

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

LEANDRO CHESINI

**COMPARAÇÃO DE MALHAS TRIANGULARES RECONSTRUÍDAS A PARTIR
DE NUVEM DE PONTOS COM O USO DE MÉTRICA DA QUALIDADE DO
ELEMENTO**

PONTA GROSSA

2022

LEANDRO CHESINI

**COMPARAÇÃO DE MALHAS TRIANGULARES RECONSTRUÍDAS A PARTIR
DE NUVEM DE PONTOS COM O USO DE MÉTRICA DA QUALIDADE DO
ELEMENTO**

**Use of element quality metric to compare triangular meshes reconstructed
from point cloud**

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

Orientador: Prof. Dr. Erikson Freitas de Moraes

PONTA GROSSA

2022



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

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

LEANDRO CHESINI

**COMPARAÇÃO DE MALHAS TRIANGULARES RECONSTRUÍDAS A PARTIR
DE NUVEM DE PONTOS COM O USO DE MÉTRICA DA QUALIDADE DO
ELEMENTO**

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

Data de aprovação: 01 de novembro de 2022

Erikson Freitas de Moraes
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Ponta Grossa

Simone Bello Kaminski Aires
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Ponta Grossa

Vinícius Camargo Andrade
Mestrado
Universidade Tecnológica Federal do Paraná - Campus Ponta Grossa

**PONTA GROSSA
2022**

Dedico este trabalho a meu avô Vitório (in memoriam), que sempre me incentivou a saber mais e me deu conselhos valiosos sobre como enfrentar as adversidades da vida.

AGRADECIMENTOS

Agradeço à minha família, pelo apoio, conversas e bons momentos, em especial à minha mãe Marcia que em suas orações pediu por esse momento de conquista e felicidade.

Agradeço à minha namorada e companheira Karoline que esteve comigo nos momentos mais difíceis, por me apoiar, incentivar e abrir mão dos nossos momentos juntos para que eu pudesse me dedicar ao desenvolvimento deste trabalho.

Agradeço aos amigos que fiz durante a graduação, todos os momentos de descontração, assim como os de intenso estudo, foram essenciais para se manter forte e feliz no decorrer desses anos, em especial à minha amiga Janaine por me acompanhar e dividir a maioria dos momentos de diversão e trabalho.

Agradeço aos professores do DAINF por compartilharem seu nobre conhecimento durante as aulas, em especial ao meu orientador de iniciação científica e TCC Prof. Dr. Erikson Freitas de Moraes que sempre me incentivou e me apoiou nos momentos mais difíceis. Agradeço também aos professores que participaram da minha banca, suas considerações contribuíram muito para o desenvolvimento deste trabalho.

Agradeço à professora Daiane Maria de Genaro Chiroli pela orientação no trabalho de extensão que tive o prazer de participar, ao professor Geraldo Ranthum e ao meu colega Fernando Tetsuya pelo apoio técnico necessário. Esse trabalho com certeza contribuiu muito para o meu desenvolvimento pessoal e profissional.

Agradeço à UTFPR onde estudei durante esses anos da graduação, instituição pública, gratuita e de qualidade, que contribui para guiar nossa sociedade pela ciência e pela educação.

Por fim, agradeço a todos que ajudaram e contribuíram direta ou indiretamente para a realização desse trabalho, foram dias difíceis e sem o apoio dos mais próximos esse momento não seria possível.

RESUMO

A realidade virtual (RV) é uma interface avançada de interação com sistemas computacionais onde o usuário pode visualizar, se movimentar e interagir com um ambiente tridimensional em tempo real. Esse ambiente é o elemento composto de formas que representam o mundo real ou abstrato definido por sua geometria, cor e iluminação. Com a geração de modelos tridimensionais, é possível ao usuário interagir com esses sistemas pela visualização do ambiente, sendo que a imagem pode ser apresentada em um monitor, no caso de sistemas não-imersivos ou com o auxílio de óculos ou capacetes de realidade virtual para sistemas imersivos de realidade virtual. Usar sistemas de RV para construir gêmeos digitais de cidades reais a fim de permitir a participação e a colaboração da sociedade nos processos de tomada de decisão, é um dos benefícios que o avanço dessa tecnologia traz para o planejamento e monitoramento urbano em cidades inteligentes. Obter os dados do mundo real faz parte da construção dos ambientes para visualização 3D e conta com ferramentas como *scanners* 3D que capturam as coordenadas x , y e z de pontos para arquivos chamados de Nuvem de Pontos. Na geração de modelos virtuais realísticos, é preciso inicialmente reconstruir a superfície amostrada na Nuvem de Pontos e então apresentá-la com o auxílio de sistemas e dispositivos compatíveis com RV. Para aumentar a fidelidade na representação dos modelos virtuais de ambientes, é preciso estabelecer a qualidade dos métodos de reconstrução e fazer a comparação. Para isso foi utilizada a métrica *radius-ratio* $\frac{\rho}{2}$ com os métodos de reconstrução *Ball-Pivoting Algorithm* e *Greedy-Projection Triangulation* aplicados em *datasets* públicos de objetos já analisados em outros trabalhos relacionados. Foram gerados histogramas para analisar a distribuição das frequências dos valores da métrica a fim de realizar a comparação das malhas reconstruídas. Nos resultados analisados o método de reconstrução *Ball-Pivoting Algorithm* gerou malhas de melhor qualidade em comparação com o método *Greedy-Projection Triangulation*.

Palavras-chave: nuvem de pontos; reconstrução; métrica; computação gráfica.

ABSTRACT

Virtual Reality (VR) is an advanced interface that interacts with computer systems where the user can visualize, move, and interact in a three-dimensional environment in real time. This environment is an element of shapes representing the real or abstract world defined by its geometry, color, and lighting. With the creation of three-dimensional models, the user can interact with these systems by visualizing the environment, and the image can be presented on a monitor, in the case of non-immersive systems, or with the aid of glasses or virtual reality helmets to virtual reality immersive systems. Using VR systems to build digital twins of real cities that allow society's participation and collaboration in decision-making processes is one of the benefits that the advancement of this technology brings to urban planning and monitoring in smart cities. Getting data from the real world is part of the construction of environments for 3D visualization and has tools such as 3D scanners that capture the x, y, and z coordinates of points to files called Point Clouds. To make realistic virtual models, one must first reconstruct the sampled surface in the Point Cloud and then present it with the support of VR-compatible systems and devices. To increase the fidelity in the representation of virtual models of environments, it is necessary to determine the quality of the reconstruction methods and compare them. Therefore, the radius-ratio $\frac{\rho}{2}$ metric was used with Ball-Pivoting Algorithm Reconstruction method and Greedy-Projection Triangulation applied to public datasets of objects already analyzed in other related works. Histograms were generated to analyze the frequency distribution of the metric values in order to compare the reconstructed meshes. In the analyzed results, the Ball-Pivoting Algorithm reconstruction method generated better quality meshes compared to the Greedy-Projection Triangulation method.

