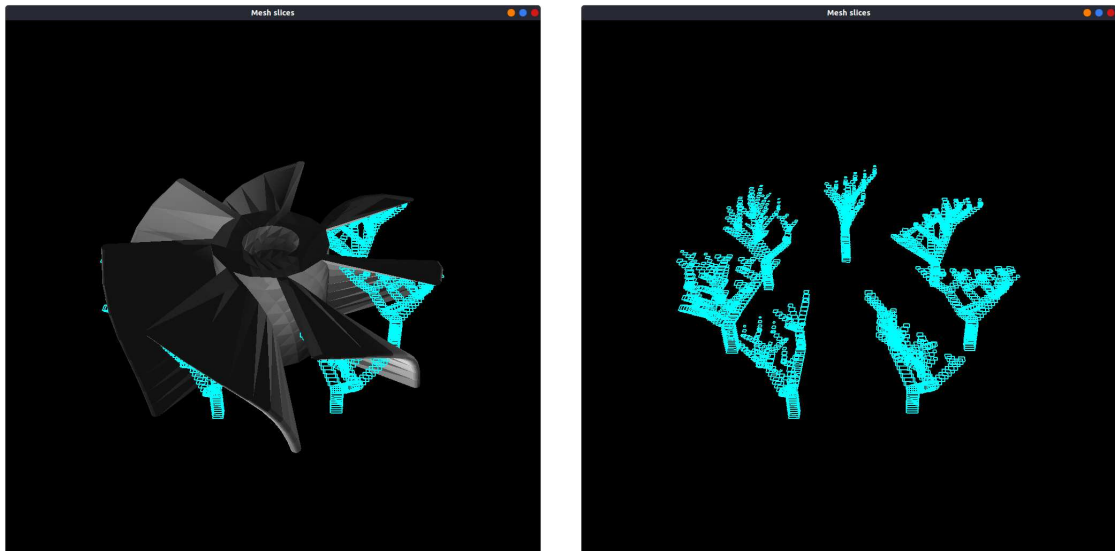
O modelo "Túnel" é um paralelepípedo com uma região vazia que atravessa o volume do modelo. O teto deste túnel se curva perpendicularmente à direção do túnel em apenas um dos lados. A Figura 37 mostra o resultado do suporte para este modelo. O modelo "Túnel" possui um volume consideravelmente maior que os demais. Uma consequência disso é que cerca de oito árvores precisaram ser formadas para o suporte do modelo, mesmo sendo uma área contígua,

Figura 35 – Efeito do ângulo mínimo para os ramos na árvore resultante.
 (a) 30° (b) 45° (c) 60°



Fonte: Autoria própria.

Figura 36 – Geração de suporte em árvore do modelo Turbina.
 (a) Malha e suporte em árvore (b) Malha omitida e suporte em árvore



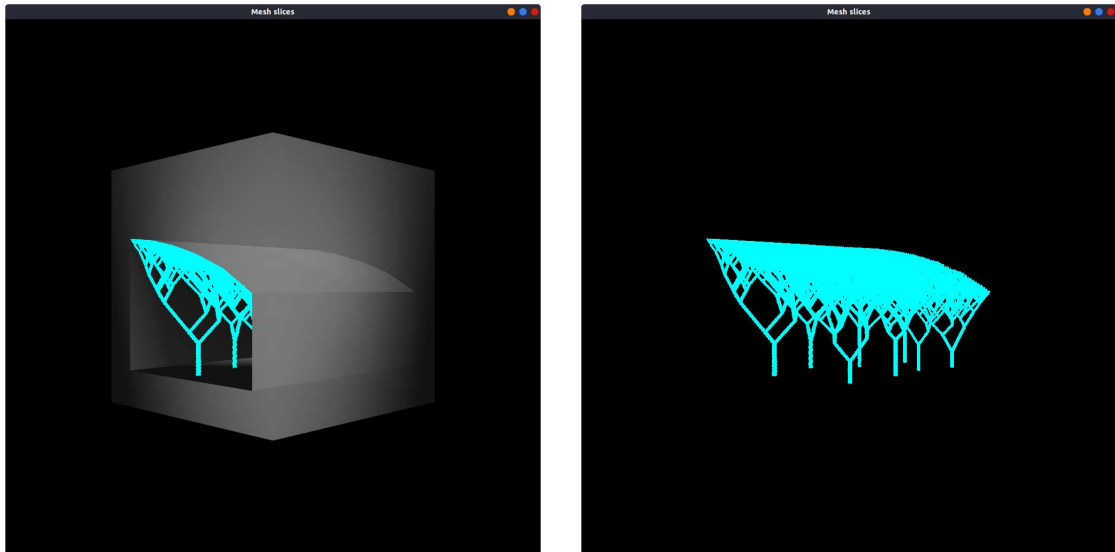
Fonte: Autoria própria.

