

UNIVERSIDADE FEDERAL DO PAMPA

Rafael de Oliveira Jarczewski

Segmentação de objetos para impressão 3D

Alegrete
2023

Rafael de Oliveira Jarczewski

Segmentação de objetos para impressão 3D

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Marcelo Resende Thielo

Alegrete
2023

Rafael De Oliveira Jarczewski

Segmentação de objetos para impressão 3D

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em: 03 de julho de 2023

Banca examinadora:

Prof. Dr. Marcelo Resende Thielo

Orientador

UNIPAMPA

Prof. Dr. Fabio Paulo Basso

UNIPAMPA

Prof. Me. Jean Felipe Patikowski Cheiran

UNIPAMPA



Assinado eletronicamente por **MARCELO RESENDE THIELO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/07/2023, às 05:45, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **JEAN FELIPE PATIKOWSKI CHEIRAN, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/07/2023, às 09:42, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **FABIO PAULO BASSO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/07/2023, às 13:14, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1171847** e o código CRC **B9458B2D**.

Dedico este trabalho a minha família, por todo auxílio e apoio que recebi durante a elaboração dessa monografia, sem todo o apoio e incentivo esse trabalho não seria possível.

Agradecimentos

Primeiramente gostaria de agradecer minha família, sem eles esse trabalho sem dúvida não seria possível. Agradecer todas as vezes que esquecia de me alimentar e minha mãe trazia algo para mim. Agradeço ao meu irmão Lucas por ter comprado uma impressora 3D e deixado eu brincar com ela, se não fosse por isso não teria tido interesse em tal área. Agradecer ao meu pai, pelos dias e meses que passa na estrada a fim de ajudar no sustento da família, muitas vezes sem poder passar conosco datas importantes. Agradecer meu primo Guilherme, por gastar horas viajando de Uruguaiana a Alegrete e por ser essa pessoa incrível e divertida. Meu avô Arisoli, por na infância ter auxiliado a despertar o meu interesse na ciência e maravilhas do mundo, mesmo que hoje ele não consiga se lembrar disso. Além disso, gostaria de agradecer a paciência deles por conseguir me aturar, sempre estar dispostos a ajudar e por apoiarem minhas escolhas.

Também gostaria de agradecer aos meus amigos, primeiro ao Arthur Chavasco por estar comigo antes mesmo de iniciar essa jornada e por me auxiliar em escapar do mundo acadêmico e sempre me apoiar, Douglas Vinícius Ledur Dullius por ter aberto meus olhos sobre minhas atitudes e auxiliar no meu crescimento pessoal, bem como as aulas de lógica apurada com que tanto aprendi, Gustavo Tamiosso por me ensinar a investigar e procurar o motivo das coisas, Mágino Leguissamo pelas belas discussões políticas e ajudar a me valorizar, um companheiro de batalha, Mariana Costa por todas piadas sem graça, mas que me fizeram rir muito, que contou durante esses anos, Vitor Siqueira por ter me dado aulas em disciplina e prioridades, e Vinicius de Souza da Silva sempre lá quando precisei de ajuda e aberto a escutar meus lamentos.

Também gostaria de agradecer ao meu orientador, professor Marcelo Thielo, que trouxe essa ideia de trabalho e permitiu que eu tomasse esse desafio para mim, bem como os conselhos para vida, e as discussões sócio-políticas.

Agradeço a todos vocês por terem feito parte dessa jornada e tornado ela possível. Vocês são incríveis e seguindo o poema de Cris Pizzimenti vocês fazem parte da minha colcha de retalhos e cada pedacinho do tecido de vocês está costurado para sempre em minha colcha de retalhos.

“O fracasso é apenas uma oportunidade de começar de novo. Porém, desta vez, com
mais sabedoria“

(Tio Iroh, Avatar: A Lenda de Aang)

Resumo

O mercado de Manufaturas Aditivas (MAs) popularmente conhecido como impressão 3D está em constante crescimento. Indubitavelmente as MAs se provaram úteis e revolucionárias para diversos segmentos da indústria. Entretanto, as máquinas de impressão 3D possuem uma limitação física atrelada ao seu volume de impressão, o que leva à necessidade de segmentar peças maiores que o volume da cama de impressão. Dividir um objeto 3D de maneira inteligente não é algo trivial e muitos trabalhos já estão sendo realizados nesse contexto, contudo ainda há poucos estudos voltados para o mercado de MAs. O que torna esse processo complexo, é a quantidade de variáveis que devem ser levadas em consideração para a segmentação. Diante das dificuldades presentes no processo de particionamento, este trabalho visa apresentar uma solução de segmentação de objetos 3D levando em consideração as dimensões de impressão. O trabalho visa produzir uma solução inteligente automatizada que receberá um objeto 3D como entrada e retornará a menor quantidade de segmentos aproveitando o volume de impressão em sua totalidade. O presente protótipo possui um leitor de objetos bem como uma interface de visualização, além de possuir um método para encontrar interseções entre vetores e faces, sendo necessário definir características que definem a qualidade de um segmento, bem como a escolha do método de aprendizado.

Palavras-chave: Impressão 3D. Manufatura Aditiva. Segmentação de Objetos. Aprendizado de máquina.

Abstract

The Additives Manufacturing (ADs) market, popularly known as 3D printing, is constantly growing. Undoubtedly, ADs have proven to be useful and revolutionary for various industry segments. However, 3D printing machines have a physical limitation related to their printing volume, which leads to the segmentation of parts for printing. Splitting a 3D object intelligently is not trivial and many works have already been carried out in this context, but there are still few studies aimed at the ADs market. What makes this process complex is the number of variables that must be taken into account for segmentation. Faced with the difficulties of the partitioning process, this work aims to present a 3D object segmentation solution taking into account the printing dimensions. The work aims to produce an intelligent automated solution that receives a 3D object and returns the smallest number of segments, taking full advantage of the print volume. The prototype implements an object scanner, an object viewer, and a method to find intersections among vertexes and sides that receives as input the quality features of the segment and the chosen learning method.

Keywords: 3D printing. Additive Manufacturing. Object Segmentation. Machine learning.

Lista de figuras

Figura 1 – Estrutura de arquivo .obj	15
Figura 2 – Exemplo de arquivo .obj	16
Figura 3 – Estrutura binária de arquivo .stl	17
Figura 4 – Plano cartesiano	21
Figura 5 – Vetor no espaço	22
Figura 6 – Plano no espaço	23
Figura 7 – Plano a partir de 3 pontos	24
Figura 8 – Representação do vetor contido no plano	24
Figura 9 – Representação de intersecção entre vetor e plano	25
Figura 10 – Representação de um vetor que não perfura e não está contido ao plano	26
Figura 11 – Processo de divisão	35
Figura 12 – Demonstração do visualizador 3D	39
Figura 13 – Diagramas de classe UML - Leitor de Objetos 3D	40
Figura 14 – Diagramas de classe - visualizador 3D	40
Figura 15 – <i>Pipeline</i> para fechar aberturas	42
Figura 16 – Contornos do objeto	42
Figura 17 – Contornos fechados	43
Figura 18 – Criação de uma face através dos pontos obtidos com a segmentação . .	43
Figura 19 – Exemplo de objeto e segmentação	44
Figura 20 – Exemplo da implicação do ângulo	45
Figura 21 – Exemplo de objetos curvos.	46
Figura 22 – Exemplo de objetos com menos curvaturas.	46
Figura 23 – Exemplo de comportamento da função de pontuação	50
Figura 24 – Objeto de entrada para o Algoritmos Genéticos (AG). Dimensões: 13.8722cm × 21.2815cm × 15.8325cm	52
Figura 25 – Comparação entre configurações	53

Lista de siglas

ABS Acrilonitrila butadieno estireno

AG Algoritmos Genéticos

API Application Programming Interface

BCC-problem The Boxes Class Cover problem

CAD Computer Aided Design

DFS Deep-first-search

FDM Fused Deposition Modelling

MA Manufatura Aditiva

PCA Principal Component Analysis

PLA Biopolímero Acido Poliláctico

RNN Redes Neuras Recorrentes

RPC-problem Rectilinear Polygon Covering problem

SLS Selective Laser Sintering

SoMBB Surface-Oriented Minimum Bounding Box

STL Standard Tessellation Language

Sumário

1	INTRODUÇÃO	12
1.1	Motivação	13
1.2	Objetivo	13
1.2.1	Objetivos específicos	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Representação digital de objetos tridimensionais	15
2.1.1	Object Files	15
2.1.2	Standard Data Format	16
2.2	Impressão 3D	17
2.2.1	Ferramentas de modelagem 3D	17
2.2.2	Ferramentas de Fatiamento de objetos 3D	18
2.2.3	Manufatura aditiva	19
2.2.4	Tipos de manufatura aditiva	20
2.2.4.1	Material sólido	20
2.2.4.2	Material Líquido	20
2.2.4.3	Material em Pó	20
2.3	Geometria analítica espacial	21
2.3.1	Vetor	21
2.3.2	Plano	22
2.3.3	Intersecção entre reta e plano	24
2.3.4	Coordenadas baricêntricas	26
2.4	Curvatura média	27
2.5	Abordagens inteligentes	27
2.5.1	Algoritmo Genético	28
2.6	Complexidade	28
3	TRABALHOS RELACIONADOS	32
3.1	Protocolo de pesquisa	32
3.2	Segmentação semântica	32
3.3	Octree	33
3.4	Análise de Componente Principal	33
3.5	Segmentação de objetos 3D	34
3.6	Diagramas de Voronoi 3D	36
4	DESENVOLVIMENTO	38
4.1	Definição de ferramentas	38
4.2	Abordagem inicial	38
4.2.1	Funcionalidades gerais	38

4.2.1.1	Leitor de arquivos .obj	39
4.2.1.2	Visualizador 3D	39
4.2.1.3	Interseções	40
4.2.1.4	Ajuste de imperfeições em objetos.	41
4.2.1.5	Segmentação de objetos	43
4.3	Função custo	44
4.3.1	Implementação da abordagem de aprendizado	47
5	RESULTADOS	49
5.1	Equação de pontuação	49
5.2	Aprendizado de máquina	52
6	CONSIDERAÇÕES FINAIS	54
	REFERÊNCIAS	55

1 Introdução

O mercado de MAs popularmente conhecido como impressão 3D está em constante crescimento. Crescimento este, que durante o ano de 2020 foi de 7,5% para 19,5% no ano de 2021 Bergels (2022). Indubitavelmente as MAs se provaram úteis e revolucionárias para diversos segmentos da indústria, utilizadas em áreas basilares da humanidade como saúde (LIAW; GUVENDIREN, 2017) e educação (ASSANTE; CENNAME; PLACIDI, 2020) moradia (PETERS, 2019) e indo para áreas como moda (VANDERPLOEG; LEE; MAMP, 2017) ou ramo alimentício (SINGHAL et al., 2020).

Contudo, o que permite a inserção dessa tecnologia em segmentos distintos com tanta adesão? O primeiro fator é a capacidade personalizável adquirida com a impressão 3D, visto que basta termos um modelo de objeto com o formato suportado pelo fatiador¹ da sua impressora (que é o suficiente para transformar o modelo digital em físico).

Um dos grandes benefícios que tal método traz consigo são, por exemplo, a alta precisão na criação de peças feitas sob medida e com alterações específicas, minimizando o tempo gasto em projetos complexos e reduzindo o alto custo de prototipagem. Em complemento a isso, a vasta gama de materiais que podem ser utilizados nas MAs possibilita adaptar seu protótipo para construções de peças que necessitam de uma resistência maior ou precisam de um atributo específico.

Entretanto, as máquinas de impressão 3D possuem uma limitação física atrelada ao seu volume de impressão. Durante o trabalho com MAs é comum desejarmos produzir uma peça maior que as dimensões volumétricas da máquina, visto que não existe nenhum processo popular de segmentação automática de objetos atrelado a dimensões de impressão, o que é realizado manualmente em ferramentas de modelagem 3D, tornando o trabalho penoso e demorado.

Segmentar um objeto 3D não é algo trivial e muitos trabalhos já estão sendo realizados nesse contexto, contudo ainda há poucos estudos voltados para o mercado de MAs.

A justificativa da complexidade reside na quantidade de variáveis que devem ser levadas em consideração durante a segmentação. No decorrer da impressão, é de extrema importância maximizar o volume de segmento por impressão, visto que quanto menos segmentos forem necessários, menor será o tempo de impressão.

Ao mesmo tempo, é necessário levar em consideração o ângulo de corte, a posição do corte, bem como a espessura das extremidades que estão sendo cortadas, dado que as peças serão agrupadas posteriormente e não é interessante existir espessuras finas em cortes. Visto dificultam o processo pós-impressão, já que espessuras muito finas dificultam a inserção de pinos além de ter pouca área de contato para colagem.

Diante das dificuldades presentes no processo de segmentação, este trabalho visa

¹ Softwares de fatiamento são ferramentas utilizadas para transformar o objeto em fatias geométricas horizontais para impressão. Explicação mais detalhada na seção 2

apresentar uma solução de particionamento automático de objetos 3D levando em consideração as dimensões de impressão, com o objetivo de permitir a impressão volumétrica de peças que excedem seu volume, como partes internas de carros, móveis, itens decorativos e qualquer outra peça desejada.

1.1 Motivação

Tendo em mente a emergência do mercado de manufatura, há uma série de questões em aberto que ainda não foram resolvidas. Produzir peças maiores do que o volume das máquinas de impressão 3D é algo relativamente corriqueiro, visto que impressoras básicas que possuem preços acessíveis para o consumidor geral, em média, contêm dimensões de impressão de $200 \times 200 \times 200$ mm (FILAMENT2PRINT, 2018), um pouco menor do que uma régua de escritório.

Este fator leva muitos comerciantes a particionar seus pedidos. No entanto, esse processo é feito manualmente através de ferramentas como Blender e Meshmixer², onde planos são posicionados ao longo do objeto e cada plano objetiva segmentar a peça em duas partes.

Esse trabalho é muito minucioso e, além de adicionar várias horas ao processo de impressão, também aumenta seu custo. Visto que não basta segmentar o objeto em partes aleatórias que caibam no volume de impressão, é necessário considerar outros fatores.

Desse modo, o custo para o consumidor final torna-se maior, além de estar sujeito à experiência do operador de impressora com softwares de modelagem.

1.2 Objetivo

Dito isto, este trabalho visa avançar nos estudos de um método genérico de segmentação volumétrica inteligente, levando em consideração as dimensões da impressora utilizada, com o objetivo de encurtar o tempo de fabricação de produtos de manufatura e, conseqüentemente, reduzir os custos do produto final.

O segmentador deverá receber um objeto arbitrário juntamente com as dimensões da impressora e retornar a menor quantidade de segmentos necessários para a impressão.