Keywords: point cloud; reconstruction; metric; computer graphics.

LISTA DE FIGURAS

Figura 1 – Sequência de tarefas para a reconstrução a partir de nuvens de pontos: a) coleta de dados usando um <i>scanner</i> Tridimensional (3D), b) dados de entrada para algoritmos de reconstrução, c) reconstrução do modelo. . .	14
Figura 2 – Triângulo e o círculo circunscrito.	19
Figura 3 – Triângulo e o círculo inscrito.	19
Figura 4 – Relação da Computação Geométrica com outras áreas da Computação.	20
Figura 5 – Segmento de reta em um espaço tridimensional.	22
Figura 6 – Exemplo de uma malha poligonal com triângulos.	22
Figura 7 – Vetor normal de uma superfície triangular.	23
Figura 8 – A bola pivotante do algoritmo <i>Ball-Pivoting Algorithm</i> (BPA).	24
Figura 9 – Triângulos com os raios dos círculos inscrito e circunscrito: a) Um triângulo não-equilátero, b) Um triângulo equilátero.	25
Figura 10 – Exemplo de histograma com notas de alunos.	26
Figura 11 – Exemplo de histograma com o polígono de frequências.	27
Figura 12 – Triangulação de uma superfície com a densidade adaptativa de vértices, onde é preciso maior detalhamento se tem mais vértices. O algoritmo <i>Greedy Projection Triangulation</i> (GP3) também se adapta nessas situações.	28
Figura 13 – Limitações do algoritmo BPA: a) Bola pivotante muito grande causa perda de detalhes, b) Bola muito pequena causa buracos na malha. . .	29
Figura 14 – Resultados da simplificação da superfície: a) Método da simplificação por região, b) Método de Lindstrom-Turk e c) Histograma dos valores da métrica <i>radius-ratio</i>	29
Figura 15 – Fluxograma da metodologia.	31
Figura 16 – Modelos tridimensionais a) Bunny e b) Dragon.	32
Figura 17 – Primeiras nove linhas do arquivo gerado pela aplicação da métrica nos triângulos de uma malha.	36

Figura 18 – Malhas do modelo Bunny para comparar visualmente o algoritmo GP3: a) malha original, b) malha similar à original, porém com a presença de buracos, c) malha sem buracos, mas com aspecto visual diferente da malha original, d) malha muito diferente da original e com a presença de buracos.	42
Figura 19 – Malhas do modelo Dragon para comparar visualmente o algoritmo GP3: a) malha original, b) malha similar à original, porém com a presença de buracos, c) malha sem buracos, mas com aspecto visual diferente da malha original, d) malha muito diferente da original e com a presença de buracos.	43
Figura 20 – Malhas do modelo Bunny para comparar visualmente o algoritmo <i>Ball-Pivoting Algorithm</i>: a) malha original, b) calibração Mesh0_0.70%, c) calibração Mesh2_0.70%, d) calibração Mesh3_0.70%.	44
Figura 21 – Malhas do modelo Dragon para comparar visualmente o algoritmo <i>Greedy Projection Triangulation</i>: a) malha original, b) melhor malha gerada pelo método, c) malha de qualidade "média", d) Pior malha gerada pelo método.	45
Figura 22 – Histograma dos valores da métrica na malha original do modelo Bunny: a) Gráfico sem escala, b) Gráfico com escala logarítmica no eixo horizontal que apresenta a quantidade de ocorrências.	45
Figura 23 – Histograma dos valores da métrica na malha original do modelo Dragon.	46
Figura 24 – Histograma dos valores da métrica na malha original do modelo Bunny com o valor máximo definido em 1000. Em a) sem aplicação de escala, b) com escala logarítmica no eixo vertical, c) com escala logarítmica no eixo horizontal e d) com escala logarítmica nos dois eixos.	47
Figura 25 – Histograma dos valores da métrica na malha original do modelo Dragon com o valor máximo definido em 1000. Em a) sem aplicação de escala, b) com escala logarítmica no eixo vertical, c) com escala logarítmica no eixo horizontal e d) com escala logarítmica nos dois eixos.	48
Figura 26 – Histograma dos valores da métrica na malha original e em três calibrações diferentes do método <i>Greedy Projection Triangulation</i> para o modelo Bunny.	49

Figura 27 – Linha de valores do histograma na malha original e em três calibrações diferentes do método <i>Greedy Projection Triangulation</i> para o modelo Bunny.	49
Figura 28 – Histograma dos valores da métrica na malha original e em três calibrações diferentes do método <i>Greedy Projection Triangulation</i> para o modelo Dragon.	50
Figura 29 – Linha de valores do histograma na malha original e em três calibrações diferentes do método <i>Greedy Projection Triangulation</i> para o modelo Dragon.	50
Figura 30 – Histograma dos valores da métrica na malha original e em três calibrações diferentes do método <i>Ball-Pivoting Algorithm</i> para o modelo Bunny.	51
Figura 31 – Linha de valores do histograma na malha original e em três calibrações diferentes do método <i>Ball-Pivoting Algorithm</i> para o modelo Bunny.	52
Figura 32 – Histograma dos valores da métrica na malha original e em três calibrações diferentes do método <i>Ball-Pivoting Algorithm</i> para o modelo Dragon.	52
Figura 33 – Linha de valores do histograma na malha original e em três calibrações diferentes do método <i>Ball-Pivoting Algorithm</i> para o modelo Dragon.	53