dado o ângulo de autossuporte. Pode-se notar também que as bases dos tronco das árvores são suportadas pelo próprio modelo neste caso.

O modelo “Cilindro Dentado” é basicamente um cilindro com paralelepípedos protuberantes em suas laterais, formando “dentes”. O modelo possui dois níveis/alturas de dentes e cada nível possui quatro dentes, sendo que num mesmo nível eles estão igualmente espaçados e alinhados com os dentes do outro nível. A Figura 38 mostra o resultado de geração de árvore para este modelo. Pode-se observar as árvores terminando em diferentes camadas do modelo: um grupo sendo suportado pelos dentes do nível inferior e outro grupo sendo suportado pela base do modelo em si.

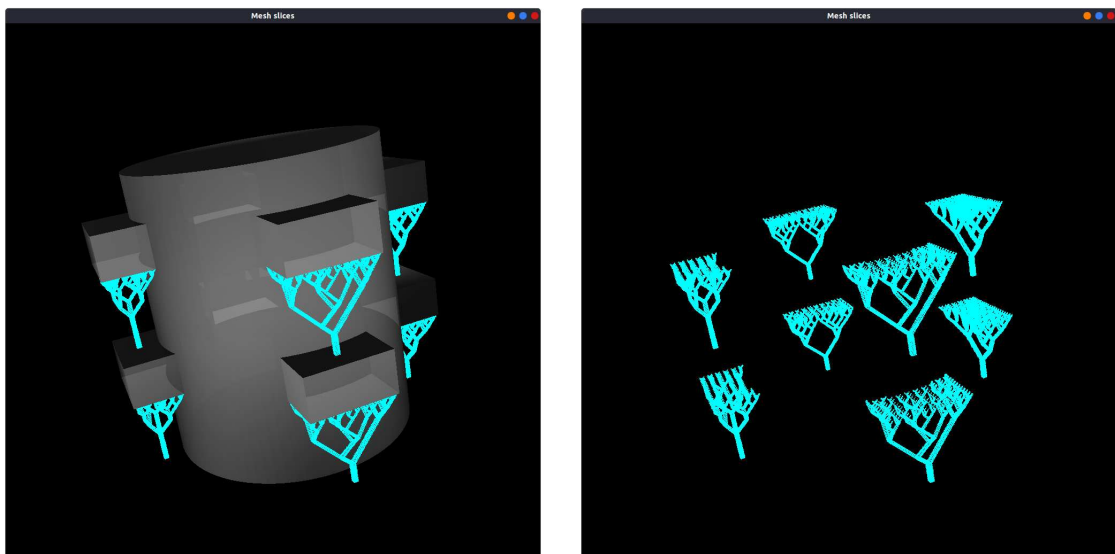
A Figura 39 mostra o resultado da geração de suporte em árvore para o modelo “Dinossauro”. O suporte correspondente ao focinho do dinossauro é o mais considerável, porém, é possível notar também suporte em parte da região dos olhos e na região da perna, próximo

Figura 37 – Geração de suporte em árvore do modelo Túnel.
(a) Malha e suporte em árvore **(b) Malha omitida e suporte em árvore**



Fonte: Autoria própria.

Figura 38 – Geração de suporte em árvore do modelo Cilindro Dentado.
(a) Malha e suporte em árvore **(b) Malha omitida e suporte em árvore**