1.2.1 Objetivos específicos

Para este trabalho temos 3 objetivos específicos que podem ser descritos da seguinte maneira:

- Realizar uma revisão literária a fim de encontrar técnicas que podem ser usadas para a segmentação de objetos 3D: Neste objetivo específico, será realizada uma pesquisa

² Ferramentas de modelagem 3D utilizadas para produção de animações, jogos, etc. Mais detalhes durante a seção 2

abrangente e detalhada da literatura científica e técnica relacionada à segmentação de objetos 3D. Serão identificadas e analisadas as diferentes técnicas existentes, como algoritmos de segmentação, métodos de aprendizado de máquina e abordagens baseadas em geometria. A revisão literária fornecerá uma base sólida de conhecimento para a seleção e desenvolvimento de técnicas adequadas de segmentação de objetos 3D.

- Propor uma equação custo que descreva de alguma forma os segmentos encontrados: Neste objetivo específico, será proposta uma equação de custo que permita descrever de forma quantitativa e objetiva os segmentos encontrados na etapa de segmentação de objetos 3D. A equação de custo considerará diversos parâmetros relevantes para o contexto de impressão. Através dessa equação, será possível avaliar e comparar diferentes segmentações, auxiliando na seleção da melhor solução para a impressão 3D.
- Utilizar Algoritmos Genéticos (AG) para validar a equação de custo proposta e como um ponto de partida para uma solução genérica: Neste objetivo específico, pretende-se empregar Algoritmos Genéticos (AG) como uma abordagem de otimização para validar a eficácia da equação de custo proposta. Os AGs serão utilizados para buscar a combinação ideal de parâmetros da equação de custo que resulte em segmentações de objetos 3D de alta qualidade. Além disso, o uso de AGs será explorado como um ponto de partida para o desenvolvimento de uma solução genérica para a segmentação de objetos 3D, permitindo a adaptação e generalização da abordagem para diferentes tipos de objetos e cenários de impressão 3D.

2 Fundamentação teórica

Neste capítulo, vamos introduzir os principais conhecimentos necessários para o trabalho. Será feita uma breve contextualização sobre o meio de impressão 3D e, em seguida, serão abordados alguns conceitos necessários de geometria, bem como uma explicação sobre os algoritmos de aprendizado utilizados neste trabalho.

2.1 Representação digital de objetos tridimensionais

Nesta seção, iremos explicar algumas formas de representação de objetos tridimensionais no meio digital. Vamos demonstrar os dois principais formatos de arquivos e quais são suas principais características.

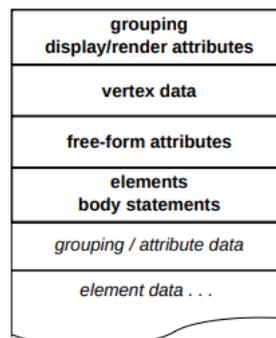
Existem diversas maneiras de representar um objeto tridimensional digitalmente. O modo mais comum e simples é através de uma representação analítica, ou seja, de triplas ordenadas (x, y, z) que são ligadas por segmentos de reta, formando assim faces, sendo o conjunto de todas essas faces denominado de malhas poligonais.

2.1.1 Object Files

A extensão `.obj`, conhecida como *Wavefront Object Files* (MURRAY; VANRYPER, 1996), é considerada universal e aceita por quase todo sistema de visualização tridimensional. Sua estrutura é simples e textual, o que a torna uma escolha comum quando é necessário ler e tratar informações de objetos (Figura 1).

As informações de grupo encontradas no cabeçalho definem coleções de elementos com o objetivo de simplificar manipulações. Além disso, podemos definir grupos de mesclagem responsáveis por identificar elementos adjacentes a serem unidos. Também é possível criar um rótulo para objetos inteiros através da instrução de nomeação de objetos (MURRAY; VANRYPER, 1996).

Figura 1 – Estrutura de arquivo `.obj`



Fonte: Coutinho (2020, pg. 4)

As informações de vértices são definidas pela letra v , seguida de três números reais representando o terno ordenado (x, y, z) (Figura 2). Além disso, também é possível identificar as coordenadas de texturas trocando a chave v por vt , ou pode-se encontrar os vetores normais substituindo por vn .

Figura 2 – Exemplo de arquivo .obj

```
# Exemplo de arquivo Wavefront OBJ

# Vértices
v 0.0 0.0 0.0
v 1.0 0.0 0.0
v 0.0 1.0 0.0

# Coordenadas de textura
vt 0.0 0.0
vt 1.0 0.0
vt 0.0 1.0

# Faces
f 1/1 2/2 3/3
```

Fonte: Font: Autor

Em seguida, temos as representações de formas livres, sejam elas curvas ou superfícies. Nessa parte, pode-se representar o tipo de curva/superfície da mesma forma que pode ser descrito com elementos de estado. Cada informação de curva deve ser especificada através da chave *cstype*, e quando necessário identificar o grau, usa-se *deg*, em casos de matrizes *bmat*.

Posteriormente, as linhas e faces são representadas utilizando a chave f , seguida de um conjunto de três valores inteiros separados por $/$, correspondendo aos índices das seguintes informações $v/vt/vn$.

Após isso, caso existam outros elementos em um mesmo arquivo .obj, a estrutura se repete a partir das definições de grupos (COUTINHO, 2020). Apesar de sua manipulação ser simples, devido à sua representação textual, os arquivos costumam ter tamanhos muito grandes.

2.1.2 Standard Data Format

No contexto da impressão 3D, é comum nos depararmos com a extensão Standard Tessellation Language (STL). O principal motivo de sua utilização é a abstração de informações desnecessárias para a impressão, como, por exemplo, coordenadas de textura. Analogamente à extensão .obj, sua representação ocorre por meio de triplas ordenadas

para identificar os vértices, e suas faces são definidas pelos índices dos vértices (Figura 3). A grande vantagem é sua representação binária, gerando arquivos simples e compactos que podem ser facilmente lidos por *softwares* de fatiamento (BURNS, 1989).

Figura 3 – Estrutura binária de arquivo .stl

Bytes	Tipo de Data	Descrição
80	ASCII	Cabeçalho
4	unsigned integer	Número de faces no arquivo
4	float	i da normal
4	float	j da normal
4	float	k da normal
4	float	x do vértice 1
4	float	y do vértice 1
4	float	z do vértice 1
4	float	x do vértice 2
4	float	y do vértice 2
4	float	z do vértice 2
4	float	x do vértice 3
4	float	y do vértice 3
4	float	z do vértice 3
2	unsigned integer	Contagem de bytes de atributo

Fonte: autor

2.2 Impressão 3D

Nesta seção, serão apresentadas as principais características da impressão 3D, bem como será fornecida uma explicação sobre a sua importância no mercado. Além disso, serão definidas as ferramentas necessárias durante o processo de impressão, a fim de compreender em qual parte do processo nossa solução se situa.

2.2.1 Ferramentas de modelagem 3D

O primeiro passo para realizar uma impressão 3D é possuir um arquivo digital que represente um modelo 3D. O processo de criação de um modelo tridimensional digital já está solidificado no mercado e em constante evolução.

Atualmente, existem diversas ferramentas especializadas que permitem a produção de objetos 3D. Essas ferramentas são utilizadas em diferentes áreas, desde jogos e prototipagem de joias e produtos decorativos até ferramentas mais robustas e profissionais voltadas para a prototipagem industrial.

O mercado de jogos se tornou tão popular que trouxe consigo uma atenção maior para a modelagem, fomentando um mercado de *softwares* de modelagem especializados para esses casos.

Ferramentas como Blender e Cinema 4D são exemplos de *softwares* robustos focados no processo de modelagem e texturização de peças.

Paralelo a esse mercado, também se apresenta o ferramental de desenhos auxiliados por computador popularmente conhecido como Computer Aided Design (CAD), que pode

ser definido como um sistema para auxiliar a criação, modificação, análise e otimização de design (SARCAR; RAO; NARAYAN, 2008).

Como demonstra o livro do qual foi retirada essa definição, o processo de *design* industrial se beneficiou muito das ferramentas de prototipagem. A capacidade de produzir com precisão peças industriais, como motores e/ou arquiteturas de prédios através de ferramentas de *software*, gerou um novo método de desenho industrial.

Além disso, trouxe benefícios tangíveis e intangíveis, como a melhoria da qualidade do trabalho, um aperfeiçoamento de detalhes, a facilidade de analisar os protótipos gerados e a flexibilidade durante o processo de prototipagem.

Ferramentas como essas, sejam elas utilizadas na indústria cinematográfica, de jogos, automobilística ou civil, potencializam o mercado de impressão 3D. Além disso, *softwares* como esses possuem centenas de *plugins* e *add-ons* que permitem facilitar o processo de prototipagem.

Recursos como geração procedural de formas e correção de imperfeições são exemplos de funcionalidades adicionais que essas ferramentas possuem. Eles permitem a incorporação de algoritmos, como proposto neste trabalho, de segmentação de objetos para impressão.

2.2.2 Ferramentas de Fatiamento de objetos 3D

Após ter em mãos um arquivo com uma nuvem de pontos e sequência de faces que representa um objeto 3D, produzido por uma ferramenta de modelagem ou digitalizado, também é comum realizar a conversão desses arquivos para formatos STL, visto que este consiste em um padrão mais simples.

Então, de alguma forma, é necessário traduzir essas informações em uma sequência de segmentos geométricos distribuídos por um plano, a fim de dar sentido à máquina de impressão. Os *softwares* responsáveis por fazer essa tradução, bem como proporcionar uma série de funcionalidades adicionais, são conhecidos como **fatiadores** (*slicers*).

Essas ferramentas cortam os objetos em camadas uniformes com determinada espessura, de acordo com as especificações da máquina. Seu objetivo é proporcionar as melhores condições para impressão, desde a segmentação em camadas dos objetos até a construção de estruturas de suporte para partes inclinadas do objeto em relação à mesa de impressão.

Em máquinas¹ que utilizam matéria-prima líquida, como resina, os fatiadores costumam permitir a inserção de pequenos furos para impedir o acúmulo de material e deformações causadas pelo vácuo.

Outro recurso bastante utilizado é o preenchimento inteligente das peças, uma vez que realizar impressões sólidas consome muito material e aumenta o tempo necessário para

¹ Tipo de manufatura popular no mercado, visto que a precisão das peças é muito apurada. Informações sobre o funcionamento podem ser encontradas em 2.2.4.2.

finalizar a peça. Assim, algoritmos que preenchem de forma inteligente partes estratégicas do objeto ou criam preenchimentos vazados são acoplados a esses fatiadores.

Ferramentas como o LuBan permitem um método de segmentação de objetos semelhante ao descrito em (KONIG et al., 2022). Desenvolvido por pesquisadores do MIT, o *software* possui uma série de características cruciais para a impressão 3D. Além de ser possível segmentar o objeto em partes, também é oferecido um sistema de pinagem para unir as peças após a impressão.

Para poder usufruir dessas vantagens, é necessário adquirir uma licença anual de US\$ 30 ou uma licença permanente de US\$ 150. Apesar de ferramentas como essa serem muito benéficas para o mercado de manufatura, atualmente apenas o LuBan (LUBAN, 2022) promete a segmentação inteligente de objetos 3D, no entanto, ainda não considera o volume de impressão que excede as dimensões da cama de impressão.

2.2.3 Manufatura aditiva

Considerando os recursos até aqui abordados, podemos prosseguir para o processo de impressão no entanto, é necessário compreender o conceito de manufatura aditiva.

A manufatura aditiva consiste em um processo de deposição de material em camadas sucessivas, as quais são determinadas por informações geométricas geradas por um *software* de fatiamento. A criação do modelo digital para o protótipo físico é completamente automatizada, sem qualquer intervenção manual durante a adição do material.

De acordo com (MUNHOZ, 1977) “...o sucesso comercial de uma empresa está associado à sua habilidade em identificar as necessidades dos clientes e, rapidamente desenvolver produtos que possam atendê-las”. Esta afirmação justifica o acolhimento da manufatura aditiva de maneira natural.

O alto consumo de produtos personalizados de alta complexidade tem exigido ao mercado agilidade em alterar o processo de fabricação de peças. Ou seja, os métodos tradicionais de produção em massa não são suficientes para atender todas as necessidades que são levantadas pelo consumidor.

O Globo realizou um estudo sobre a procura de produtos personalizados e constatou que 25% dos consumidores pagariam mais em produtos feitos especialmente para elas (NETO, 2018).

Como explicado em MUV (2019) a impressão 3D encurta o processo de desenvolvimento de um produto, visto que o estágio de prototipagem se torna mais eficiente. Outro fator originário da manufatura aditiva está na precisão dos testes, por conta dos detalhes que podem ser realizados em um modelo digital tridimensional.

Além disso, realizar experimentos com o consumidor final, visto que trazem consigo informações preciosas para o aperfeiçoamento do projeto, ademais reduz os custos de todo o processo de prototipagem, já que economiza o tempo e material na produção de novos experimentos.

2.2.4 Tipos de manufatura aditiva

A classificação do processo de MAs é um assunto bem controverso na academia. Muitos julgam a necessidade de uma a classificação simples (CANEDO-ARGUELLES, 1999) enquanto outros argumentam que essa classificação não é suficiente para atender a necessidade dos engenheiros e clientes, visto que é mais interessante entender quais são os pontos de fusão e gastos de energia (JIMÉNEZ et al., 2019). Vamos focar aqui no método tradicional de classificação por material.

2.2.4.1 Material sólido

Máquinas de manufaturas que utilizam esses materiais são conhecidas como Fused Deposition Modelling (FDMs). Essas máquinas utilizam polímeros termoplásticos como seu principal recurso como, por exemplo, Acrilonitrila butadieno estireno (ABS), Biopolímero Acido Poliláctico (PLA), ceras, etc. Através de bicos extrusores aquecidos, o material é injetado através de pressão em uma superfície aquecida a fim de construir o objeto. Impressoras dessa categoria costumam ser pequenas e compactas, com grande variedade de materiais que raramente são tóxicos (CANEDO-ARGUELLES, 1999).

2.2.4.2 Material Líquido

Equipamentos de estereolitografia conhecidos como (Stereolithography, SL) são máquinas que possuem um tanque de resina líquida fotopolimerizável, ou seja, enrijece quando em contato com luz com determinado comprimento de onda. Essas impressoras possuem um cabeçote com um feixe de luz responsável por polimerizar as camadas controladamente de acordo com as instruções do computador, além de possuir um suporte basilar para o objeto que se move verticalmente. Peças produzidas através dessa técnica possuem uma precisão alta com bom acabamento (CANEDO-ARGUELLES, 1999).

2.2.4.3 Material em Pó

Analogamente a impressoras a base de material sólido, esses equipamentos denominados Selective Laser Sintering (SLS), utilizam material termoplástico, como Nylon, polímeros, etc., porém pulverizado, para confecção das peças.