LISTA DE TABELAS

Tabela 1 – Parâmetros do método de reconstrução <i>Ball-Pivoting Algorithm</i>	33
Tabela 2 – Parâmetros do método de reconstrução <i>Greedy Projection Triangulation</i>	34
Tabela 3 – Configurações da máquina utilizada	37
Tabela 4 – Ferramentas utilizadas	37
Tabela 5 – Parâmetros fixados durante a variação do <i>Ball radius</i>	38
Tabela 6 – Quantidade de triângulos gerados com o <i>Ball-Pivoting Algorithm</i> no modelo <i>Bunny</i>	39
Tabela 7 – Quantidade de triângulos gerados com o <i>Ball-Pivoting Algorithm</i> no modelo <i>Dragon</i>	40
Tabela 8 – Calibrações do método de reconstrução <i>Ball-Pivoting Algorithm</i> analisadas	41
Tabela 9 – Calibrações do método de reconstrução <i>Greedy Projection Triangulation</i> analisadas	41
Tabela 10 – Valor do coeficiente Q calculado para a malha original do modelo <i>Bunny</i> e as malhas reconstruídas pelos métodos <i>Greedy Projection Triangulation</i> e <i>Ball-Pivoting Algorithm</i>	54
Tabela 11 – Valor do coeficiente Q calculado para a malha original do modelo <i>Dragon</i> e as malhas reconstruídas pelos métodos <i>Greedy Projection Triangulation</i> e <i>Ball-Pivoting Algorithm</i>	54

LISTA DE ABREVIATURAS E SIGLAS

Siglas

3D	Tridimensional
BPA	<i>Ball-Pivoting Algorithm</i>
CAD	<i>Computer-aided Design</i>
GP3	<i>Greedy Projection Triangulation</i>
PCL	<i>Point Cloud Library</i>
RV	Realidade Virtual

LISTA DE SÍMBOLOS

Letras Gregas

ρ Razão de aspecto do triângulo

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	16
1.2	Justificativa	16
1.3	Organização do Trabalho	17
2	REFERENCIAL TEÓRICO	18
2.1	Conceitos Matemáticos	18
2.2	Formação e manipulação de imagens	20
2.2.1	Computação Geométrica	20
2.2.2	Nuvem de Pontos	20
2.2.3	Primitivas Gráficas	21
2.3	Reconstrução a Partir de Nuvem de Pontos	23
2.3.1	Ball-Pivoting Algorithm	24
2.3.2	Greedy Projection Triangulation	24
2.4	Métricas de Qualidade para Malhas Triangulares	25
2.4.1	Razão de aspecto do triângulo	25
2.4.2	Gráficos para analisar a razão de aspecto dos triângulos da Malha	26
3	TRABALHOS RELACIONADOS	28
4	DESENVOLVIMENTO	31
4.1	Malhas Triangulares: Arquivos utilizados	31
4.2	Geração de malhas com os métodos de reconstrução	33
4.2.1	Ball-Pivoting Algorithm	33
4.2.2	Greedy Projection Triangulation	34
4.3	Aplicação da métrica nas malhas geradas	35
5	RESULTADOS	37
5.1	Ambiente de Trabalho	37
5.2	Calibrações dos métodos de reconstrução	38
5.2.1	Ball-Pivoting Algorithm	38
5.2.2	Greedy Projection Triangulation	41
5.3	Características das malhas geradas	42
5.4	Comparação dos valores obtidos da métrica	45

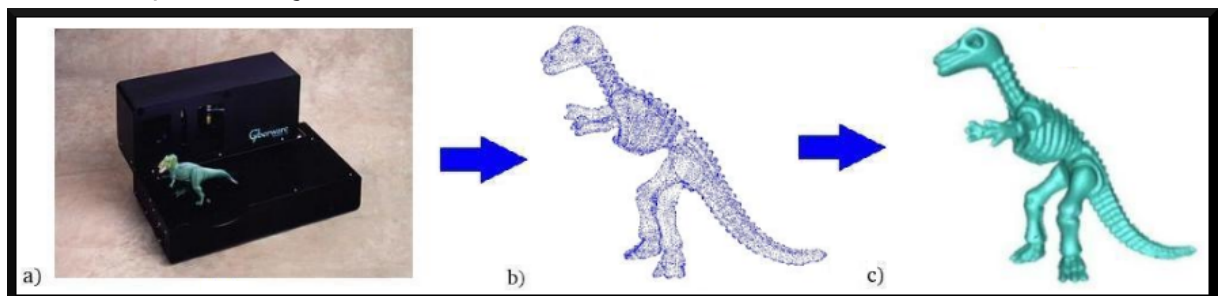
6	CONCLUSÃO	55
6.1	Trabalhos Futuros	56
	REFERÊNCIAS	57

1 INTRODUÇÃO

Ao longo de sua história, o ser humano sempre necessitou expressar-se, nos primórdios o fazia por meio de desenhos primitivos que ilustravam animais e pessoas em cenas de caça e batalhas em representações da realidade vivida pelos povos pré-históricos, passando pelo desenvolvimento da escrita (ALVES, 2006), figuras, pinturas, teatro, ópera, ilusionismo e até o cinema nas mais diversas formas de expressões artísticas. Mais recentemente, o computador potencializou o avanço e trouxe outros formatos de representação, como a multimídia, que envolve textos, imagens, sons, vídeos e animações, e a hipermídia, que permite a navegação em conteúdos multimídia de forma não-linear e interativa (TORI; KIRNER; SISCOOTTO, 2006, p. 2). A união dessas tecnologias mais recentes possibilitou gerar ambientes tridimensionais digitais que permitem a interação de pessoas em simulações da realidade ou do imaginário em tempo real, com o uso da realidade virtual (TORI; KIRNER; SISCOOTTO, 2006, p. 6-10).

Em um sistema de realidade virtual, o ambiente virtual é o elemento construído a partir formas geométricas que representam elementos do mundo real ou abstrato por meio da especificação de sua geometria e também de outros atributos como a cor e a iluminação (TORI; KIRNER; SISCOOTTO, 2006). Dadas as várias possibilidades de aplicação da realidade virtual, como exemplo, criar gêmeos digitais de cidades inteiras para o auxílio em processos colaborativos e participativos de análise, projeto e tomadas de decisão (DEMBSKI *et al.*, 2020), está entre os objetivos da Realidade Virtual (RV) permitir a visualização de objetos 3D para a sua inspeção em tempo real no caso de peças e equipamentos industriais, permitir o treinamento cirúrgico em aplicações médicas, visualizar superfícies planetárias em aplicações científicas (TORI; KIRNER; SISCOOTTO, 2006) e auxiliar no planejamento urbano quando aplicada em *smart cities* (JAMEI *et al.*, 2017).