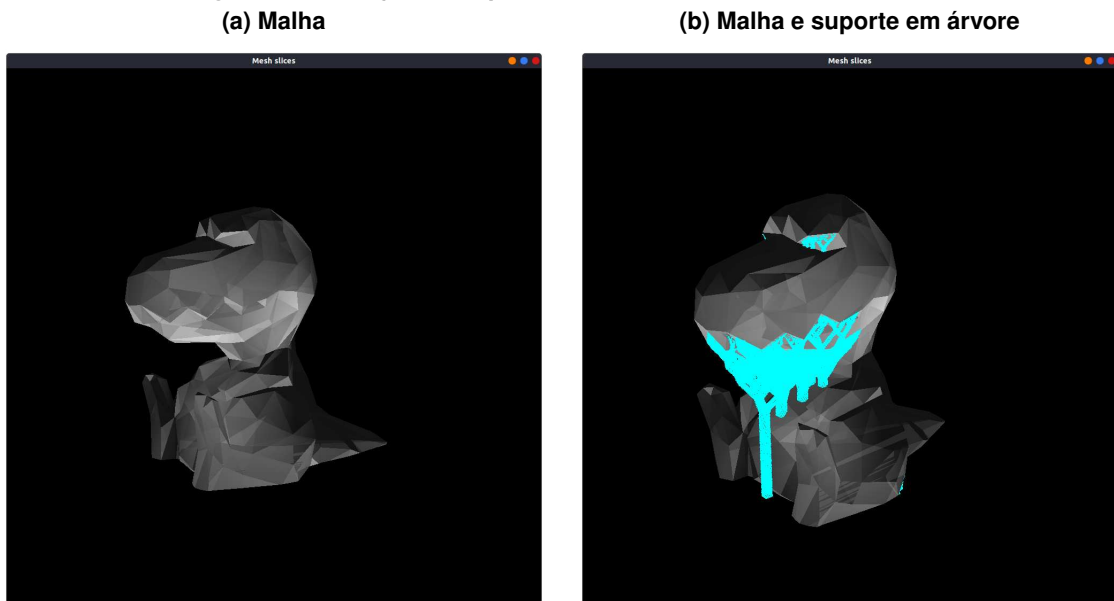
Fonte: Autoria própria.

à cauda (isso ocorre nos dois lados do modelo). Pode-se notar que no focinho o suporte foi dividido em quatro árvores, todas terminando em diferentes camadas.

4.2 Análise de tempo do algoritmo

A Tabela 1 apresenta os resultados de tempo para a computação dos suportes usando diferentes estratégias: a básica, a considerando autossuporte e a de árvores. O tempo reportado

Figura 39 – Geração de suporte em árvore do modelo Dinossauro.



Fonte: Autoria própria.

corresponde ao intervalo de tempos encontrados na avaliação de dez execuções para cada modelo.

Tabela 1 – Tempos para a geração de suporte comparando estratégias.

Modelo	Tempo (s)		
	Básico	Autossuporte	Árvore
Cilindro dentado	2.76 ± 0.02	2.56 ± 0.04	3.90 ± 0.05
Dinossauro	1.45 ± 0.02	2.74 ± 0.01	4.59 ± 0.05
T	0.081 ± 0.004	0.120 ± 0.004	0.71 ± 0.01
Túnel	0.54 ± 0.01	0.64 ± 0.01	34.7 ± 1.1
Rampa	0.132 ± 0.005	0.165 ± 0.004	0.64 ± 0.01
Turbina	0.65 ± 0.01	0.76 ± 0.01	1.21 ± 0.02

Fonte: Autoria própria.

Pode-se notar que o suporte em árvore pode levar consideravelmente mais tempo que as demais estratégias em alguns modelos, já em outros, embora levando sempre um tempo maior, tem uma duração aceitável para testes; o modelo “Túnel” foi o mais demorado, levando em média 34.7 segundos. Essa disparidade se deve ao fato de que o modelo possui um volume muito maior que os demais, tendo tanto a área de uma camada quanto a altura consideravelmente maiores. Como a distância entre as camadas é fixa para todos os modelos, o modelo “Túnel” possui o maior número de camadas. Além disso, como a densidade de nós também é fixa para os modelos, possui muito mais nós-folha que os demais. O fato de haver uma curvatura do túnel faz com que a densidade de nós seja consistente por muitas camadas, efetivamente não reduzindo o número de candidatos de junção avaliados para cada nó tão rapidamente como nos demais modelos.