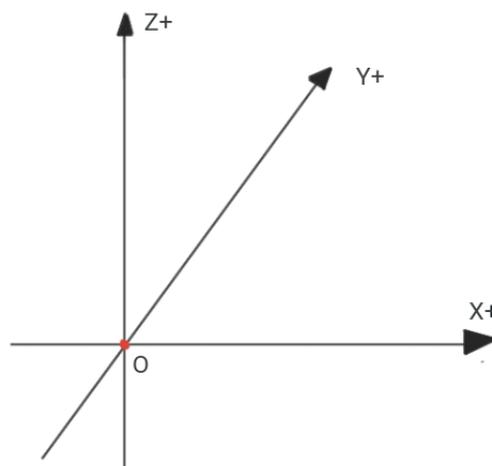
Também existem alguns materiais metálicos que podem ser utilizados. Essas máquinas possuem um tanque onde são despejadas camadas de pó que são pré-aquecidas e, em seguida, um cabeçote controlado pelo sistema da impressora emite um laser que sinteriza o pó até solidificá-lo. Esse processo é realizado camada por camada até o final da peça. (CANEDO-ARGUELLES, 1999)

2.3 Geometria analítica espacial

Alguns conceitos de geometria analítica são necessários para entender o processo de segmentação. Tendo isso em mente, esta seção tem como objetivo apresentar uma breve explicação de alguns desses conceitos necessários.

Podemos definir um espaço euclidiano E dado três eixos mutuamente perpendiculares OX, OY, OZ com intersecção na origem O (BEZERRA, 2010). Todo ponto no R^3 , ou seja espaço tridimensional dos números reais, é representado por uma tripla ordenada (x, y, z) de números reais onde cada um deles representa a projeção vertical do ponto sobre os seus respectivos eixos (Figura 4) (CAVALCANTE, 2015).

Figura 4 – Plano cartesiano



Fonte: Autor

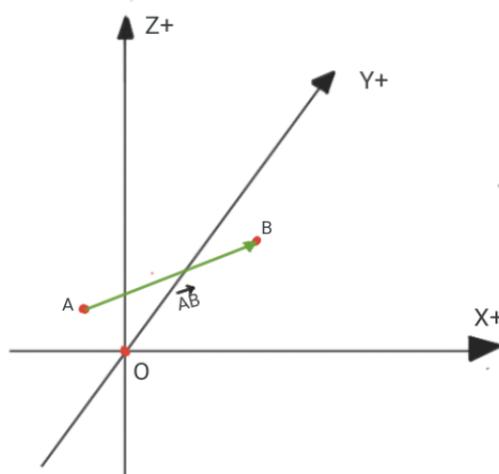
2.3.1 Vetor

Como definido por Bezerra (2010), um vetor pode ser definido como um segmento de reta orientado, ou seja, deve conter sentido, magnitude e direção (Figura 5). Então, um segmento de reta é um par ordenado (A, B) de pontos no espaço e todos os segmentos com o mesmo sentido, direção e magnitude são considerados equipolentes.

1. Podemos definir um vetor através da seguinte equação: $\vec{V} = P_0 + t \cdot \vec{V}_r$ de tal forma que P_0 corresponde a um ponto contido no vetor, t é um número real que serve como deslocamento a partir do P_0 , e \vec{V}_r representa a direção do vetor e pode ser definido por um segmento de reta orientado.
2. Podemos calcular sua magnitude da seguinte forma:

$$|AB| = \sqrt{(B_x - A_x)^2 + (B_y - A_y)^2 + (B_z - A_z)^2} \quad (2.1)$$

Figura 5 – Vetor no espaço



Fonte: Autor

Através da equação 2.1 conseguimos calcular o comprimento de arestas das faces em nossos objetos. Tal informações é crucial para normalização de valores.

2.3.2 Plano

Intuitivamente, quando definimos um sentido e um eixo podemos pensar em uma sequência de planos paralelos a esse eixo com tal sentido entretanto, quando definimos um ponto fixo $P \in R^3$, fica claro que só existe um e somente um plano com esse sentido que contém tal ponto (BEZERRA, 2010).

Podemos definir matematicamente um plano, apenas com um vetor \vec{V} e um ponto fixo P_0 . Um plano π que passe pelo ponto P_0 e contenha \vec{V} como vetor normal dá-se como conjunto de todos os pontos P do R^3 de tal forma que o vetor $\overrightarrow{P_0P}$ é ortogonal a \vec{V} (Figura 6). Ou seja todo ponto $P \in \pi$ é ortogonal ao vetor \vec{V} , logo $\overrightarrow{P_0P} \cdot \vec{V} = 0 \rightarrow (P - P_0) \cdot \vec{V} = 0 \rightarrow P \cdot \vec{V} - P_0 \cdot \vec{V} \rightarrow P \cdot \vec{V} = P_0 \cdot \vec{V}$. Transformando para equação cartesiana, dado que $P_0(x_0, y_0, z_0)$ e $V_0(a, b, c)$ temos que:

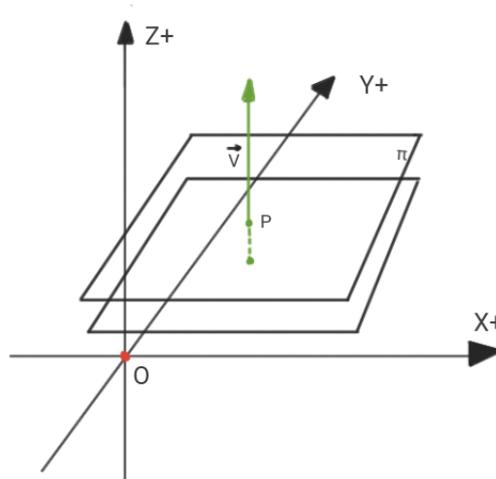
$$\begin{aligned} P \cdot \vec{V} &= P_0 \cdot \vec{V} \\ (x, y, z) \cdot (a, b, c) &= (x_0, y_0, z_0) \cdot (a, b, c) \\ ax + by + cz &= ax_0 + by_0 + cz_0 \end{aligned} \quad (2.2)$$

Definindo $-d = ax_0 + by_0 + cz_0$, de tal forma, podemos reescrever a igualdade 2.2 da seguinte forma:

$$\begin{aligned} ax + by + cz &= -d \\ ax + by + cz + d &= 0 \end{aligned} \quad (2.3)$$

Também é pertinente entendermos como identificar um plano gerado a partir de 3 pontos. Dado três pontos A, B, e C, não-colineares podemos encontrar 2 vetores \overrightarrow{BA}

Figura 6 – Plano no espaço



Fonte: Autor

e \vec{BC} . Assim, podemos calcular o vetor ortogonal através do produto vetorial entre eles eq. 2.5.

Como os três pontos A, B e C estão contidos no plano π , os vetores \vec{BA} e \vec{BC} também estão contidos ao plano, o que leva à conclusão que o vetor ortogonal aos dois vetores também é ortogonal ao plano que os contém (Figura7).

Portanto, através do vetor normal $\vec{N} = \vec{BA} \times \vec{BC}$ e qualquer um dos três pontos podemos encontrar a equação geral do plano utilizando as equações já vistas.

Desenvolvendo o determinante nós conseguimos a seguinte configuração.

$$\vec{V} = DET \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ BA_x & BA_y & BA_z \\ BC_x & BC_y & BC_z \end{vmatrix} \quad (2.4)$$

$$\vec{V} = \hat{i}(BA_y BC_z - BC_y BA_z) + \hat{j}(BA_z BC_x - BC_z BA_x) + \hat{k}(BA_x BC_y - BC_x BA_y) \quad (2.5)$$

A equação 2.5 pode ser interpretado como a equação do plano no qual a face do objeto está contida, visto que o vetor normal é dado através da equação do plano quando $x = y = z = 1$. Tendo a equação geral do plano $\pi : ax + by + cz + d = 0$, temos:

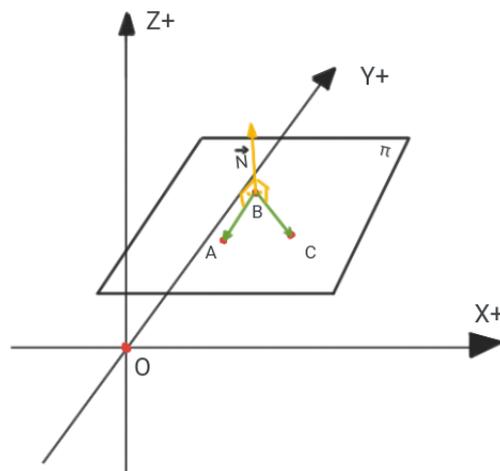
$$\pi : \hat{i}(BA_y BC_z) - \hat{i}(BC_y BA_z)x + \quad (2.6)$$

$$\hat{j}(BA_z BC_x) - \hat{j}(BC_z BA_x)y + \quad (2.7)$$

$$\hat{k}(BA_x BC_y) - \hat{k}(BC_x BA_y)z + d = 0 \quad (2.8)$$

Então conseguimos com a equação 2.6 definir a equação do plano com base nos 3 pontos pertencentes a face.

Figura 7 – Plano a partir de 3 pontos



Fonte: Autor

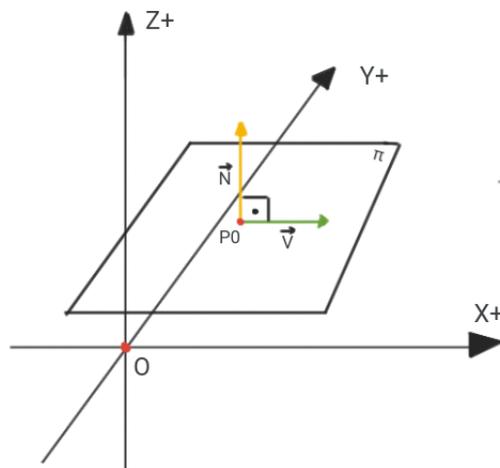
2.3.3 Intersecção entre reta e plano

Dado as definições anteriores podemos agora compreender uma das bases do algoritmo de segmentação. Para podermos segmentar os objetos, é necessário encontrar o ponto de intersecção de vetor e um plano. Vamos focar em três casos:

1. O vetor está contido ao plano:

Dado um vetor $\vec{V} = P_0 + t \cdot \vec{V}_r$ e um plano π com normal \vec{N}_π , a condição é verdadeira, se e somente se, $P_0 \in \pi$ e o vetor \vec{V}_r for perpendicular a \vec{N}_π , ou seja $\vec{V} \subset \pi \leftrightarrow P_0 \in \pi \cap \vec{V}_r \cdot \vec{N}_\pi = 0$ (CAVALCANTE, 2015) (Figura 8).

Figura 8 – Representação do vetor contido no plano



Fonte: Autor

2. Reta fora o plano π :

Podemos calcular o ponto de intersecção entre um vetor², e um plano utilizando o seguinte raciocínio (Figura 9): Dados dois pontos P_0 e P_1 podemos determinar um vetor $\vec{V} = P_0 + t \cdot \overrightarrow{P_0P_1}$, e sabendo que o plano $\pi = ax + by + cz + d = 0$, é possível encontrar a equação paramétrica do vetor, ou seja:

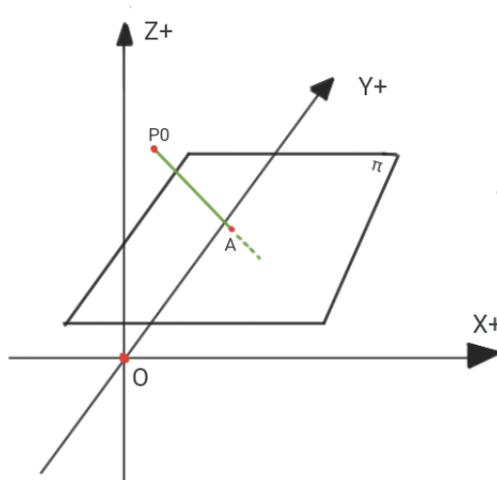
$$\begin{aligned} \vec{V} &= (x_0, y_0, z_0) + t(x_1 - x_0, y_1 - y_0, z_1 - y_0) \\ \vec{V} &= (x_0 + t(x_1 - x_0), y_0 + t(y_1 - y_0), z_0 + t(z_1 - y_0)) \end{aligned} \quad (2.9)$$

com a equação 2.9, basta substituir o valores na equação do plano e isolar o valor de t , da seguinte forma:

$$\begin{aligned} ax + by + cz + d &= 0 \\ a(x_0 + t(x_1 - x_0)) + b(y_0 + t(y_1 - y_0)) + c(z_0 + t(z_1 - y_0)) + d &= 0 \\ ax_0 + at(x_1 - x_0) + by_0 + bt(y_1 - y_0) + cz_0 + ct(z_1 - y_0) &= -d \\ at(x_1 - x_0) + bt(y_1 - y_0) + ct(z_1 - y_0) &= -(d + (ax_0 + by_0 + cz_0)) \\ t(a(x_1 - x_0) + b(y_1 - y_0) + c(z_1 - y_0)) &= -(d + (ax_0 + by_0 + cz_0)) \\ t &= \frac{-(d + (ax_0 + by_0 + cz_0))}{(a(x_1 - x_0) + b(y_1 - y_0) + c(z_1 - y_0))} \end{aligned} \quad (2.10)$$

Com o valor de t descoberto (Equação 2.10) basta voltar a equação vetorial e encontrar o ponto de intersecção.

Figura 9 – Representação de intersecção entre vetor e plano



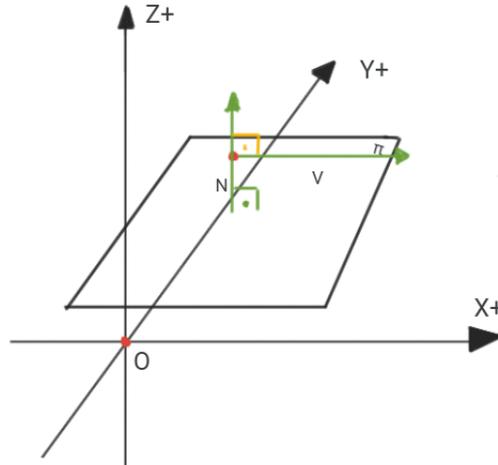
Fonte: Autor

3. Também existem os casos em que a reta não cruza o plano, e para esses casos basta saber se a reta é perpendicular à normal do plano π e $P_0 \notin \pi$. Assim,

² Os vetores, neste trabalho, podem ser entendido como arestas de uma face do objeto.

podemos calcular o produto escalar e caso seja 0, a reta é ortogonal à normal, portanto paralela ao plano e então não o corta (Figura 10).

Figura 10 – Representação de um vetor que não perfura e não está contido ao plano



Fonte: Autor

Com isso, é possível calcular a intersecção entre qualquer segmento de reta e um plano a fim de encontrar os pontos de corte durante o processo de segmentação.

2.3.4 Coordenadas baricêntricas

Para esse trabalho vamos supor que as faces dos objetos são triangulares, então não é só suficiente encontrar um ponto de intersecção entre um vetor e um plano. É necessário saber se esse ponto encontrado está contido na face. Para isso vamos utilizar a definição de coordenadas baricêntricas ao nosso favor (Equação 2.11).

$$P = \frac{uA + vB + wC}{u + v + w} \quad (2.11)$$

De acordo com Figueiredo (2010), coordenadas baricêntricas são escalares que, dado um triângulo ABC e um ponto P , podemos encontrar as coordenadas que descrevem P no plano através de uma média ponderada dos pontos A , B e C .