Figura 1 – Sequência de tarefas para a reconstrução a partir de nuvens de pontos: a) coleta de dados usando um scanner 3D, b) dados de entrada para algoritmos de reconstrução, c) reconstrução do modelo.



Fonte: Adaptado de Gois (2004)

Uma das formas de construir modelos de visualização 3D é obter dados do mundo real para posterior representação computacional. A obtenção destes dados é feita por *scanners* 3D e as informações obtidas por esses equipamentos são somente as coordenadas x, y e z de pontos distribuídos em um espaço tridimensional, portanto, sem qualquer representação da relação

entre os mesmos, o que traz uma representação de baixa fidelidade em relação ao cenário no mundo real (GOIS, 2004; BERGER *et al.*, 2017). A esse conjunto de dados é dado o nome de nuvem de pontos, que serve como base para a aplicação de algoritmos de reconstrução de superfícies, também chamada de geração de malhas superficiais, sendo abordadas nesse trabalho somente as malhas triangulares. A Figura 1 apresenta a sequência básica de tarefas para a reconstrução de superfícies a partir de nuvens de pontos, onde a partir de um objeto do mundo real é gerada uma representação computacional aproximada (GOIS, 2004).

O processo de reconstrução de superfícies triangulares é chamado de triangulação, Gois (2004) apresenta a seguinte definição: Seja X um conjunto de pontos em \mathbb{R}^2 e M uma triangulação de X , para M ser considerada uma triangulação de Delaunay é preciso que dentro do circuncírculo de cada triângulo gerado não exista nenhum outro ponto de X .

Gois (2004) traz em seu trabalho algumas propriedades importantes dessa triangulação no espaço \mathbb{R}^2 , em que o ângulo mínimo de cada triângulo é maximizado e também é possível obter a triangulação em tempo $O(n \log n)$, o que não ocorre em \mathbb{R}^3 , onde o custo computacional inviabiliza a aplicação de métodos baseados em Delaunay. Há uma vasta literatura (BERGER *et al.*, 2017; BERNARDINI *et al.*, 1999; DIGNE, 2014; MARTON; RUSU; BEETZ, 2009) no que diz respeito à reconstrução de modelos 3D a partir de nuvens de pontos, onde constam métodos que foram classificados por Gois (2004) como métodos baseados em esculpimento, métodos baseados em funções implícitas, métodos incrementais e métodos baseados em modelos deformáveis.

Métodos incrementais de reconstrução são uma boa alternativa para gerar malhas superficiais sem a necessidade de obter a triangulação de Delaunay para toda a nuvem de pontos no espaço tridimensional \mathbb{R}^3 , dado que este é um processo computacionalmente custoso (GOIS, 2004), onde a vantagem desses métodos é gerar uma triangulação de Delaunay local para cada ponto e conectá-las de forma a gerar uma malha triangular global.

Pébay e Baker (2003) analisaram as métricas comumente utilizadas para determinação da qualidade em malhas triangulares e definiram com provas matemáticas limites precisos de valores para essas métricas a fim de fornecer resultados úteis para uma melhor avaliação de malhas triangulares. Em muitos casos, os valores-limite das métricas foram declarados ou assumidos sem prova, mas é necessário estabelecê-los com rigor (PÉBAY; BAKER, 2003).

O presente trabalho abordará a aplicação de métodos incrementais de reconstrução já existentes na literatura a fim de aplicar uma métrica nas malhas triangulares geradas e determinar suas qualidades, e ainda, como medir o tempo necessário para o processo de triangulação. A triangulação e análise serão feitas em nuvens de pontos de modelos já conhecidos presentes na literatura (CURLISS; LEVOY; TURK, 1994) e o resultado da triangulação será inserido em um ambiente de visualização 3D com o uso do *software Meshlab* (RANZUGLIA *et al.*, 2013).

1.1 Objetivos

O objetivo do presente trabalho é gerar malhas triangulares dos objetos *Bunny* e *Dragon* com o algoritmo da Bola Pivotante BPA e o método de Triangulação com Projeção Crescente GP3 para avaliar a qualidade através do uso da métrica *radius-ratio*. Os objetivos específicos são:

- Gerar as malhas triangulares a partir das nuvens de pontos obtidas;
- Aplicar a métrica escolhida nas malhas triangulares geradas;
- Gerar histogramas com os valores obtidos pela aplicação métrica;
- Comparar visualmente as malhas, assim como os histogramas;
- Propor um coeficiente único para representar a qualidade geral da malha.

1.2 Justificativa

Considerando o grande aumento da população global e a escassez de recursos, se faz necessário pensar na sustentabilidade como parte do planejamento futuro da sociedade. Atualmente a evolução da tecnologia da informação e comunicação tem mudado a forma como as cidades funcionam e ainda, trás à tona o conceito de cidades inteligentes ou *smart cities* (JAMEI *et al.*, 2017). As *smart cities* podem ser definidas como as cidades que objetivam aumentar a qualidade de vida de seus moradores, proporcionando o desenvolvimento sustentável através do uso de tecnologias da informação e comunicação (JAMEI *et al.*, 2017).

Uma das tecnologias que podem ser utilizadas em cidades inteligentes é a realidade virtual, definida como uma tecnologia computacional que tem a finalidade de representar digitalmente ambientes e objetos reais ou imaginários e sua demanda principal é a construção de modelos de visualização 3D. Essa tecnologia pode trazer melhorias nos processos de predição de situações-problema que surgem com a aplicação de tecnologias emergentes em ambientes complexos, como as cidades inteligentes, por possibilitar benefícios como avaliar consequências de projetos de cidades a partir de várias perspectivas e entender os diferentes tipos de informação a respeito de planejamento urbano (JAMEI *et al.*, 2017).

Para determinar a eficiência de um método de reconstrução de superfícies é necessário medir não somente o tempo de execução necessário, mas também a qualidade da malha gerada, o que se torna uma tarefa complicada, posto que, existem diversos aspectos envolvidos, dentre eles a qualidade visual, a quantidade e a razão de aspecto dos triângulos gerados (JUNIOR, 2008, p. 89-102).