Na sequência são realizadas algumas análises sobre características do algoritmo que podem impactar no tempo de execução. A segunda coluna da Tabela 2 contém o número de vértices de cada modelo após o processo de fatiamento que ocorre antes da análise de suporte; a terceira coluna contém o total de nós de árvore gerados para cada modelo (incluindo todos os tipos de nós da árvore). Como a grande maioria dos nós, excetuando apenas os nós que se unem à base, participam do processo de construção da árvore, há correlação direta com a complexidade de geração do suporte do modelo, ou seja, quanto maior o número de nós, mais tempo será necessário para realizar a computação. Pode-se notar que o modelo “Túnel” possui cerca de sete vezes menos vértices que o modelo “Cilindro dentado”, porém um pouco mais de quatro vezes o número de nós. A complexidade do modelo, quanto ao número de polígonos e à complexidade do próprio polígono não é suficiente para influenciar o processo de geração de suporte. A complexidade da análise depende da topologia do modelo, não do número de polígonos.

Tabela 2 – Complexidade do modelo após fatiamento e número de nós gerados.

Modelo	Nº vértices	Nº nós
Cilindro dentado	48656	1696
Dinossauro	11477	1772
T	1654	300
Túnel	7622	7280
Rampa	2929	198
Turbina	8084	277

Fonte: Autoria própria.

De acordo com o processo de busca pelo melhor nó, como descrito na Seção 3.4.3, várias camadas podem acabar sendo analisadas para garantir que um nó encontrado seja de fato o melhor candidato para o nó sendo processado. A Tabela 3 contém estatísticas relacionadas ao número de camadas consideradas (mínimo, mediana e máximo) para cada um dos modelos.

Tabela 3 – Número de camadas avaliadas por nó.

Modelo	Mínimo	Mediana	Máximo
Cilindro dentado	3	3	48
Dinossauro	1	2	75
T	3	3	51
Túnel	2	4	106
Rampa	3	3	62
Turbina	3	7	37

Fonte: Autoria própria.

Pode-se notar que não há grande divergência entre o mínimo e a mediana, indicando que a maioria dos casos são semelhantes ao caso mínimo. Vale notar que, em geral, o algoritmo visita poucas camadas para cada nó. Os máximos encontrados podem ser explicados pelos últimos nós processados, que à procura de um par, acabam percorrendo todas as camadas abaixo.

A Tabela 4 contém medidas sobre o número de nós avaliados em cada busca. O fato de que a distribuição de nós de folha é uniforme ao longo de toda uma camada, faz com que grande parte dos primeiros nós candidatos encontrados já sejam o melhor candidato, uma vez que está sendo utilizada uma busca em largura. Como existe uma maior densidade de nós folha, há um benefício significativo no processamento.

Tabela 4 – Número de candidatos avaliados para melhor nó.

Modelo	Mínimo	Mediana	Máximo
Cilindro dentado	1	1	3
Dinossauro	1	1	9
T	1	1	3
Túnel	1	2	9
Rampa	1	1	2
Turbina	1	2	8

Fonte: Autoria própria.

O tamanho de cada célula, onde os nós são indexados, tem um papel importante na otimização. Nos experimentos, as células foram dadas como regiões quadradas cujo tamanho dos lados é igual a duas vezes a distância entre cada nó-folha distribuído. Isso faz com que cada célula tenha até nove nós na camada inicial e essa característica faz com que a busca por uma junção seja terminada rapidamente nas iterações correspondentes aos das camadas dos nós-folha.

A Tabela 5 mostra a estatística relacionada com o número de células percorridas. O caso mínimo sendo um se deve ao fato de que a própria célula de um nó é sempre percorrida. Os valores máximos com muita discrepância entre os casos médios são resultado da busca em largura tendo que percorrer várias células quando a camada possui poucos candidatos e estes não estão nas proximidades. No caso do modelo 'Túnel', como é um modelo largo, há muitas células e nós, podendo resultar em vastas regiões sem nenhum nó.

Tabela 5 – Número de células percorridas por nó.

Modelo	Mínimo	Mediana	Máximo
Cilindro dentado	1	18	1299
Dinossauro	1	25	321
T	1	18	127
Túnel	1	52	8092
Rampa	1	18	37
Turbina	1	32	360

Fonte: Autoria própria.