Ou seja, se dado três escalares u , v e w que podem ser utilizados como pesos para uma média ponderada que nos resulta nas coordenadas que definem o ponto P , os valores u , v e w são considerados coordenadas baricêntricas (Equação 2.12).

É importante comentar que o termo coordenada baricêntrica não representa, nesse caso, coordenadas no plano cartesiano.

Além disso, existe uma relação das coordenadas baricêntricas com a área do triângulo, de modo que as coordenadas baricêntricas que representam o ponto podem ser adquiridas através da área dos subtriângulos considerando o ponto P , ou seja, $S_{ABC} =$

$S_{PBC} + S_{PCA} + S_{PAB}$, logo entendendo que a soma das três áreas nada mais é do que a área do triângulo ABC , podemos definir que:

$$\frac{S_{ABC}}{S_{PBC} + S_{PCA} + S_{PAB}} = 1. \quad (2.12)$$

Então se $S_{PBC} + S_{PCA} + S_{PAB} \neq 1$ o ponto não está contido na face.

2.4 Curvatura média

É fundamental abordar uma característica topológica de grande relevância quando se trata de análise de malhas. Para compreender esse conceito, podemos recorrer a uma analogia.

Ao inserir um lápis em um ponto da superfície de uma bola de futebol, o lápis ficará perpendicular à superfície, ou seja, seguirá a direção da normal. No entanto, em torno desse ponto, ocorrem várias curvaturas diferentes.

A curvatura média, como próprio nome diz, corresponde a médias dessas curvaturas. Essa métrica desempenha um papel importante, pois auxilia na compreensão das características das superfícies. Indica se uma superfície é mais convexa, mais côncava ou plana.

Em casos discretos, como malhas triangulares, podemos usar aproximações já conforme demonstrado em estudos anteriores (GATZKE; GRIMM, 2006; RUSINKIEWICZ, 2004).

Identificar essa característica é crucial durante o processo de particionamento, pois, após a impressão, pode demandar um esforço manual significativo para unir segmentos altamente curvados.

2.5 Abordagens inteligentes

Para solucionar problemas para os quais a sua resolução não é tão intuitiva e não possuem uma solução direta, é comum a tentativa de utilizar algoritmos de aprendizado de máquina. Por exemplo, o reconhecimento de faces, reconhecimento de padrões, e em problemas de segmentação semântica de objetos (GARCIA-GARCIA et al., 2017).

Considerando a complexidade inerente ao problema que o presente estudo busca abordar, foi decidido utilizar Algoritmos Genéticos (AG), devido à sua abordagem altamente robusta, capaz de oferecer interpretabilidade tanto dos resultados obtidos quanto do processo utilizado para alcançá-los.

Ao contrário das Redes Neurais Recorrentes (RNN), conhecidas por sua natureza caixa-preta no que diz respeito à compreensão dos motivos que levam à obtenção de determinados resultados.

Quando dizemos que uma RNN tem uma natureza caixa-preta, queremos dizer que é difícil compreender completamente como a rede neural chega a certos resultados ou

tomadas de decisão. Isso ocorre porque as RNNs têm uma estrutura de *loops* que lhes permite manter e atualizar estados ocultos ao longo do tempo. Esses estados ocultos são fundamentais para a capacidade da rede de aprender dependências de longo prazo em sequências.

2.5.1 Algoritmo Genético

AGs possuem seu fundamento na teoria de seleção natural de Charles Darwin (SIVANANDAM; DEEPA, 2008). Os organismos biológicos mais aptos a determinadas circunstâncias são aqueles que sobrevivem e proliferam em maior quantidade.

Inúmeros trabalhos já demonstraram sua eficiência como Cagnoni et al. (1999) que utiliza AGs para segmentar imagens médicas tridimensionais. AGs são fortes candidatos para entrar neste trabalho durante processo de identificar quais são os melhores valores para os principais parâmetros que deverão ser considerados durante o processo de segmentação.

A proposta é utilizar os algoritmos genéticos para minimizar uma equação, visando gerar uma configuração com a menor quantidade de segmentos com as partes mais simples para impressão.

AGs são técnicas de meta-heurística com base no princípio da evolução natural para encontrar soluções para problemas complexos. Seu funcionamento consiste na geração de uma população inicial de soluções potenciais (chamadas cromossomos ou indivíduos) e, em seguida, na utilização de operações de reprodução, mutação e seleção para gerar novas gerações de soluções.

Os algoritmos genéticos são utilizados para resolver problemas de otimização, como, por exemplo, encontrar a melhor rota em um problema do caixeiro-viajante ou encontrar a melhor configuração de parâmetros em um problema de aprendizado de máquina.

A meta-heurística é um conjunto de técnicas utilizadas para encontrar boas soluções para problemas complexos, mas não garantem que sejam a melhor solução. Os algoritmos genéticos são uma das técnicas de meta-heurística mais populares e eficazes (BLUM; ROLI, 2003).

2.6 Complexidade

Podemos dizer, de forma leiga, que a complexidade é o tempo que um algoritmo leva para ser executado em função do tamanho da instância de entrada (FEOFILOFF, 2021a).

Dentre as classificações, podemos definir como P (ou problemas de decisão polinomiais) aqueles problemas que possuem uma solução que execute em tempo polinomial e resolva todas as suas instâncias. Exemplos de problemas que pertencem à classe P

incluem verificar se um grafo é conectado ou se um número é primo.

Outra categoria é a dos problemas NP (ou problemas de decisão não-deterministas polinomiais). Esses problemas são aqueles que são polinomialmente verificáveis. Isso significa que, dado uma solução da instância do problema, existe um algoritmo que execute em tempo polinomial e que diga, com veracidade, se a solução gerada é ou não uma solução para tal.

A diferença entre as classes P e NP é que problemas na classe P podem ser resolvidos de forma eficiente, enquanto problemas na classe NP podem ser verificados de forma eficiente, mas a solução pode ser computacionalmente cara. A questão se todo problema NP está em P é conhecida como P versus NP ainda não foi provado se é verdadeiro ou falso.

Além disso, existe uma classe conhecida como NP-difícil que para um problema X ser considerado NP-difícil é necessário que todos os problemas da classe NP sejam polinomialmente redutíveis a X Feofloff (2021b). Problemas NP-difíceis são, ao menos, tão difíceis quanto os problemas mais difíceis dentro de NP, mas não estão necessariamente dentro de NP. Se estiverem também dentro de NP (demonstrado por reduções em tempo polinomial entre o problema em questão e um problema conhecido de NP), eles são NP-completos. (SIPSER, 1996).

Exemplos de problemas NP-difíceis incluem o problema da Mochila (*Knapsack Problem*) que é um problema NP-Completo que foi demonstrado que é tão difícil quanto os problemas mais difíceis em NP. Sua definição é da seguinte maneira: Dado um conjunto de itens, cada um com um valor e um peso, determinar a combinação de itens que maximize o valor total, sujeito a uma restrição de peso (CORMEN, 2012). Além disso, o problema de cobertura de conjuntos. Que nada mais é do que um problema de otimização que pode ser usado em diversos problemas de alocação de recursos.

Uma instância (A, B) do problema consiste em um conjunto finito A e uma família B de subconjuntos de A , tal que todo elemento de A pertence a, no mínimo, um subconjunto em B . Então problema é encontrar um subconjunto de tamanho mínimo $\subseteq B$ cujos membros cubram todo o A (CORMEN, 2012).

O motivo destes problemas serem considerados NP-difíceis, é por sua solução exigir uma busca em espaços muito grandes de possíveis soluções, o que torna computacionalmente penoso.

Problemas NP-completo são tão difíceis quanto problemas NP, e restringem-se a problemas de decisão, ou seja, aqueles que sua resposta é ‘sim’ ou ‘não’. Estes possuem uma propriedade onde caso exista uma solução polinomial para um problema NP-completo, logo todo problema NP também deverá possuir, visto que NP-completo pertence a classe dos problemas NP (CORMEN, 2012).

NP-difícil é uma classe de problemas importantes, pois muitos problemas reais são NP-difíceis e encontrar soluções eficientes para eles tem sido um desafio para a comunidade

científica. Algoritmos de meta-heurística como algoritmos genéticos são frequentemente usados para tentar resolver esses problemas, mas ainda não existe uma solução geral.

Para se tornar um bom projetista de algoritmos, você deve entender os rudimentos da teoria da NP-completude. Se puder determinar que um problema é NP-completo, estará dando uma boa evidência de sua intratabilidade. Então, como engenheiro, seria melhor empregar seu tempo no desenvolvimento de um algoritmo de aproximação [...] ou resolvendo um caso especial tratável, em vez de procurar um algoritmo rápido que resolva o problema exatamente. (CORMEN, 2012, 5.)

Antes de classificar o grau de complexidade do nosso problema, devemos entender do que ele se trata. Abstraindo-o, nada mais temos que uma nuvem de vértices no espaço e queremos descobrir suas categorias, que tratam-se de um grupos de pontos que definem um segmento do objeto, para assim agrupá-los.

Não obstante, é idealizado uma caixa que divide cada categoria sem possuir pontos de outra. Problemas assim já são estudados no campo bidimensional, conhecido como The Boxes Class Cover problem (BCC-problem). Como demonstrado em Bereg et al. (2012) fazendo uma redução a outro problema NP-completo, Rectilinear Polygon Covering problem (RPC-problem).

Problemas dessa categoria são considerados NP-completos. Empiricamente podemos observar similaridades entre os problemas. É muito provável que o problema seja tão desafiador quanto um problema reconhecidamente NP-completo e, devido às limitações de escopo e tempo do TCC, pode não ser possível demonstrar formalmente essa complexidade. No entanto, vamos assumir essa complexidade com todos os riscos associados a essa suposição, incluindo a possibilidade de um equívoco.

Antes de classificar o grau de complexidade do nosso problema, devemos entender do que ele se trata. Abstraindo-o, nada mais temos que uma nuvem de vértices no espaço e queremos descobrir suas categorias, que tratam-se de um grupos de pontos que definem um segmento do objeto.

Não obstante, é idealizado uma caixa que divide cada categoria sem possuir pontos de outra. Problemas assim já são estudados no campo bidimensional, conhecido como The Boxes Class Cover problem BCC-problem.

Trata-se de um problema de otimização combinatória que consiste em encontrar o menor número possível de conjuntos (classes) de caixas que podem ser usadas para cobrir todas as caixas de entrada, de tal forma que nenhuma caixa seja coberta por mais de uma classe. Cada classe de caixas é definida como um subconjunto de caixas, onde cada caixa pode pertencer a várias classes (GAREY; JOHNSON, 1982).

Como demonstrado em (BEREG et al., 2012) a partir de um problema considerado NP-completo, Rectilinear Polygon Covering problem (RPCproblem), foi reduzido em tempo polinomial ao BCC-problem assim comprovando sua NP-completude. Empiri-

camente podemos observar similaridades entre os problemas, e não é difícil concluir que se trata de um problema tão difícil quanto ao seu familiar bidimensional.

3 Trabalhos Relacionados

Essa seção visa apresentar os trabalhos relacionados ao tema. Além disso, será demonstrado qual é o estado da arte a fim de compreender quais suas limitações, para então com isso entender o contribuição desse trabalho para a academia. Por existirem poucos trabalhos direcionados para segmentação de objetos para impressão 3D, além de artigos que visam apresentar soluções para limitação de volume de impressão, foi optado por uma revisão literária integrativa com intuito de analisar algumas técnicas relacionadas ao trabalho que podem contribuir com a solução do problema.

3.1 Protocolo de pesquisa

A busca dos trabalhos relacionados foi dividida em duas etapas:

1. Coleta de artigos: Foi selecionado dois dos principais motores de busca, Google Scholar e IEEE Xplore. Foram utilizados frases de consulta visando encontrar a maior quantidade de material relacionado, frases tais como:
 - Em português: (Manufatura Aditiva OR impressão 3D) AND (Segmentação OR Limitação volumétrica).
 - Em inglês: (Additive Manufacturing OR 3D Printing) AND (Segmentation OR Volume Limitation).
2. Análise dos trabalhos encontrados

Durante essa etapa, foram definidos critérios de exclusão relacionados aos temas que os artigos devem ter.

 - Limitação volumétrica de impressão 3D.
 - Segmentação de objetos 3D.
 - Segmentação de manufatura aditiva.
 - Problemas volumétricos no processo de manufatura aditiva
 - Processo de segmentação de objetos em impressão 3D

Após a análise, os artigos foram divididos em categorias. Além dos artigos relacionados à solução de segmentação, também foram coletados três artigos, um relacionado a representação de objetos 3D com octree e outros 2 sobre a utilização de componentes principais. Visto que são abordagens que podem ser utilizadas durante a resolução desse trabalho.

3.2 Segmentação semântica

A segmentação semântica consiste em dividir um conjunto de *pixels* de acordo com o contexto de suas informações, afim de predizer suas categorias semânticas com base em uma etiqueta definida, ferramenta utilizada para, por exemplo, identificar em

uma imagem os móveis de um cômodo. Atualmente essa área vem se expandindo para o contexto de objetos 3D, onde o processamento que antes era em *pixels* agora está na nuvem de pontos. Esses estudos são fundamentais quando trabalhamos com cenas compostas por diversos objetos unidos em um único arquivo, visto que através das abordagens existentes conseguimos categorizar esses agrupamentos de pontos e destacar da cena cada objeto.

He et al. (2021) faz um levantamento de 150 artigos sobre o modelo de segmentação semântica baseado em aprendizado profundo, bem como um breve comparativo entre eles. Além disso, são demonstradas aplicações da segmentação 3D no contexto de vídeos 3D fornecidos por automóveis autônomos e também para diagnósticos médicos. Ao final os autores concluíram que, dentre as abordagens analisadas, destacaram-se redes baseadas em convolução de pontos visto que levam cada dado em consideração quando estão sendo treinadas. Entretanto, essa abordagem ainda está em estudo e pode ser muito custosa.

3.3 Octree

Podemos optar por utilizar uma malha *voxel* com as dimensões de impressão entretanto essa abordagem pode ser muito custosa, visto que em representações 3D o custo computacional cresce cubicamente, o que pode ser contornado de algumas maneiras, onde abordagens utilizando *octree* irão nos auxiliar em modelos de alta resolução.

Em Tang (1992), os autores tem como objetivo explicar como um objeto 3D pode ser representado através de uma *octree*, bem como quais são seus pontos negativos e positivos. Além disso, são demonstrados diferentes tipos de implementação da abordagem, como *octree* pura, abordagem tradicional que segmenta cada parte que contiver pedaços do objeto dentro de seu volume em octantes de cubos. Outra abordagem exemplificada no artigo, denominada *polytree*, que ao contrário da anterior, sua subdivisão avança até encontrar uma das fronteiras do objeto, armazenando informações sobre o limite dentro do nó da árvore. Também apresenta técnicas de manipulação dessa estrutura, como algoritmos de divisão em conquista, operações de transformação de coordenadas, dentre outras. Além disso, demonstra aplicações dessa técnica em sistemas de modelagem de sólidos, inclusive no processo de otimização dos cálculos em *ray tracing* de modo que é usado para identificar quando um objeto é efetivamente atingido pelos raios, levando em consideração apenas octantes que a luz atravessa.