Segundo Gois (2004, p. 5), é preciso realizar comparações teóricas e práticas de métodos de reconstrução de forma simultânea e apresentar exemplos para não deixar vaga a

interpretação da eficiência dos algoritmos. Dessa forma, o presente trabalho se torna relevante ao reconstruir computacionalmente modelos já conhecidos (CURLISS; LEVOY; TURK, 1994) e comparar a qualidade das malhas geradas com o auxílio de métodos definidos na literatura (BERNARDINI *et al.*, 1999; MARTON; RUSU; BEETZ, 2009) e apresentados na seção 2.3.

1.3 Organização do Trabalho

Este trabalho está estruturado em seis capítulos. No capítulo dois, são apresentados os trabalhos relacionados encontrados na literatura. Além disso, o comportamento dos algoritmos e a sua classificação são comentados e discutidos. Em seguida, após a apresentação dos trabalhos relacionados, é apresentada a fundamentação teórica em relação a reconstrução de malhas poligonais a partir de nuvens de pontos. A fundamentação teórica é necessária para o desenvolvimento deste trabalho, bem como a forma de analisar o comportamento da métrica aplicada. Nos capítulos 4 e 5, é apresentada a metodologia proposta para a execução dos objetivos, as implementações e a análise realizada. Por fim, no capítulo 6, realiza-se a conclusão do trabalho e são apresentadas sugestões para possíveis estudos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo são abordados os assuntos necessários para a compreensão e desenvolvimento do presente trabalho. Na Seção 2.1 são apresentados conceitos matemáticos sobre nuvens de pontos 3D e os círculos do triângulo, na Seção 2.2 são definidos alguns conceitos teóricos sobre a geração de nuvens de pontos e a representação de ambientes 3D e malhas triangulares. Já na Seção 2.3 é tratado sobre a reconstrução a partir de nuvem de pontos na perspectiva dos métodos incrementais abordados neste trabalho, assim com alguns detalhes do seu funcionamento. Por fim, na Seção 2.4 é discutido sobre o uso da razão de aspecto dos triângulos de uma malha como métrica para determinar sua qualidade, e como a geração de histogramas pode ser útil para a comparação de qualidade das malhas geradas.

2.1 Conceitos Matemáticos

Essa seção apresenta alguns conceitos matemáticos necessários para a compreensão de espaços e nuvens de pontos 3D, assim como os círculos do triângulo, que servem como base para a métrica de qualidade $\frac{\rho}{2}$.

A nuvem de pontos N é definida na Equação 1 para representar uma superfície amostrada em espaço tridimensional com as coordenadas x, y e z , chamado de espaço euclidiano. Essa notação será utilizada ao longo do trabalho para representar os pontos amostrados que representam a superfície dos objetos a serem reconstruídos.

$$N = \{P_{(x,y,z)} \in \mathbb{R}^3 | x, y, z \in \mathbb{R}\} \quad (1)$$

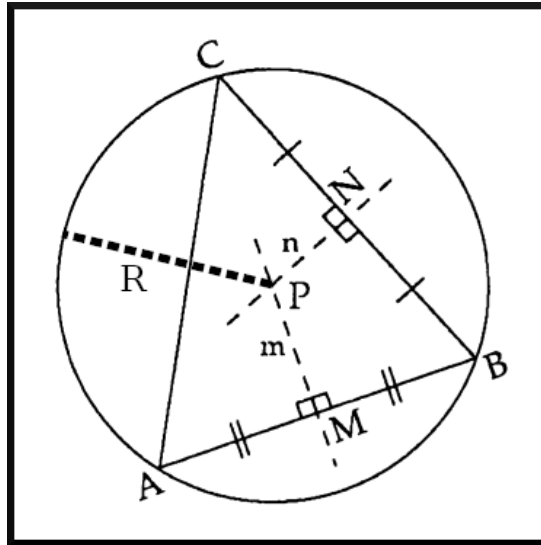
O espaço euclidiano tridimensional pode ser definido matematicamente por meio da métrica euclidiana usual apresentada na Equação 2, onde o resultado é a distância entre dois pontos $P = (P_x, P_y, P_z)$ e $Q = (Q_x, Q_y, Q_z) \in \mathbb{R}^3$ (GOIS, 2004, p. 10).

$$\|P - Q\| = \sqrt{(P_x - Q_x)^2 + (P_y - Q_y)^2 + (P_z - Q_z)^2} \quad (2)$$

Por definição, na Geometria Euclidiana Plana um triângulo sempre possui um círculo de raio R que toca em seus três pontos, como ilustrado na Figura 2. Uma vez que o triângulo está inscrito no círculo, podemos dizer que o círculo está circunscrito no triângulo ou chamá-lo de circuncírculo.

Para obter o centro P do círculo circunscrito apresentado na Figura 2, devemos traçar uma reta m perpendicular ao lado \overline{AB} do triângulo que passa por seu ponto médio M e também traçar outra reta n perpendicular ao lado \overline{BC} do triângulo que passa por seu ponto médio N . O encontro dessas duas retas é o centro do círculo circunscrito que é o Ponto P equidistante dos pontos A, B e C , então para se obter o raio, calcula-se a distância do centro a qualquer um dos pontos do triângulo (BARBOSA, 2006, p. 135-136).

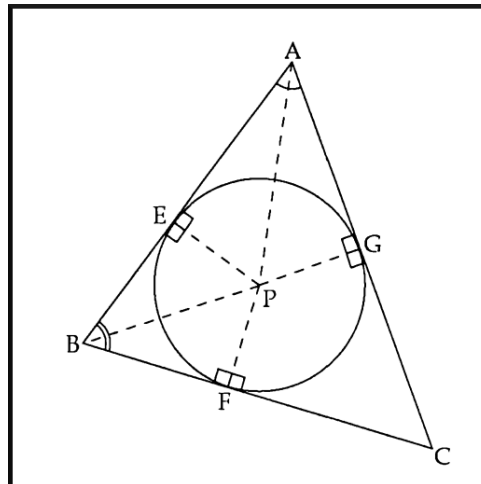
Figura 2 – Triângulo e o círculo circunscrito.



Fonte: Adaptado de BARBOSA (2006).

Também por definição da Geometria Euclidiana Plana, o triângulo sempre irá possuir um círculo inscrito, podendo ser chamado de incírculo, que tangencia os três lados do triângulo como ilustrado na Figura 3.

Figura 3 – Triângulo e o círculo inscrito.



Fonte: (BARBOSA, 2006).