A Tabela 6 contém as estatísticas sobre o total de nós analisados por modelo, contabilizando a quantidade de candidatos produzidos assim como a de nós comparados para encontrar o mais próximo. O mínimo é tal pois sempre é produzido pelo menos um candidato: a base. Como a busca por um nó mais próximo em uma camada usa uma busca em largura, cada anel de células do limite de busca precisa ser analisado. Como a busca mínima (quando o candidato

encontrado está na mesma célula) envolve nove células e cada célula pode conter até nove nós devido à escolha das dimensões das células, um valor comum é oitenta e um. Valores muito maiores são devidos a buscas em diferentes camadas que repetem esse processo e o uso da busca iterativa ao invés da busca em largura, que acaba passando por todos os nós restantes da camada.

Tabela 6 – Total de nós analisados para produção de um nó.

Modelo	Mínimo	Mediana	Máximo
Cilindro dentado	1	16	221
Dinossauro	1	24	405
T	1	16	74
Túnel	1	20	91
Rampa	1	21	77
Turbina	1	23.5	86

Fonte: Autoria própria.

4.3 Análise de volume dos suportes

Embora o tempo para computação das árvores possa ser maior, a economia de volume de material é notável. A Tabela 7 reporta o volume de suporte em relação ao volume do modelo original, sem suporte. Um volume relativo de 100%, por exemplo, significa que o volume do suporte é igual ao volume do modelo original, sem suporte, efetivamente duplicando a quantidade de material necessária para imprimir o modelo.

Tabela 7 – Volume relativo de material para a geração de suporte comparando estratégias.

Modelo	Volume relativo (%)		
	Básico	Autossuporte	Árvore
Cilindro dentado	15.85	15.85	0.31
Dinossauro	37.36	19.16	3.04
T	72.53	72.53	1.58
Túnel	55.16	55.07	1.22
Rampa	59.31	7.11	0.14
Turbina	283.99	190.66	12.45

Fonte: Autoria própria.

O modelo “Turbina” passou de 283.99%, com suporte básico, para 12.45%. Isso se deve ao fato de as hélices do modelo possuírem um pequeno volume, porém, quando projetadas até a base, correspondente ao método básico, a área descrita e, por consequência, o volume se tornam muito maiores.

Os volumes tanto do modelo “Cilindro Dentado” quanto do modelo “Túnel” não mudaram significativamente entre a estratégia básica e a que considera o ângulo de autossuporte, pois não há regiões aparentes com ângulo maior que 45 graus, logo praticamente nenhuma região pode ser simplificada. A pequena diferença observada no modelo “Túnel” se deve a pequenos desalinhamentos entre os polígonos que descrevem as camadas.

Com a estratégia da árvore, porém, há uma melhoria considerável, passando de 15.85% para 0.31% de material necessário no caso do “Cilindro Dentado” e de 55.16% para 1.22% no caso do “Túnel”. O caso com os maiores benefícios ao usar a estratégia de árvore ocorreu para a “Rampa”, que teve uma melhoria de mais de 423 vezes no uso de material extra necessário quando comparado à estratégia básica.

5 CONSIDERAÇÕES FINAIS

A computação de estruturas de suporte é uma tarefa essencial no processo de planejamento de manufatura aditiva para permitir a fabricação correta de peças com regiões suspensas. Estruturas de suporte, no entanto, geram impacto em quantidade de material para impressão e tempo de processamento.

Este trabalho considerou a construção de estruturas de suporte baseadas em árvores para diminuir a quantidade de material necessário. O método desenvolvido é baseado em amostrar pontos de suporte sobre regiões suspensas em fatias de um modelo, e unir os pontos através da criação de junções que formam ramos de uma árvore. A busca exaustiva pelas melhores junções é um processo computacionalmente proibitivo e, por esse motivo, o método adota uma heurística de busca em largura por pares mais próximos. Os experimentos realizados demonstram a eficácia do método para redução de material necessário para as estruturas de suporte. Em relação ao tempo computacional, o método apresentou em muitos casos de teste tempos pouco maiores do que aqueles de métodos simples, embora para alguns tipos de modelos geométricos possa ser exigido um tempo computacional bem maior.