3.4 Análise de Componente Principal

A Principal Component Analysis (PCA) é fundamental para resolução do nosso problema, visto que a rotação do nosso objeto impacta diretamente em como uma possível malha de *voxel* irá ser construída. Então, rotacionar o objeto de modo que minimize a quantidade de cubos gerados, de mesmo modo, utilizar PCA especificamente nos segmentos

encontrados a fim de encontrar partes que podem ser destacadas do objeto deixam a solução mais eficiente e robusta.

No artigo de Xiao et al. (2019), os autores propõem uma técnica para encontrar a posição que, dado um objeto com diferentes rotações, as coordenadas expressas não sejam alteradas através de uma rotação, ou seja, utilizando PCA são identificadas as coordenadas principais e então através das coordenadas transformadas é reposicionado o objeto. Além disso, é demonstrada a sua utilidade proporcionando o aumento na eficácia, se acoplado a modelos de classificação para reconhecimento de objetos 3D. Além disso, também é provado o aumento na *performance* de algoritmos de reconstrução de objetos 3D quando é utilizada sua técnica em conjunto. Encontrar o posicionamento ideal pode ser fundamental se optarmos por utilizar abordagem de *voxel*, visto que a angulação do objeto pode gerar mais cubos em um determinado eixo e isso deve ser considerado durante o processo de aprendizado já que temos como objetivo a minimização de segmentos.

Já no artigo Sfikas, Theoharis e Pratikakis (2011), os autores apresentam um processo de automatização para normalizar a pose de um objeto 3D utilizando PCA para a normalização de alinhamento em conjunto com Surface-Oriented Minimum Bounding Boxes (SoMBBs), ou seja, uma caixa que serve como delimitador. Os autores definem o conceito de normalizar pose, como o procedimento de normalização de rotação, translação e escala, ou seja encontrar os melhores valores para essas transformações que permita o encaixe em um SoMBBs levando em consideração o volume mínimo e diferença angular mínima entre o normais das faces do SoMBBs e as normais da face do objeto 3D. Assim quando é dito normalizar alinhamento, refere-se apenas a normalização de rotação. De mesmo modo que o artigo anterior, os autores demonstram a eficiência da sua abordagem através de uma comparação com outras metodologias do estado da arte, ao final conclui-se que o ganho de desempenho é de 3% no sistema final.

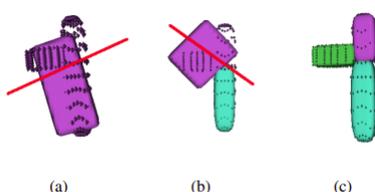
3.5 Segmentação de objetos 3D

Aqui encontram-se artigos que demonstram alguma técnica interessante a se considerar ou abordagens do estado da arte. Trabalhos que de alguma forma mostraram-se promissores na segmentação de objetos.

O trabalho de Chevalier, Jaillet e Baskurt (2003) apresenta uma nova abordagem fundamentada nos conceitos de super quádrlica e super elipsóide (STUDOCU, 2017). É demonstrada uma alternativa ao trabalho de Leonardis (LEONARDIS; JAKLIC; SOLINA, 1997), o qual propõe um mecanismo de segmentação de nuvem de pontos usando sementes, ou seja pontos distribuídos de maneira arbitrária pelo objeto. Essas sementes, a cada passo, crescem em formato elipsóide, chegando ao estado de desenvolvida quando não consegue se expandir sem perder essa propriedade. Também foi pensado em um mecanismo de absorção de sementes que possam ser mescladas sem perder a propriedade de elipsóide. Entretanto, como demonstrado no artigo, isso tem diversos problemas, dentre

eles, não há uma maneira automática de saber se os pontos vão estar em cada segmento de objeto da nuvem de pontos. Para resolver esse e outros problemas que a abordagem de Leonardis possui, os autores desenvolveram um algoritmo de divisão e conquista. Ao contrário da anterior, os autores começam com um conjunto da nuvem de pontos que pode ser representado como um elipsóide, e através de um parâmetro de distorção e um limiar o algoritmo decide quando será dividido por um plano ortogonal ao eixo de inércia (Figura 11).

Figura 11 – Processo de divisão



Fonte: Chevalier, Jaillet e Baskurt (2003, p. 8)

Logo em seguida, é realizado o processo de mesclagem dos elipsóides adquiridos com objetivo de minimizar a distorção. Ao final, os autores conseguiram resultados excelentes em nuvem de pontos esparsos. Esse artigo é fundamental para nosso trabalho, visto que ao realizar esses procedimentos teremos formas geométricas mais simples o que poderá simplificar o processo de segmentação para impressão 3D.

No trabalho Konig et al. (2022), é apresentada uma solução genérica para resolver uma limitação física que está no volume de impressão em MAs. O artigo apresenta um algoritmo atrelado ao software CATIA para criar zonas de pinagem e abas para colagem em fatias de objetos segmentados. Dá-se início ao processo com a seleção manual da posição inicial dos planos de corte, seus ângulos são calculados automaticamente, visando deixar a base ao qual teve o corte em paralelo à base de impressão. Unindo três técnicas de junção, encaixe geométrico, encaixe por pressão e junção com colagem é realizada a *Hybrid Joint Design*, uma junção híbrida que é implementada pelos autores durante o processo de prototipagem da peça. O algoritmo, através de características da impressão manufaturada e conceitos físicos como distribuição de força e estabilidade da peça, consegue detectar locais das áreas segmentadas que devem receber determinado tipo de junção e inserir abas, pinos e encaixes automaticamente, o que é interessante para nosso trabalho, visto que a impressão de objetos segmentados tem como objetivo a junção posteriormente.

Benes et al. (2014) demonstra o estado da arte de segmentação de objetos 3D para MAs. O artigo propõe um algoritmo para economizar o tempo de impressão, a utilização de material bem como uma solução para impressão de conchas de objetos maiores que a área de impressão. Os autores realizam um processo de esvaziamento da peça para

logo efetuar a segmentação da casca gerada, assim agrupando as peças com base em uma função custo que calcula a eficiência com que elas podem se juntar em um volume menor. Ou seja, o resultado final consiste em um grupo de peças estrategicamente posicionadas dentro de um volume definido, otimizando o material e o tempo de impressão. O trabalho é separado em 4 etapas:

- Pré-processamento: onde será recebido uma nuvem de pontos que é convertida uma casca fina utilizando *voxels* de mesmo tamanho por fim é gerado uma malha tetraédrica.
- Em seguida, temos a etapa de segmentação, que recebe essa malha tetraédrica como entrada e leva em consideração algumas condições que tornam o processo de montagem da peça após a impressão mais simples.
- A seguir é realizado o processo de sementeação e expansão, ou seja, são escolhidos pontos de tal modo que cada grupo inicial ocupe 1% do tamanho total da malha e isso é feito até que não seja possível distribuir igualmente as sementes iniciais. O processo de expansão utiliza o algoritmo de *k-means* (SANTOS, 2019) para encontrar tetraedros vizinhos a fim de expandir sua área de ocupação igualmente. Ao final, o algoritmo retorna um conjunto de segmentos $S = \{S_1, S_2, \dots, S_n\}$. O próximo passo consiste em *pré-merging*, onde são resolvidos os problemas de segmentos pequenos e vizinhos com áreas transversais difíceis de se conectar após a impressão.
- Por fim, é realizado o processo de empacotamento e organização das peças, o qual tem como objetivo encontrar a melhor maneira de posicionar as peças dentro do limite de impressão ocupando o menor volume possível.

Ao final os autores conseguiram um resultado satisfatório para a segmentação de objetos, entretanto não foi planejado nenhum mecanismo para junção dos segmentos além do tradicional mecanismo de junção por colagem.

3.6 Diagramas de Voronoi 3D

Os diagramas de Voronoi tridimensionais são métodos muito eficientes para descrever distribuições espaciais usando a relação entre vizinhança de pontos como demonstrado em Borovkov e Odell (2007). Então, foi realizado um levantamento de trabalhos que utilizam essa técnica para segmentação de objetos 3D com objetivo de investigar a contribuição que essa abordagem pode proporcionar ao trabalho.

Ying et al. (2015) apresenta um método de segmentação utilizando diagrama de Voronoi tridimensional. Utilizando os atributos que tal abordagem proporciona, como a área da superfície da célula Voronoi, seu volume, quantidade de vizinhos diretos, dentre outras, em conjunto com a distância euclidiana, os autores conseguem calcular diferentes

valores de importância da célula em seu contexto. Além disso, é demonstrado um estudo em uma nuvem de pontos aleatória sobre a influência que diferentes valores de importâncias causam no agrupamento dos pontos. Após a conclusão da importância de considerar a relação de vizinhança entre células Voronoi, utilizando um cluster hierárquico o trabalho apresenta a utilização desses valores para segmentar um modelo 3D de um cavalo com diferentes quantidades de *clusters* produzindo resultados satisfatórios.

No trabalho Lu, Lévy e Wang (2012), os autores desenvolvem uma solução para esqueletização de objetos 3D. Entretanto, como comentado no artigo, apesar de não ser o objetivo, a abordagem contribui com a segmentação de objetos. Ao contrário do artigo anterior, os autores optaram por utilizar a tesselação centroidal de Voronoi, um tipo especial de tesselação Voronoi onde cada ponto gerador de uma célula Voronoi é também seu centro de massa. Os autores utilizam o esqueleto adquirido para mover as bordas de *clusters* de células de Voronoi a fim coincidir com as intersecções das retas que representam as juntas do esqueleto, de forma que os segmentos tornam-se visualmente mais precisos quando se fala de segmentação contextual.

4 Desenvolvimento

Durante essa seção serão apresentados os passos realizados para elaboração desse trabalho bem como, breves resultados a fim de elucidar o progresso durante o tempo.

4.1 Definição de ferramentas

Durante a análise dos trabalhos relacionados, bem como pesquisas paralelas, foi definido que o trabalho será desenvolvido em C++, por tratar de uma linguagem de médio nível, orientada a objetos. Além disso, muitos fatiadores, até mesmo CAD são desenvolvidos em C ou C++ o que pode facilitar futuramente uma integração, em adendo a esse fato, seu desempenho pode facilitar os testes.

Ademais, foi definido a utilização da Application Programming Interface (API) OpenGL (KHRONOS, 1992) e o *toolkit* Freeglut (OLSZTA, 2005) por se tratar de uma ferramenta robusta de computação gráfica com funcionalidades que podem facilitar o desenvolvimento do trabalho, como rotações, translações e dentre outras.

4.2 Abordagem inicial

Para resolução do trabalho, as tarefas foram divididas em duas categorias que a ferramenta deve fazer. Etapas não relacionadas a abordagem de aprendizado e relacionadas.

Dentre as etapas que não são relacionadas ao método de aprendizado podemos listar as seguintes:

- Desenvolver um leitor de objetos 3D
- Desenvolver uma ferramenta de visualização de objetos
- Desenvolver métodos para encontrar os pontos de interseções entre objetos.
- Ajuste de imperfeições em objetos.
- Segmentação de objetos

Já para os processos que estão relacionados ao aprendizado de máquina, podem ser listados desse modo:

- Função custo.
- Implementação da abordagem de aprendizado.

4.2.1 Funcionalidades gerais

Durante essa subseção será demonstrado todo o processo que foi necessário para concluir as etapas que não estão relacionadas ao método de aprendizagem.

4.2.1.1 Leitor de arquivos .obj

O primeiro passo que a ferramenta precisa realizar, que independe do método de segmentação, consiste no leitor de objetos 3D. Inicialmente a ferramenta irá dar suporte a arquivos com extensão .obj¹. Entretanto, para nossa tarefa não temos interesse em uma série de informações que são encontradas nos arquivos com tal formato.

A maneira de resolução desse problema consiste na construção de alguns métodos para interpretar o padrão de um .obj ignorando informações desnecessárias tais como comentários, normais, texturas, etc. O ponto chave para a solução está na utilização das expressões regulares que identificam somente as faces e os vértices.

Expressões utilizadas:

1. Regex utilizada para selecionar os vértices:

“(v)\s+(-?d+(?:\.\d+?)\s+(-?\d+(?:\.\d+?)\s+(-?\d+(?:\.\d+?)”

2. Regex utilizada para identificar uma linha com face:

“^f.* ”

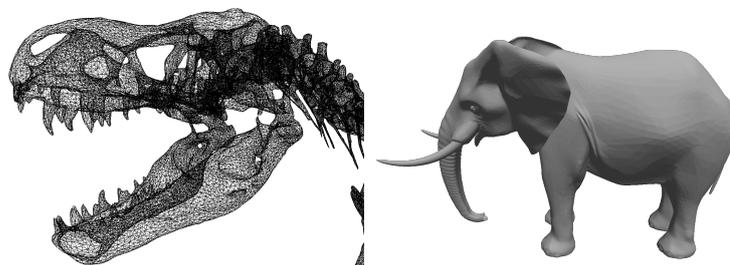
3. Regex utilizada para identificar os índices dos vértices na linha de uma face:

” [0-9]+”

4.2.1.2 Visualizador 3D

Para visualizar o objeto, foi necessário desenvolvermos um mecanismo responsável por realizar os controles de interface utilizando a API OpenGL. Assim, com ela é possível movimentar a câmera no espaço e visualizar o objeto em diversos ângulos.

Figura 12 – Demonstração do visualizador 3D

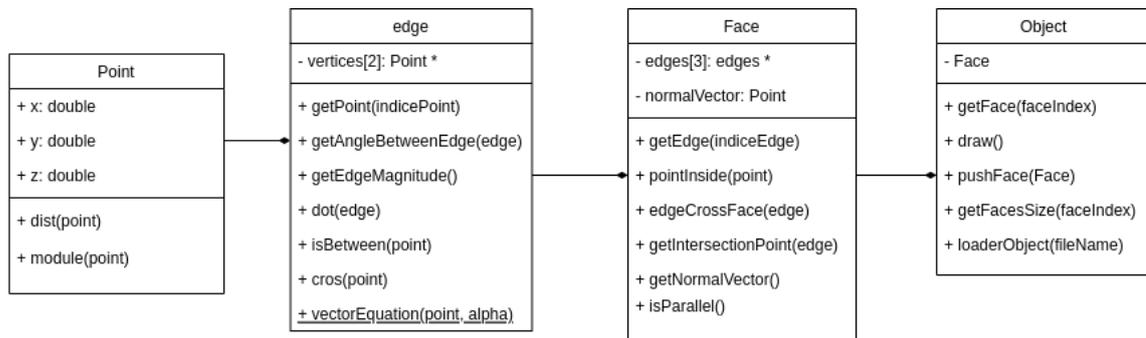


Fonte: Autor

Ambas ferramentas tiveram um planejamento prévio sobre quais seriam as funções necessárias e como deveriam estar organizadas como demonstra o seus diagramas de classe (Figuras 13 e 14).