Para obter o centro desse círculo, devemos traçar a reta bissetriz dos ângulos \hat{A} e \hat{B} do triângulo, sendo que na interseção dessas se encontra o Ponto P . Esse ponto é equidistante dos lados \overline{AB} , \overline{AC} e \overline{BC} do triângulo, então para obter o raio r do círculo inscrito é preciso traçar uma reta perpendicular a qualquer lado do triângulo, sendo que $r = \overline{PE} = \overline{PF} = \overline{PG}$ (BARBOSA, 2006, p. 137-140).

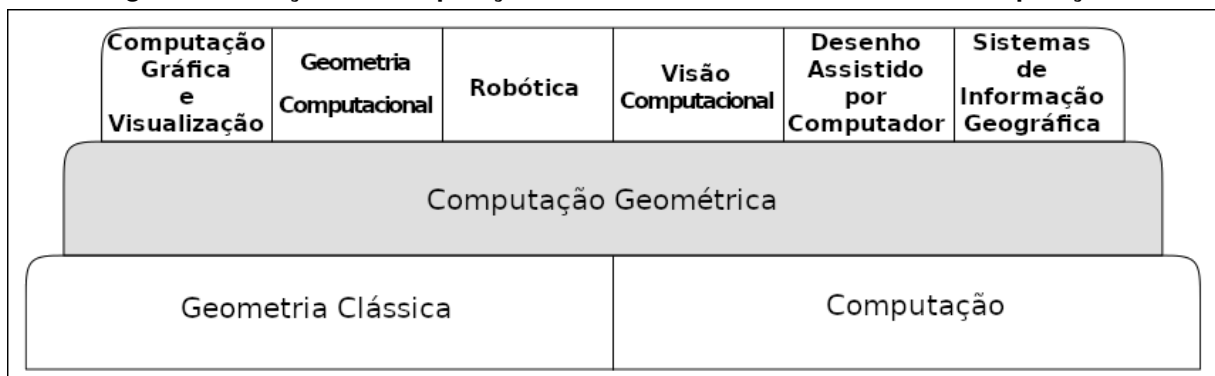
2.2 Formação e manipulação de imagens

Para um melhor entendimento dos métodos de reconstrução e visualização abordados no desenvolvimento deste trabalho, serão definidos nessa seção os conceitos matemáticos que se fazem essenciais para realizar operações de transformações geométricas, representar nuvens de pontos e gerar a aproximação da superfície que representa o ambiente 3D amostrado.

2.2.1 Computação Geométrica

De acordo com Ghali (2008) a Computação Geométrica não pode ser considerada Computação Gráfica nem Geometria Computacional, pois é definida como a matemática e engenharia que as fundamentam, assim como Visão Computacional e os demais conceitos apresentados na Figura 4.

Figura 4 – Relação da Computação Geométrica com outras áreas da Computação.



Fonte: Adaptado de Ghali (2008)

A área de Geometria Computacional pode ser definida como a ferramenta de representação de segmentos de linhas e polígonos discretos. O estudo de algoritmos para resolução de problemas geométricos em um computador tem significativo crescimento acompanhada de áreas como a computação gráfica, robótica e *Computer-aided Design* (CAD), pois a forma é uma propriedade intrínseca de objetos do mundo real (SKIENA; REVILLA, 2003).

2.2.2 Nuvem de Pontos

O avanço no desenvolvimento de tecnologias para a representação de objetos e ambientes reais em computadores trouxe consigo o aumento da relevância no uso de nuvens de pontos 3D. As tecnologias variam de métodos ativos, como *scanners* óticos à laser, *scanners* estruturados de luz e *scanners* LiDAR a métodos passivos, como o uso de visão estéreo com múltiplas câmeras que fazem a geração de pontos amostrados da superfície de objetos ou ambientes de interesse (BERGER *et al.*, 2017, p. 301).

Os métodos de reconstrução de superfícies têm diferentes requisitos de entrada referentes à nuvem de pontos, sendo que o requisito mínimo comum entre todos os algoritmos é um conjunto de pontos 3D que representam a superfície do objeto ou ambiente (BERGER *et al.*, 2017, p. 306). Os *datasets* presentes no repositório de escaneamento 3D da Universidade Stanford (CURLLESS; LEVOY; TURK, 1994) que foram usados no desenvolvimento deste trabalho cumprem com o requisito mínimo que é a informação tridimensional de cada ponto amostrado, mas também possuem a malha já reconstruída que representa a topologia, ou *wireframe* ideal dos objetos escaneados. No caso dos métodos *Ball-Pivoting Algorithm* (BERNARDINI *et al.*, 1999) e *Greedy-Projection Triangulation* (MARTON; RUSU; BEETZ, 2009) essa malha não é utilizada como entrada inicial para a execução, mas foi utilizada nesse trabalho como referência para as comparações com as malhas geradas.

Como a unidade de medida das coordenadas dos pontos amostrados varia de acordo com o equipamento usado para fazer o escaneamento, não foi encontrado um padrão entre as malhas analisadas. Na implementação do algoritmo de reconstrução *Ball-Pivoting Algorithm* encontrada em (MUNTONI; CIGNONI, 2021) é definida uma diagonal D , usada para definir o parâmetro tamanho da bola pivotante. D pode ser representada pela equação 2 da distância euclidiana, tendo $P = (max(x), max(y), max(z))$ como o maior valor encontrado de cada coordenada dos pontos do arquivo, e $Q = (min(x), min(y), min(z))$ definido pelo valor mínimo dessas coordenadas.

2.2.3 Primitivas Gráficas

Primitivas gráficas são elementos geométricos de baixa complexidade usados na computação gráfica para apresentar objetos ou ambientes computacionais na superfície de exibição, por exemplo, um monitor de vídeo, definida como uma matriz de *pixels* (TRAINA; OLIVEIRA, 2006, p. 23). Cada *pixel* de uma tela pode ser considerado um ponto do plano euclidiano definido na Equação 3.