Como possibilidade de trabalhos futuros, há melhorias no método que podem ser consideradas. Atualmente, a base das árvores é dada pela fatia de base do modelo ou por uma fatia intermediária em que a árvore se sustenta sobre o próprio modelo. Isto ocorre pois o tronco de uma árvore é sempre vertical e imediatamente abaixo do ponto sendo avaliado. Alternativamente, em alguns casos, os ramos das árvores poderiam ser conectados ao próprio modelo. Desta forma, é possível diminuir tanto o tamanho quanto a quantidade de ramos e, consequentemente, diminuir o volume da árvore e a quantidade de material necessário à sua fabricação. Divergindo dessa estratégia de usar o material do próprio modelo como sustentação, pode ser interessante desviar as estruturas de árvores para que sua base seja preferencialmente sobre a fatia da base do modelo. Esse tipo de estratégia diminui o contato do suporte com o modelo, facilitando a sua remoção após a impressão e diminuindo deformações no modelo produzidas pelo contato de estruturas de suporte. Outra melhoria a fim de diminuir o contato do suporte com o modelo, seria adaptar o método de conexão dos pontos de suporte (folhas da árvore) com o modelo para que fosse o mais fino possíveis, similar ao que é feito por Vanek, Galicia e Benes (2014), na construção geométrica. Finalmente, a espessura dos ramos pode ser parametrizada de maneiras mais complexas a fim de garantir estabilidade estrutural e, ao mesmo tempo, diminuir a quantidade de material necessário.

REFERÊNCIAS

- CHALASANI, K.; JONES, L.; ROSCOE, L. Support generation for fused deposition modeling. *In: 1995 International Solid Freeform Fabrication Symposium*. [s.n.], 1995. Disponível em: <http://utw10945.utweb.utexas.edu/Manuscripts/1995/1995-28-Chalasan.pdf>.
- CHEN, X.; MCMAINS, S. Polygon offsetting by computing winding numbers. *In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. [S.l.: s.n.], 2005. v. 4739, p. 565–575.
- FIGUEIREDO, L. H. de; CARVALHO, P. C. P. **Introdução à Geometria Computacional**. Brazil: Colóquio de Matemática Aplicada - IMPA, 1991.
- GILBERT, E. N.; POLLAK, H. O. Steiner minimal trees. **SIAM Journal on Applied Mathematics**, Society for Industrial and Applied Mathematics, v. 16, n. 1, p. 1–29, 1968. ISSN 00361399. Disponível em: <http://www.jstor.org/stable/2099400>.
- JIANG, J.; STRINGER, J.; XU, X. Support optimization for flat features via path planning in additive manufacturing. **3D Printing and Additive Manufacturing**, v. 6, p. 171–179, 06 2019.
- JIANG, J.; XU, X.; STRINGER, J. Support structures for additive manufacturing: A review. **Journal of Manufacturing and Materials Processing**, v. 2, n. 4, 2018. ISSN 2504-4494. Disponível em: <https://www.mdpi.com/2504-4494/2/4/64>.
- JIANG, J.; XU, X.; STRINGER, J. Optimization of process planning for reducing material waste in extrusion based additive manufacturing. **Robotics and Computer-Integrated Manufacturing**, v. 59, p. 317–325, 2019. ISSN 0736-5845.
- MINETTO, R. *et al.* An optimal algorithm for 3d triangle mesh slicing. **Computer-Aided Design**, v. 92, p. 1–10, 2017. ISSN 0010-4485. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0010448517301215>.
- O'ROURKE, J. **Computational Geometry in C**. USA: Cambridge University Press, 1994. ISBN 0521440343.
- SCHNEIDER, P. J.; EBERLY, D. **Geometric Tools for Computer Graphics**. USA: Elsevier Science Inc., 2002. ISBN 1558605940.
- SHAPELY. **The Shapely User Manual**. 2021. <https://shapely.readthedocs.io/en/stable/manual.html>. [Online; accessed 04-November-2021].
- THOMAS, D. **The development of design rules for selective laser melting**. 2009. Tese (Doutorado) — University of Wales Institute, Cardiff, 2009.
- VANEK, J.; GALICIA, J. A. G.; BENES, B. Clever support: Efficient support structure generation for digital fabrication. **Comput. Graph. Forum**, v. 33, n. 5, p. 117–125, 2014.
- VOLPATO, N.; CARVALHO, J. d. Introdução à manufatura aditiva ou impressão 3D. *In: Manufatura Aditiva: tecnologias e aplicações da impressão 3D*. 1st. ed. São Paulo, SP, Brazil: Blucher, 2017. p. 15–30. ISBN 9788521211518.
- YU-AN, J.; YONG, H.; JIAN-ZHONG, F. Support generation for additive manufacturing based on sliced data. **The International Journal of Advanced Manufacturing Technology**, v. 80, p. 2041–2052, 05 2015.