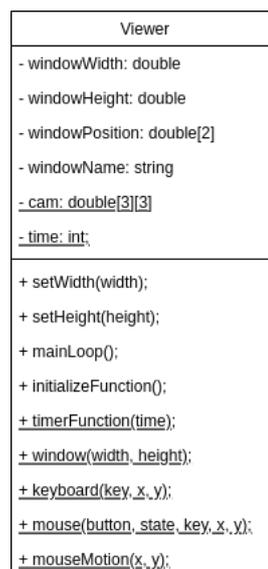
¹ Consultar seção 2.1.1

Figura 13 – Diagramas de classe UML - Leitor de Objetos 3D



Fonte: Autor

Figura 14 – Diagramas de classe - visualizador 3D



Fonte: Autor

4.2.1.3 Interseções

Além do visualizador, precisamos explicar como foi realizado o métodos para encontrar interseções entre objetos. Isso é realizado através dos conhecimentos passados durante a seção 2.3. Para fins didáticos, vamos supor um exemplo:

Dados 3 vértices A, B, C pertencentes a uma face do objeto podemos então deduzir que:

$$AB = (B_x - A_x, B_y - A_y, B_z - A_z)$$

$$AC = (C_x - A_x, C_y - A_y, C_z - A_z)$$

Através das eq: 2.5 e 2.6 de $\vec{AB} \times \vec{AC}$ conseguimos chegar no vetor normal e na equação do plano.

$$\begin{aligned} & \hat{i}(AB_y AC_z) - \hat{i}(AC_y AB_z)y + \\ & \hat{j}(AB_z AC_x) - \hat{j}(AC_z AB_x)y + \\ & \hat{k}(AB_x AC_y) - \hat{k}(AC_x AB_y)z + d = 0 \end{aligned} \quad (4.1)$$

Com o conhecimento que o produto escalar entre vetores ortogonais é 0, então se um \vec{AB} que pertence ao plano π e o vetor normal $n = a\hat{i} + b\hat{j} + c\hat{k}$ também é ortogonal a ele, logo:

$$\begin{aligned} & (AB_x, AB_y, AB_z)n = 0 \\ & aAB_x + bAB_y + cAB_z = 0 \\ & a(B_x - A_x) + b(B_y - A_y) + c(B_z - A_z) = 0 \\ & aB_x - aA_x + bB_y - bA_y + cB_z - cA_z = 0 \\ & aB_x + bB_y + cB_z = aA_x + bA_y + cA_z \end{aligned} \quad (4.2)$$

Ou seja, obtemos a equação geral do plano: $aA_x + bA_y + cA_z = -d$.

Agora, queremos entender se existe uma aresta constituída por dois pontos CeD perfura a face do objeto. Analisando a equação vetorial:

$$\begin{aligned} & C + t_0 CD \\ & (C_x, C_y, C_z) + t_0(CD_x, CD_y, CD_z) \\ & (CD_x t_0 + C_x, CD_y t_0 + C_y, CD_z t_0 + C_z) \end{aligned} \quad (4.3)$$

Nós queremos um t_0 que satisfaça a equação do plano bem como a vetorial, a fim de encontrar o ponto de interseção. Logo podemos jogar esse vetor na equação do plano e isolar o t_0 como demonstra a equação a seguir com t_0 já isolado. Com isso conseguimos descobrir o ponto de intersecção de acordo com (Equação 2.10).

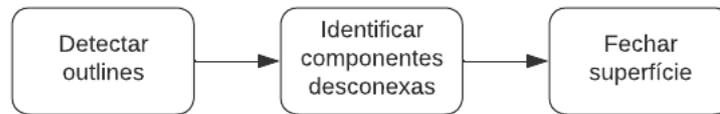
Além disso, é necessário verificar se o ponto descoberto pertence à aresta da face. Para isso, basta verificar se o ponto está entre os extremos da aresta.

Ademais precisamos garantir que a aresta corta a face triangular do objeto, visto que usamos a equação do plano que contém a face. Para isso utilizamos as coordenadas baricêntricas para identificar se o mesmo pertence a face triangular do objeto, como já demonstrado em na seção 2.3.4. Todos esses passos realizados são sequencialmente executados em uma função contida no objeto Face.

4.2.1.4 Ajuste de imperfeições em objetos.

Durante a realização dos experimentos, foi realizado uma série de experimentos com diversos objetos, dentre eles, foi identificado que existem superfícies abertas (ver Figura 16), interferindo diretamente na metodologia utilizada para calcular o volume do objeto (ZHANG; CHEN, 2001). Então para solucionar tal problema, foi realizado uma sequência de passos com objetivo de fechar tais aberturas.

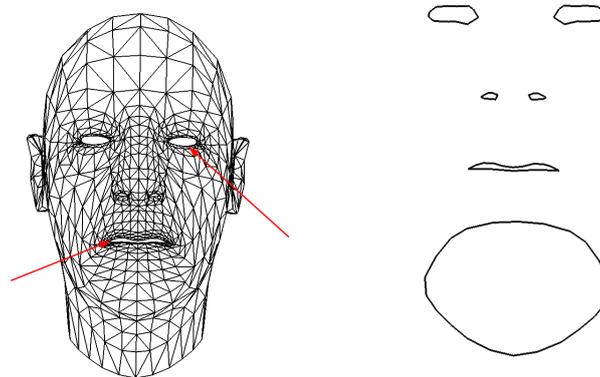
Figura 15 – Pipeline para fechar aberturas



Fonte: Autor

Durante a primeira etapa (veja Figura 15) identificamos todas arestas que só pertencem a uma face. Assim conseguimos saber quais são os pontos que devemos trabalhar. Entretanto, como podemos ver na figura 16 existe arestas que não possuem relação entre si. Para solucionar tal problema foi utilizado um algoritmo de Deep-first-search (DFS) onde através do rastro deixado conseguimos identificar as componentes desconexas.

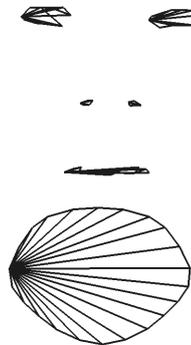
Figura 16 – Contornos do objeto



Fonte: Autor

Identificadas as coordenadas, procede-se à ordenação priorizando x, seguida por y e, por último, z. Em seguida, é construída uma aresta com os dois primeiros pontos. Essa aresta servirá como indicador para a próxima etapa de ordenação, com base no ângulo formado pelas novas arestas criadas a partir do ponto 0 em direção aos demais pontos. Por fim, uma vez que os pontos foram ordenados, é possível montar as novas faces como demonstrado na figura 17.

Figura 17 – Contornos fechados

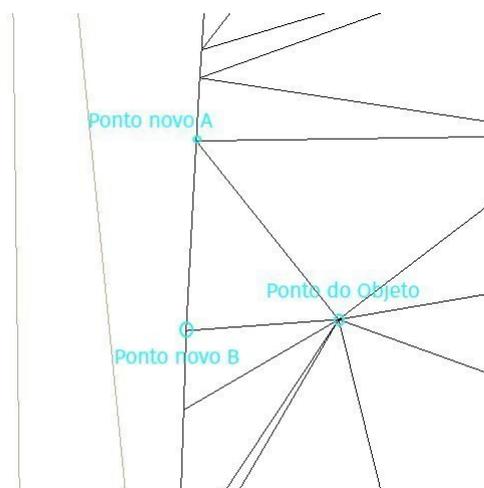


Fonte: Autor

4.2.1.5 Segmentação de objetos

A aplicação realiza a segmentação das faces do objeto que estão em contato com as faces do cubo calculando os pontos de intersecção 4.2.1.3. Os pontos encontrados são ligados aos pontos de objeto completamente contido no cubo, de modo a gerar faces triangulares como podemos ver na imagem figura 18. As novas faces geradas em conjunto com as outras que estão completamente contidas dentro do cubo formam um segmento que está armazenado dentro do cubo.

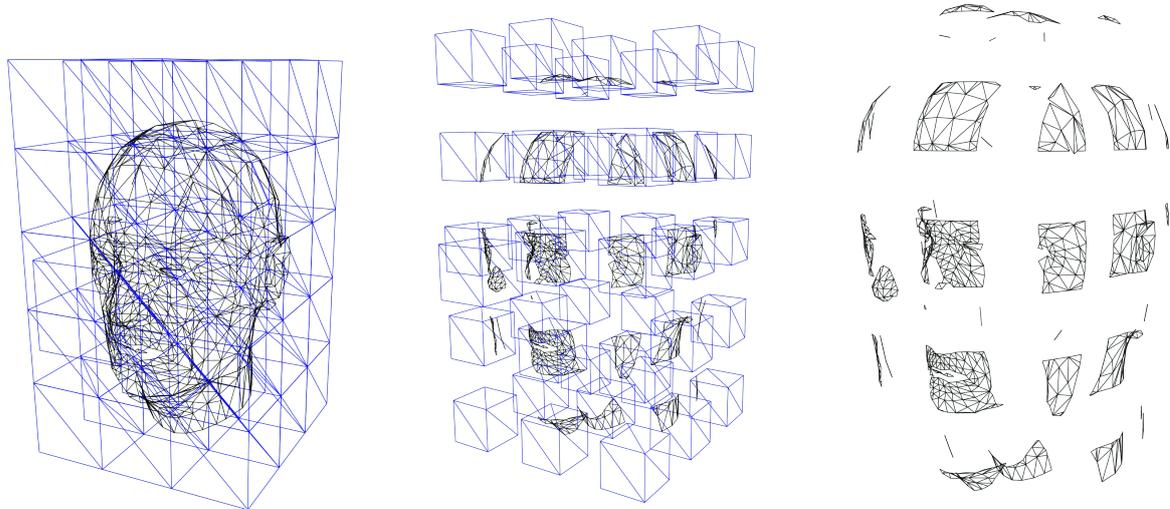
Figura 18 – Criação de uma face através dos pontos obtidos com a segmentação



Fonte: Autor

As dimensões de impressora definem o tamanho de cada cubo. Com isso é criada uma malha de cubos de modo que todo o objeto esteja contido em pelo menos um cubo.

Figura 19 – Exemplo de objeto e segmentação



Fonte: Autor

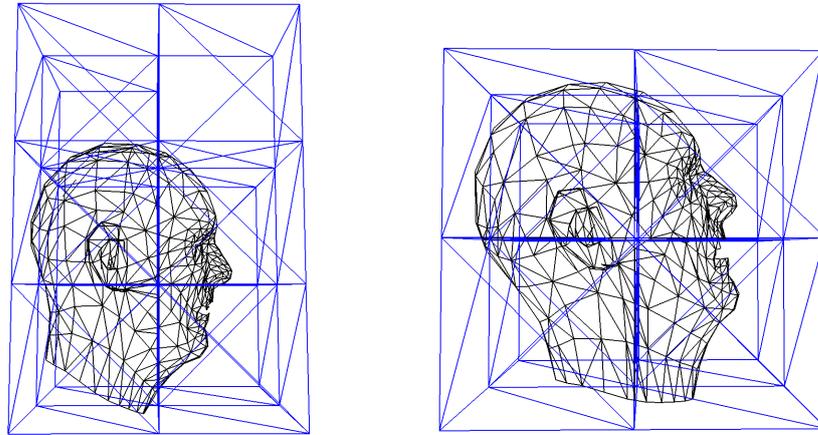
4.3 Função custo

Um dos pontos importantes para o processo de aprendizado é uma função custo que diga qual cubo de impressão é melhor que outro. Entretanto existem diversos parâmetros quando estamos falando de impressão 3D.

Tempo de impressão, volume, quantidade de suporte utilizado, deformações em objetos, ângulo de impressão, dentre outros. Para tal problema, decidimos olhar para pontos principais com objetivo de minimizar a quantidade de segmentos impressos. Visto que outros fatores podem ser minimizados indiretamente.

Por exemplo, maximizando o volume de objeto impresso por impressão, é economizado tempo. Preocupando-se com os ângulos ao qual o objeto está posicionado, faz com que menos ou mais cubos sejam necessários para cobri-lo (veja Figura 20). Pontos como a posição ao qual o *grid* está centralizado também afetam os cortes no objeto.

Figura 20 – Exemplo da implicação do ângulo



Fonte: Autor

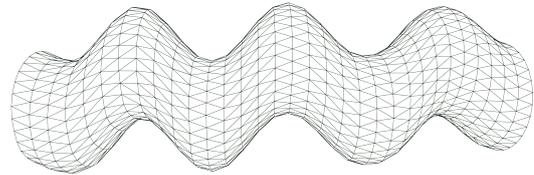
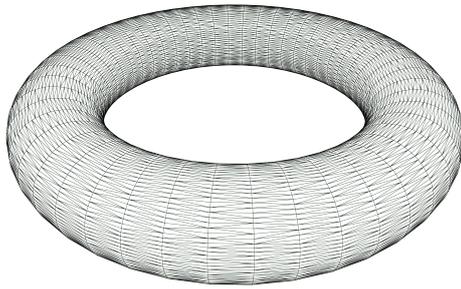
As variáveis relevantes para a função são denotadas como l , h e w , que correspondem ao deslocamento dos seguintes parâmetros em relação ao objeto: comprimento, altura e largura da impressora, respectivamente. Os ângulos ao longo dos eixos x , y e z são representados por θ , ϕ e ρ . A quantidade de cubos é indicada por N_C , enquanto o volume do objeto é definido como V_{obj} . O volume do segmento associado ao cubo i é representado por $V(s_i)$.

Outro fator que está sendo considerado é o índice de curvatura média dos pontos do segmento, mencionado anteriormente na seção 2.4. Essa métrica nos auxilia a quantificar o grau de curvatura de um segmento. Por exemplo, objetos com curvatura acentuada apresentarão valores mais elevados (ver Figura 21). Por outro lado, objetos retos ou com poucas curvas exibirão valores menores (ver Figura. 22).

Ao realizar impressões, buscamos evitar segmentos com alto índice de curvatura média e baixo volume, uma vez que essas características indicam paredes finas suscetíveis a deformações significativas, podendo acarretar problemas pós-impressão.

Portanto, ao formular a equação 4.4, utilizamos o índice de curvatura média como um mecanismo de penalização. Quanto maior for a curvatura e menor for o volume do segmento, maior será a penalidade atribuída.

Figura 21 – Exemplo de objetos curvos.

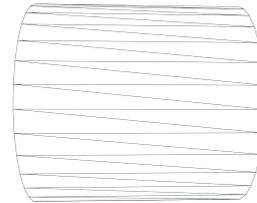
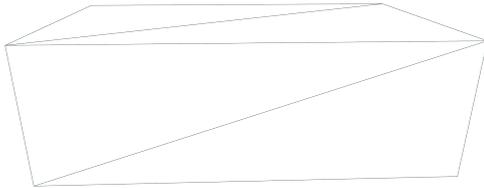


(a) Dimensões do objeto:
 $50.7063\text{cm} \times 9.62936\text{cm} \times 50.7063.$
 $\kappa(s_i):$
 26746.3

(b) Dimensões do objeto:
 $82.0971\text{cm} \times 30.5964\text{cm} \times 20.2658.$
 $\kappa(s_i):$
 6613.7

Fonte: Autor

Figura 22 – Exemplo de objetos com menos curvaturas.