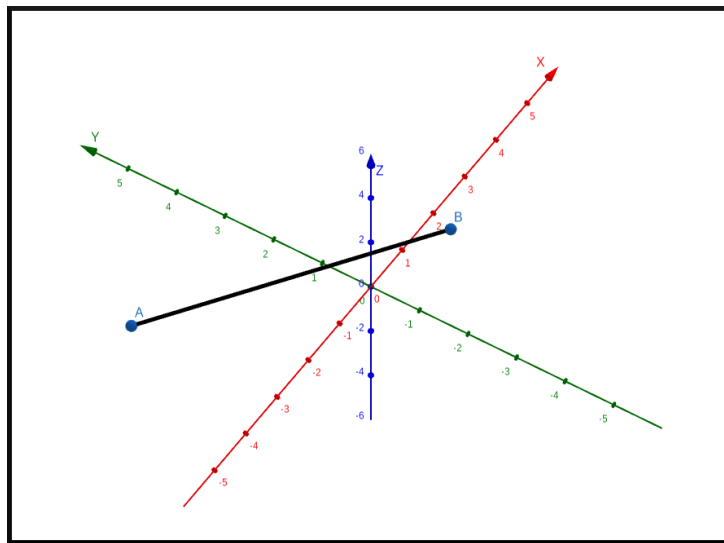
$$P(x, y) = \begin{bmatrix} P_x \\ P_y \end{bmatrix} \quad (3)$$

O ponto definido na Equação 4 serve como base para a aplicação de transformações geométricas. Apresenta três valores de coordenada que definem sua posição no espaço euclidiano \mathbb{R}^3 e além de estar presente nas nuvens de pontos, é utilizado para definir a posição do observador da cena, posição de objetos, áreas de recorte e fronteiras. (TRAINA; OLIVEIRA, 2006, p. 63).

$$P(x, y, z) = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (4)$$

A Figura 5 apresenta um segmento de reta no espaço 3D \overline{AB} , onde $A = (-3, -3, 0)$ e $B = (1, -1, 2) \in \mathbb{R}^3$. Um segmento de reta pode ser definido a partir de dois pontos de um mesmo espaço \overline{PQ} , onde $P = (P_x, P_y, P_z)$ e $Q = (Q_x, Q_y, Q_z) \in \mathbb{R}^3$, ou a partir da interseção entre dois planos.

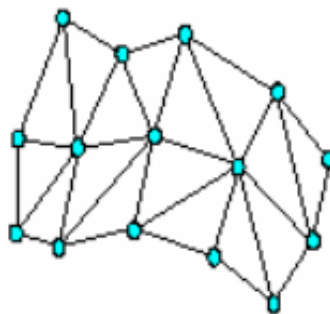
Figura 5 – Segmento de reta em um espaço tridimensional.



Fonte: Autoria própria

Um Triângulo pode ser definido pela união de três segmentos de reta e seus respectivos pontos, representado por $\Delta ABC = \overline{AB} \cup \overline{BC} \cup \overline{CA}$, onde os segmentos de reta não podem ser colineares.

Figura 6 – Exemplo de uma malha poligonal com triângulos.

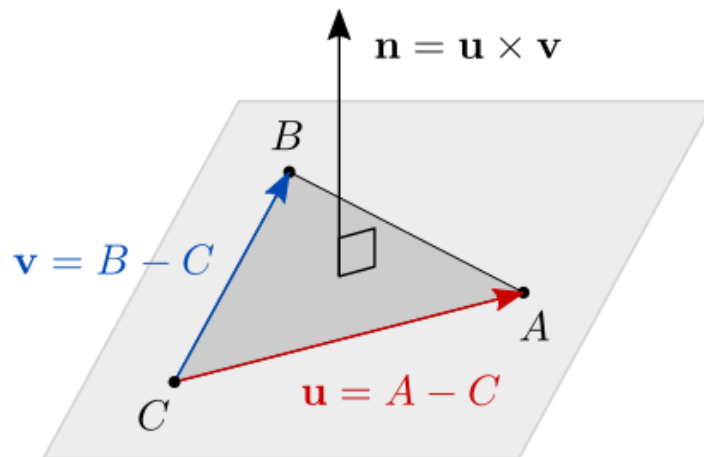


Fonte: Adaptado de Tori, Kirner e Siscoutto (2006)

A forma mais simples de definir uma superfície é representá-la como um plano (TORI; KIRNER; SISCOOTTO, 2006, p. 251), que pode ser construída com três pontos distintos e que não pertencem a uma mesma reta formando um triângulo. Uma superfície representada por um conjunto de triângulos é denominada superfície poligonal, como a ilustração da Figura 6, onde a representação de objetos não planos é feita por um grande número desses polígonos e apesar de ser semelhante à forma original, traz evidências claras da aproximação realizada (TORI; KIRNER; SISCOOTTO, 2006, p. 252).

Considerando um plano no espaço euclidiano, a sua normal, ou vetor normal, é o vetor perpendicular à superfície que pode ser calculado pela multiplicação de dois outros vetores definidos no plano do triângulo (ANTON; RORRES, 2012). Esses dois vetores podem ser obtidos na subtração dos vértices que formam duas arestas do triângulo, conforme a Figura 7.

Figura 7 – Vetor normal de uma superfície triangular.



Fonte: (BATAGELO; MARQUES, 2021)

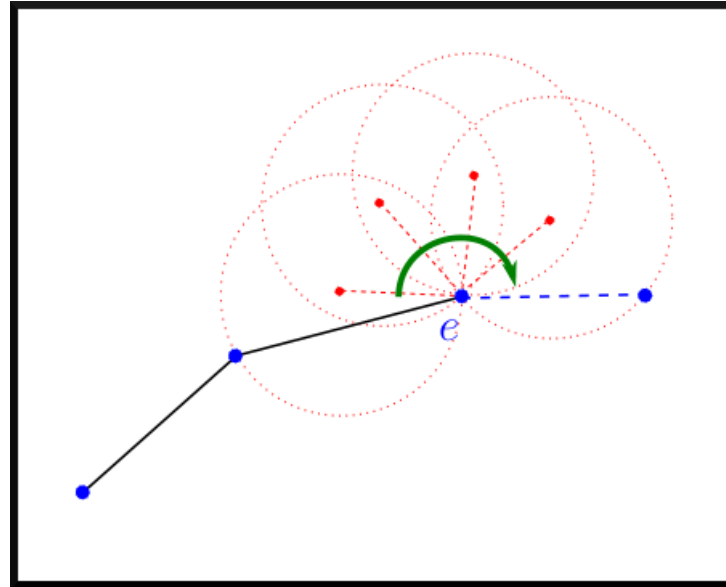
2.3 Reconstrução a Partir de Nuvem de Pontos

O problema tradicionalmente conhecido como reconstrução de superfícies consiste na representação digital de uma superfície que foi previamente escaneada e como resultado desse escaneamento, se obtém um conjunto de pontos discretos que representa a sua forma física (BERGER *et al.*, 2017). Em seu trabalho, Gois (2004) apresenta a definição formal para o problema da reconstrução de superfícies da seguinte forma: Considere uma superfície $S \subset \mathbb{R}^3$ e uma amostragem de pontos $P \subset S$, definida como nuvem de pontos de S . A reconstrução é em grande parte dos trabalhos uma aproximação linear por partes S' de S com a intenção de satisfazer as condições: a) S' deve manter a mesma topologia de S e b) Dado um ponto de S' , a sua distância em relação à superfície S deve ser suficientemente pequena.