(a) Dimensões do objeto:
 $52.5337\text{cm} \times 31.0221\text{cm} \times 92.7009.$
 $\kappa(s_i):$
 49.4462

(b) Dimensões do objeto:
 $20\text{cm} \times 20.2722\text{cm} \times 20.2722.$
 $\kappa(s_i):$
 383.27

Fonte: Autor

Além disso, nosso objetivo é maximizar a ocupação volumétrica, minimizando a quantidade de cubos utilizados. Para alcançar esse objetivo, optamos por considerar a razão entre a quantidade de cubos e o volume total do objeto.

Isso se deve ao fato de que uma maior quantidade de cubos resulta em uma pontuação inferior, além do fato de que a quantidade de cubos tende a aumentar naturalmente em relação ao espaço ocupado pelo objeto. Portanto, por meio dessa métrica, buscamos alcançar uma quantidade ideal de cubos em relação ao volume, como evidenciado por (Equação. 4.4)

$$\Xi = \min_{l,h,w,\theta,\phi,\rho} (1 - \lambda) \frac{N_C}{V_{obj}} + \lambda \left(\frac{\frac{\sum_{i=1}^{N_C} \kappa(s_i)}{N_C}}{\frac{\sum_{i=1}^{N_C} V(s_i)}{N_C} + \sqrt{\frac{\sum_{i=1}^{N_C} (V(s_i) - \bar{V})^2}{N_C}}} \right) \quad (4.4)$$

Utilizamos um peso λ com valores entre 0 a 1 com objetivo de construir uma

competição entre os termos. Assim, quanto maior for o seu valor, menor será o primeiro termo e maior será o segundo.

Ao optarmos por utilizar abordagem como essa conseguimos uma interpolação entre os valores visando encontrar um valor intermediário.

De mesmo modo que não queremos que exista uma grande quantidade de segmentos, não queremos um alto índice de deformação, então avaliamos apenas o seu valor absoluto.

4.3.1 Implementação da abordagem de aprendizado

Neste capítulo, descreveremos sistematicamente a elaboração do algoritmo genético desenvolvido para este trabalho a linguagem de programação escolhida para a implementação manual do método de aprendizado é o Python, devido à sua ampla disponibilidade de bibliotecas e facilidade de uso.

A escolha pela implementação manual do algoritmo genético foi feita com o objetivo de ter um maior controle sobre os parâmetros e o comportamento dos passos do algoritmo. Dessa forma, podemos ajustar e personalizar cada etapa de acordo com as necessidades específicas do problema abordado na monografia.

A seguir, descreveremos cada etapa do algoritmo genético utilizado:

1. Inicialização da População:

- É criada uma população inicial de indivíduos, onde cada indivíduo representa uma solução possível para o problema. Conjunto de 7 parâmetros.
- Os indivíduos são gerados aleatoriamente, levando em consideração as características e restrições do problema em questão.

2. Avaliação da Aptidão (*Fitness*):

- Cada indivíduo da população é avaliado quanto à sua aptidão, ou seja, o quão bem ele resolve o problema.
- A função de aptidão é definida de acordo com o objetivo do problema. (Veja. 4.4).

3. Seleção dos Pais:

- O método de seleção de pais utilizado é a *roulette wheel* (roleta viciada).
- Cada indivíduo tem uma probabilidade de ser selecionado como pai, proporcional à sua aptidão. Indivíduos com maior aptidão têm maior chance de serem escolhidos.

4. Cruzamento (*Crossover*):

- O método de *crossover* é realizado de forma randomizada, onde os valores dos pais 1 e pais 2 são combinados para gerar um novo filho.

- Os pontos de cruzamento também são escolhidos aleatoriamente, permitindo a combinação de características dos pais em diferentes partes do filho.

5. Mutação:

- A mutação ocorre de forma probabilística, ou seja, com uma determinada taxa de probabilidade.
- Caso ocorra a mutação, uma alteração é realizada no valor de uma característica do indivíduo (Veja seção 4.3), que é randomizada dentro de um intervalo real.

6. Substituição da População:

- Após o cruzamento e a mutação, a nova população é formada pelos filhos gerados.
- A substituição pode ser feita de diferentes maneiras, entretando neste trabalho optamos pela substituição completa da população anterior pelos seus filhos.

7. Critérios de Parada:

- Os critérios de parada são definidos com base no comportamento do algoritmo e nos resultados obtidos.
- Um critério utilizado é a diferença entre a média de Fitness das populações, caso essa diferença seja muito pequena, é um indício de que o algoritmo está convergindo para um consenso.
- Além disso, é definido um limite máximo de gerações, evitando a execução infinita do algoritmo.

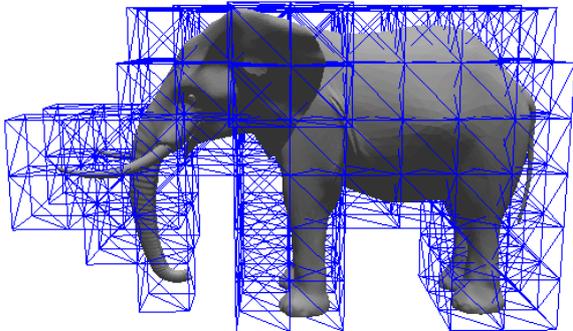
5 Resultados

No decorrer desta seção serão demonstrados os principais resultados atingidos nesse trabalho bem como realizado uma análise dos pontos alcançados e quais são os próximos passos.

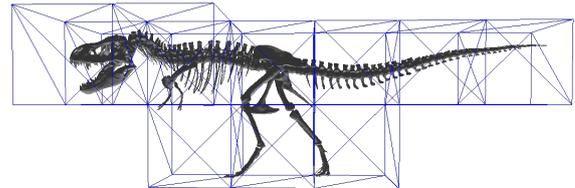
5.1 Equação de pontuação

Para validação da equação foi realizado testes com objetivo de explorar diferentes aspectos. Primeiro para entender se ela reflete o comportamento desejado, foi realizado teste com diferentes dimensões de impressão para diferentes objetos (veja Figura 23).

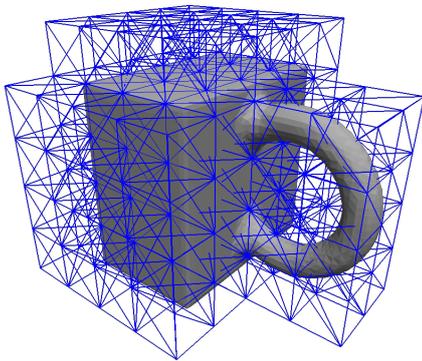
Figura 23 – Exemplo de comportamento da função de pontuaçãoção



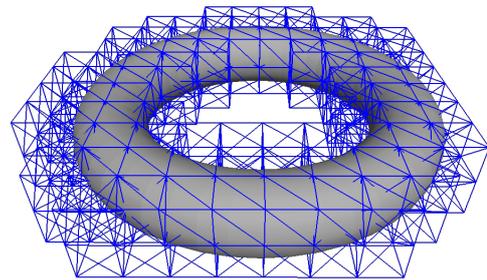
(a) Dimensões do objeto: $226.755\text{cm} \times 297.479\text{cm} \times 466.419\text{cm}$. Dimensões do cubos: $50\text{cm} \times 50\text{cm} \times 50\text{cm}$ (155 cubos). Ξ : 0.259231



(b) Dimensões do objeto: $240.951\text{cm} \times 158.202\text{cm} \times 93.9555\text{cm}$. Dimensões do cubos: $50\text{cm} \times 50\text{cm} \times 50\text{cm}$ (16 cubos). Ξ : 4.95345



(c) Dimensões do objeto: $20\text{cm} \times 20\text{cm} \times 25.9904\text{cm}$. Dimensões do cubos: $5\text{cm} \times 5\text{cm} \times 5\text{cm}$ (104 cubos). Ξ : 4399.03



(d) Dimensões do objeto: $50.7063\text{cm} \times 9.62936\text{cm} \times 50.7063\text{cm}$. Dimensões do cubos: $5\text{cm} \times 5\text{cm} \times 5\text{cm}$ (164 Cubos). Ξ : 4172.28

Fonte: Autor

Note que tanto em 23(a) quanto em 23(b), os objetos possuem grandes dimensões, o que torna natural que as impressoras necessitem de um volume maior de impressão para produzir o objeto. Em ambos os casos, apesar da grande quantidade de cubos, seu custo é baixo. Isso pode ocorrer por alguns motivos:

- O índice de curvatura das partes segmentadas é baixo, e o índice de volume é alto. Isso pode ser observado na Figura 23(a), onde a maior parte do elefante é composta por curvaturas suaves, como o tronco e as patas. Por outro lado, partes como o rabo, a tromba e as presas apresentam um alto índice de curvatura e um baixo volume. No entanto, quando consideramos o elefante como um todo, essas diferenças

se equilibram. Os volumes de impressão que abrangem o corpo do elefante possuem um alto volume, o que reduz a penalização que poderia ocorrer devido às outras partes, justificando, assim, o baixo custo.

- Há um alto índice de curvatura de segmentos com um volume grande, além de poucos volumes de impressão. Ao contrário do primeiro caso, a Figura 23(b) claramente possui um alto índice de curvatura. Entretanto, existem dois fatores que corroboram com o baixo custo da configuração encontrada. O fato de apenas 16 volumes de impressão serem necessários para a produção do objeto já é um fator de minimização. Além disso, essa característica faz com que haja mais volume dos segmentos destacados do objeto por impressão, o que minimiza a penalização. Embora ocorra uma penalização devido ao alto índice de curvatura em cada segmento, essa penalização não é alta.

Conclui-se que, na maioria dos casos, quanto maior for o cubo, menor será o índice de curvatura em relação ao seu volume, o que resulta em um custo menor. Como já comentado na Figura 23(a), temos uma grande quantidade de cubos, entretanto o seu custo é baixo. Isso se deve ao fato de eles possuírem uma quantidade considerável de cubos com baixo índice de curvatura, ou seja, a relação entre as curvaturas dos segmentos e seus respectivos volumes é baixa, resultando em um custo menor.

Por outro lado, na Figura 23(b), temos novamente um objeto grande com alto índice de curvatura. Entretanto, as dimensões de impressão são maiores, o que leva a um volume contido dentro do segmento superior. Logo, a relação entre a taxa de curvatura e o volume torna-se superior, porém equilibrada. Além disso, a relação entre a quantidade de cubos e o volume do objeto também é menor, ajudando a minimizar seu custo.

Agora para os casos 23(c) e 23(d) os custos são muito grandes. Para entender o resultado podemos olhar para os seguintes aspectos.

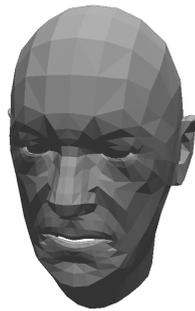
- Há uma grande quantidade de cubos e um baixo volume de impressão. Os segmentos são pequenos e possuem alto índice de curvatura. O principal fator que justifica as Figuras 23(c) e 23(d) possuírem um custo alto é a relação entre o seu volume e a quantidade de cubos necessários para sua produção. Observa-se que, no caso da Figura 23(a), temos uma grande quantidade de volumes de impressão, o que não reflete este caso, já que igualmente temos muitos cubos. Entretanto, o volume do objeto é muito inferior, provocando um custo muito alto para a impressão.

A utilização de cubos menores em uma segmentação que apresente um alto índice de curvatura resulta em um custo superior, devido à maior presença de segmentos pequenos com curvaturas significativas. Ou seja, existem mais partes que possuem um índice de curvatura elevada, porém com baixo volume.

5.2 Aprendizado de máquina

Nesta seção, descreveremos os testes realizados para avaliar o desempenho do algoritmo genético implementado. Para execução do algoritmo foi utilizado o objeto de entrada demonstrado na figura 24, utilizando os seguintes parâmetros:

Figura 24 – Objeto de entrada para o AG. Dimensões: 13.8722cm × 21.2815cm × 15.8325cm



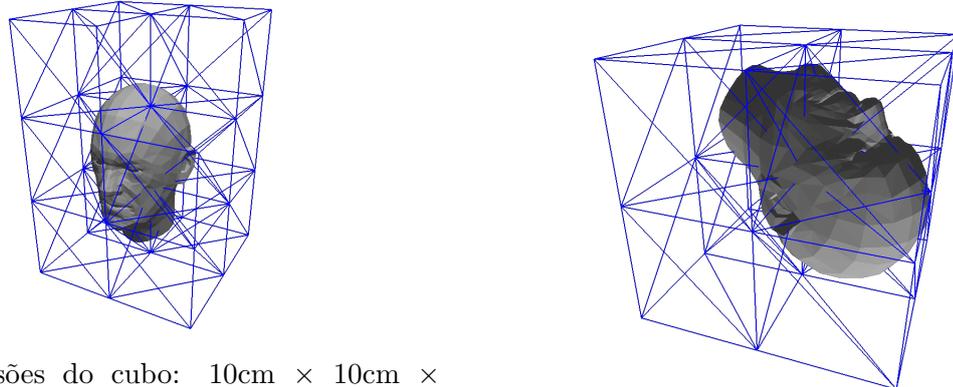
Fonte: Autor

- Número de indivíduos por população: 100
- Número de gerações: 150
- Porcentagem de *crossover*: 60%
- Porcentagem de mutação: 5%
- Intervalo real da mutação: [-100, 100]
- Erro entre gerações: 0.000001

Esses parâmetros foram escolhidos com base em considerações teóricas e experimentais, visando equilibrar a exploração do espaço de busca com a convergência para uma solução satisfatória. De antemão, vale-se comentar que o teste realizado levou aproximadamente 15h de execução. O objetivo do estudo foi avaliar a capacidade do algoritmo genético em encontrar a solução utilizando a eq. 4.4 para um objeto 3D específico. Após a execução dos testes, chegamos a algumas conclusões preliminares:

1. Houve convergência para o objeto 3D em estudo:
 - O algoritmo genético demonstrou ser capaz de convergir para uma solução que atende ao objeto 3D em análise (Veja Fig. 25).
 - Ao longo das gerações, observou-se uma melhoria gradual na aptidão dos indivíduos, indicando que o algoritmo foi capaz de otimizar as características necessárias para alcançar a solução desejada.

Figura 25 – Comparação entre configurações



(a) Dimensões do cubo: $10\text{cm} \times 10\text{cm} \times 10\text{cm}$. Objeto sem configurações - Ξ : 15.2069

(b) Configurações encontradas pelo AG - Cost : 8.8172

Fonte: Autor

2. O resultado é mais satisfatório do que o inicial sem alterar as configurações:
 - Comparando o resultado obtido com o resultado inicial, verificou-se uma melhoria significativa na aptidão dos indivíduos.
 - Isso indica que o algoritmo genético, mesmo utilizando os parâmetros iniciais, foi capaz de encontrar uma solução mais promissora.
3. Não podemos tirar conclusões generalizadas para todos os objetos 3D:
 - É importante ressaltar que os resultados obtidos nos testes não garantem que o algoritmo genético funcionará igualmente bem para todos os objetos 3D.
 - Cada objeto pode apresentar características únicas que exigem ajustes nos parâmetros do algoritmo ou até mesmo uma abordagem diferente.

Dessa forma, embora os testes tenham apresentado resultados promissores e tenham demonstrado a eficácia do algoritmo genético para o objeto 3D em estudo, é necessário realizar estudos adicionais e testes em diferentes objetos antes de tirar conclusões mais amplas. Esses resultados preliminares fornecem uma base sólida para a continuidade da pesquisa e aperfeiçoamento do algoritmo genético no contexto de objetos 3D.

6 Considerações Finais

Após a análise estruturada dos trabalhos relacionados, observamos que há uma lacuna que este trabalho poderá preencher. A presente monografia abordou a aplicação da função custo promissora como uma abordagem satisfatória para a descrição de segmentos com o objetivo de alcançar os objetivos propostos.

Entretanto, durante o desenvolvimento deste trabalho, verificou-se que o algoritmo genético empregado no processo de aprendizado de máquina, juntamente com a função de custo, forneceu resultados promissores, mas demonstrou-se a necessidade de mais estudos. Essas limitações ocorrem devido à complexidade inerente à segmentação de objetos tridimensionais e aos cálculos envolvidos, o que exigiria um estudo mais aprofundado e uma otimização adequada do algoritmo.

Dessa forma, faz-se necessário mencionar os trabalhos futuros que podem ser desenvolvidos com o intuito de superar as limitações encontradas neste estudo. Um aspecto crucial seria a busca por um algoritmo mais eficiente que viabilize um método de aprendizado capaz de lidar de forma adequada com a segmentação de objetos tridimensionais. Tal algoritmo poderia ser explorado em conjunto com uma função de pontuação mais refinada que levem em consideração aspectos como suportes necessários para os segmentos, a fim de alcançar resultados mais consistentes e confiáveis.

Nesse sentido, a realização de pesquisas adicionais e a experimentação de diferentes abordagens no campo do aprendizado de máquina seriam essenciais para avançar no desenvolvimento de um método mais eficaz e eficiente para a segmentação de objetos 3D. Além disso, a exploração de técnicas de otimização e a análise de novos conjuntos de dados poderiam contribuir para aprimorar a qualidade e a precisão dos resultados obtidos.

Em suma, este trabalho apresentou a equação de pontuação promissora como uma abordagem válida e satisfatória para a descrição de segmentos. No entanto, destacou-se a necessidade de futuros trabalhos que busquem aprofundar-se nas características do algoritmo genético utilizado, através do desenvolvimento de um método de aprendizado mais eficiente e de maior convergência. Essas pesquisas futuras são fundamentais para avançar no campo da segmentação de objetos 3D e contribuir para o desenvolvimento de soluções mais eficazes.

Referências

- ASSANTE, D.; CENNAMO, G. M.; PLACIDI, L. 3d printing in education: an european perspective. In: **2020 IEEE Global Engineering Education Conference (EDUCON)**. [S.l.: s.n.], 2020. p. 1133–1138. Citado na página 12.
- BENES, B. et al. Packmerger: A 3d print volume optimizer. **Computer Graphics Forum**, v. 33, 05 2014. Citado na página 35.
- BEREG, S. et al. The class cover problem with boxes. **Computational Geometry**, v. 45, n. 7, p. 294–304, 2012. ISSN 0925-7721. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0925772112000375>>. Citado na página 30.
- BERGELS, D. **Wohlers Report 2022 Finds Strong Industry-Wide Growth**. 2022. Acessado em: 22/10/2022. Disponível em: <<https://wohlersassociates.com/press-releases/wohlers-report-2022-finds-strong-industry-wide-growth/>>. Citado na página 12.
- BEZERRA, I. P. C. e. S. L. H. **Geometria Analítica**. [S.l.]: UFSC/EAD/CED/CFM, 2010. v. 2. Florianópolis. Citado 2 vezes nas páginas 21 e 22.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 35, n. 3, p. 268–308, sep 2003. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/937503.937505>>. Citado na página 28.
- BOROVKOV, K. A.; ODELL, D. A. Simulation studies of some voronoi point processes. **Acta Applicandae Mathematicae**, v. 96, n. 1, p. 87–97, May 2007. ISSN 1572-9036. Disponível em: <<https://doi.org/10.1007/s10440-007-9093-2>>. Citado na página 36.
- BURNS, M. **Importância da prototipagem 3D no desenvolvimento de produtos**. 1989. Acessado em: 22/12/2022. Disponível em: <http://www.fabbers.com/tech/STL_Format>. Citado na página 17.
- CAGNONI, S. et al. **Genetic algorithm-based interactive segmentation of 3D medical images**. [s.n.], 1999. v. 17. 881-895 p. ISSN 0262-8856. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0262885698001668>>. Citado na página 28.
- CANEDO-ARGUELLES, M. D. S. E. L. Estado actual del prototipado rápido y futuro de Éste. **Actas del XI Congreso Internacional de Ingeniería Gráfica**, v. 3, p. 1242–1255, 1999. ISSN 84-699-0475-2. Disponível em: <<https://dialnet.unirioja.es/servlet/libro?codigo=481>>. Citado na página 20.
- CAVALCANTE, L. M. **Matemática Geometria Analítica I**. [S.l.]: EdUECE, 2015. v. 2. Citado 2 vezes nas páginas 21 e 24.
- CHEVALIER, L.; JAILLET, F.; BASKURT, A. Segmentation and superquadric modeling of 3D objects. In: **Journal of Winter School of Computer Graphics, WSCG'03, Pilsen, Rep. Tchèque, 2003**. Plzen, Czech Republic: [s.n.], 2003. p. 00. Disponível em: <<https://hal.archives-ouvertes.fr/hal-01522280>>. Citado 2 vezes nas páginas 34 e 35.

- CORMEN, T. T. **Algoritmos - Teoria e Prática**. [S.l.: s.n.], 2012. v. 3. Citado 2 vezes nas páginas 29 e 30.
- COUTINHO, F. **Object Files (.obj)**. 2020. Acessado em: 23/12/2022. Disponível em: <<http://fegemo.github.io/cefet-cg/attachments/obj-spec.pdf>>. Citado 2 vezes nas páginas 15 e 16.
- FEOFILOFF, P. **Complexidade e problemas NP-completos**. 2021. Acessado em: 03/01/2023. Disponível em: <https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/NPcompleto.html>. Citado na página 28.
- FEOFILOFF, P. **Complexidade e problemas NP-completos**. 2021. Acessado em: 08/01/2023. Disponível em: <www.ime.usp.br/~pf/analise_de_algoritmos/>. Citado na página 29.
- FIGUEIREDO, H. J. B. e José Osório de. **Usando Coordenadas Baricentricas para Estudar a Geometria do Triângulo**. 2010. Acessado em: 29/12/2022. Disponível em: <http://www.mat.ufpb.br/bienalsbm/arquivos/Mini_Cursos_Completos/MC8Completo.pdf>. Citado na página 26.
- FILAMENT2PRINT. **Que impressora 3D comprar? Conselhos dos experientes**. 2018. Acessado em: 30/12/2022. Disponível em: <https://filament2print.com/pt/blog/35_que-impressora-3d-comprar.html>. Citado na página 13.
- GARCIA-GARCIA, A. et al. A review on deep learning techniques applied to semantic segmentation. **CoRR**, abs/1704.06857, 2017. Disponível em: <<http://arxiv.org/abs/1704.06857>>. Citado na página 27.
- GAREY, M.; JOHNSON, D. The boxes-and-arrows problem is np-complete. **SIAM journal on applied mathematics**, SIAM, v. 35, n. 1, p. 147–159, 1982. Citado na página 30.
- GATZKE, T.; GRIMM, C. Estimating curvature on triangular meshes. **International Journal of Shape Modeling**, v. 12, p. 1–28, 06 2006. Citado na página 27.
- HE, Y. et al. **Deep Learning based 3D Segmentation: A Survey**. 2021. Citado na página 33.
- JIMÉNEZ, M. et al. Additive manufacturing technologies: An overview about 3d printing methods and future prospects. **Complexity**, Hindawi, v. 2019, p. 9656938, Feb 2019. ISSN 1076-2787. Disponível em: <<https://doi.org/10.1155/2019/9656938>>. Citado na página 20.
- KHRONOS. **OpenGL**. 1992. Acessado em: 30/12/2022. Disponível em: <<https://www.opengl.org/>>. Citado na página 38.
- KONIG, N. et al. Method for segmentation and hybrid joining of additive manufactured segments in prototyping using the example of trim parts. **Designs**, v. 6, n. 1, 2022. ISSN 2411-9660. Disponível em: <<https://www.mdpi.com/2411-9660/6/1/2>>. Citado 2 vezes nas páginas 19 e 35.
- LEONARDIS, A.; JAKLIC, A.; SOLINA, F. Superquadrics for segmenting and modeling range data. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 19, n. 11, p. 1289–1295, 1997. Citado na página 34.

- LIAW, C.-Y.; GUVENDIREN, M. Current and emerging applications of 3d printing in medicine. **Biofabrication**, IOP Publishing, v. 9, n. 2, p. 024102, jun 2017. Disponível em: <<https://dx.doi.org/10.1088/1758-5090/aa7279>>. Citado na página 12.
- LU, L.; LéVY, B.; WANG, W. Centroidal voronoi tessellation of line segments and graphs. **Comput. Graph. Forum**, The Eurographs Association John Wiley Sons, Ltd., Chichester, GBR, v. 31, n. 2pt4, p. 775–784, may 2012. ISSN 0167-7055. Disponível em: <<https://doi.org/10.1111/j.1467-8659.2012.03058.x>>. Citado na página 37.
- LUBAN. **luban3d**. 2022. Acessado em: 22/12/2022. Disponível em: <<https://www.luban3d.com/>>. Citado na página 19.
- MUNHOZ, A. L. J. **Manufatura aditiva. Tecnologias e aplicações da impressão 3d**. [S.l.]: Blucher, 1977. v. 1. Citado na página 19.
- MURRAY, J.; VANRYPER, W. **Encyclopedia of Graphics File Formats**. O'Reilly & Associates, 1996. (O'Reilly Series). ISBN 9781565921610. Disponível em: <<https://books.google.com.br/books?id=IYsZAQAIAAJ>>. Citado na página 15.
- MUV. **Importância da prototipagem 3D no desenvolvimento de produtos**. 2019. Acessado em: 21/12/2022. Disponível em: <<https://blog.muv.ind.br/importancia-prototipagem-3d>>. Citado na página 19.
- NETO, J. S. **Como funciona o algoritmo K-means**. 2018. Acessado em: 21/12/2022. Disponível em: <<https://www.computersciencemaster.com.br/como-funciona-o-algoritmo-k-means/>>. Citado na página 19.
- OLSZTA, P. W. **Freelut**. 2005. Acessado em: 09/07/2023. Disponível em: <<https://freelut.sourceforge.net/>>. Citado na página 38.
- PETERS, A. **O primeiro bairro impresso em 3D do mundo agora tem suas primeiras casas**. 2019. Acessado em: 22/10/2022. Disponível em: <<https://www.fastcompany.com/90440406/the-worlds-first-3d-printed-neighborhood-now-has-its-first-houses>>. Citado na página 12.
- RUSINKIEWICZ, S. Estimating curvatures and their derivatives on triangle meshes. In: **Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004**. [S.l.: s.n.], 2004. p. 486–493. Citado na página 27.
- SANTOS, V. dos. **Como funciona o algoritmo K-means**. 2019. Acessado em: 19/12/2022. Disponível em: <<https://www.computersciencemaster.com.br/como-funciona-o-algoritmo-k-means/>>. Citado na página 36.
- SARCAR, M.; RAO, K.; NARAYAN, K. **Computer Aided Design and Manufacturing**. PHI Learning, 2008. ISBN 9788120333420. Disponível em: <<https://books.google.com.br/books?id=zXdivq93WIUC>>. Citado na página 18.
- SFIKAS, K.; THEOHARIS, T.; PRATIKAKIS, I. Rosy+: 3d object pose normalization based on pca and reflective object symmetry with application in 3d object retrieval. **International Journal of Computer Vision**, v. 91, p. 262–279, 2011. Citado na página 34.

- SINGHAL, S. et al. 3d food printing: paving way towards novel foods. **Scielo**, Anais da Academia Brasileira de Ciências, v. 92, n. 3, 2020. Disponível em: <<https://www.scielo.br/j/aabc/a/pT3CBycdMq6f7MNsKNbDXbr/?lang=en>>. Citado na página 12.
- SIPSER, M. **Introduction to the Theory of Computation**. 1st. ed. [S.l.]: International Thomson Publishing, 1996. ISBN 053494728X. Citado na página 29.
- SIVANANDAM, S.; DEEPA, S. Genetic algorithms. In: **Introduction to genetic algorithms**. [S.l.]: Springer, 2008. p. 15–37. Citado na página 28.
- STUDOCU. **Quadrática**. 2017. Acessado em: 19/12/2022. Disponível em: <<https://www.studocu.com/pt/document/universidade-nova-de-lisboa/calculo-ii/quadricas/2060484>>. Citado na página 34.
- TANG, Z. Octree representation and its applications in cad. **Journal of Computer Science and Technology**, v. 7, n. 1, p. 29–38, Jan 1992. ISSN 1860-4749. Disponível em: <<https://doi.org/10.1007/BF02946163>>. Citado na página 33.
- VANDERPLOEG, A.; LEE, S.-E.; MAMP, M. The application of 3d printing technology in the fashion industry. **International Journal of Fashion Design, Technology and Education**, Taylor Francis, v. 10, n. 2, p. 170–179, 2017. Disponível em: <<https://doi.org/10.1080/17543266.2016.1223355>>. Citado na página 12.
- XIAO, Z. et al. **Endowing Deep 3D Models with Rotation Invariance Based on Principal Component Analysis**. arXiv, 2019. Disponível em: <<https://arxiv.org/abs/1910.08901>>. Citado na página 34.
- YING, S. et al. Point cluster analysis using a 3d voronoi diagram with applications in point cloud segmentation. **ISPRS International Journal of Geo-Information**, v. 4, n. 3, p. 1480–1499, 2015. ISSN 2220-9964. Disponível em: <<https://www.mdpi.com/2220-9964/4/3/1480>>. Citado na página 36.
- ZHANG, C.; CHEN, T. Efficient feature extraction for 2d/3d objects in mesh representation. In: **Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)**. [S.l.: s.n.], 2001. v. 3, p. 935–938 vol.3. Citado na página 41.