Gois (2004) traz em seu trabalho a classificação de alguns métodos de reconstrução, entre eles estão os métodos incrementais que têm como princípio considerar uma aresta inicial

obtida pela conexão entre dois pontos que é a fronteira de avanço da reconstrução, e a partir desta segue de forma iterativa obtendo e agregando as demais arestas para gerar a superfície.

Figura 8 – A bola pivotante do algoritmo BPA.



Fonte: Digne (2014)

2.3.1 Ball-Pivoting Algorithm

O método *Ball-Pivoting Algorithm* apresentado por Bernardini *et al.* (1999) é um algoritmo incremental que considera uma 2-variedade em \mathbb{R}^3 , uma amostragem densa e uma bola de raio ρ que não atravesse a variedade sem tocar em pontos da amostragem. Sua execução inicia considerando que a bola de raio ρ toca em três pontos da nuvem, que definem um *seed triangle*. Para avançar na construção da malha essa bola é pivoteada ao redor de uma aresta já conectada até tocar um outro ponto, e caso a bola não contenha nenhum outro ponto dentro dela, uma nova aresta é criada. O processo de pivoteamento da bola é ilustrado na Figura 8.

A vantagem do algoritmo BPA (BERNARDINI *et al.*, 1999) é dispensar o cálculo da triangulação de Delaunay, que representa a maior parcela do custo associado aos algoritmos de esculpimento.

2.3.2 Greedy Projection Triangulation

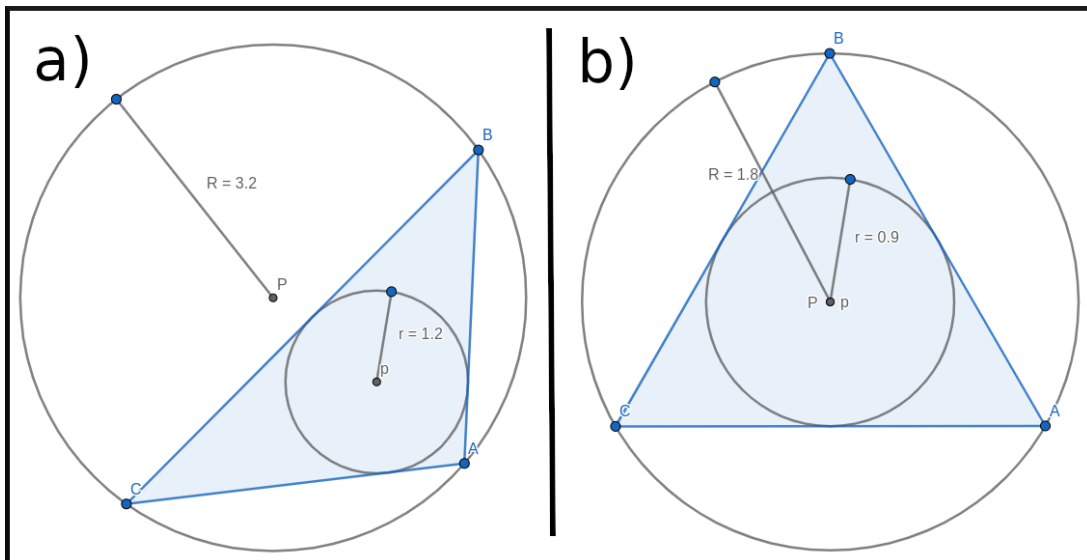
O trabalho de Marton, Rusu e Beetz (2009) propõe um método de reconstrução incremental para aplicação em *datasets* com grande quantidade de pontos e testado em modelos de ambientes internos e externos. Nesse algoritmo, que possui uma abordagem baseada em planos de projeção local, é selecionado um triângulo inicial e novos triângulos são conectados até que não hajam mais pontos a serem processados ou não seja possível gerar triângulos válidos,

sendo que no segundo caso é selecionado um novo triângulo e a triangulação começa novamente (MARTON; RUSU; BEETZ, 2009). A geração de novos triângulos se dá pela seleção dos vizinhos mais próximos dos pontos determinados por uma esfera de raio adaptativo que varia de acordo com a densidade local na nuvem de pontos, com a classificação de triângulos válidos feita através do ângulo máximo e, opcionalmente, pelo ângulo mínimo (MARTON; RUSU; BEETZ, 2009).

2.4 Métricas de Qualidade para Malhas Triangulares

Em seu trabalho Dietrich (2008, p. 99-101) utiliza a razão entre os raios dos círculos inscrito e circunscrito no triângulo (ρ) para avaliar a qualidade dos triângulos das malhas. Também afirma que as métricas de qualidade de triângulos quantificam o impacto da qualidade destes sobre a aplicação, ou seja, podem avaliar o impacto do aspecto dos triângulos sobre o processamento subsequente da malha. Estas métricas foram (e são) desenvolvidas principalmente na área de Mecânica Computacional, onde é analisado como a forma do triângulo influencia na convergência da solução do problema, nos erros de interpolação e no condicionamento dos sistemas que representam o comportamento do objeto.

Figura 9 – Triângulos com os raios dos círculos inscrito e circunscrito: a) Um triângulo não-equilátero, b) Um triângulo equilátero.



Fonte: Autoria própria.

2.4.1 Razão de aspecto do triângulo

Em seu trabalho Pébay e Baker (2003) define a métrica $\frac{\rho}{2}$ como sendo a razão entre o raio R do circuncírculo e o raio r do incírculo. A fórmula para o cálculo da métrica é apresentada na Equação 5. A Figura 9 apresenta graficamente as medidas desses raios para dois triângulos

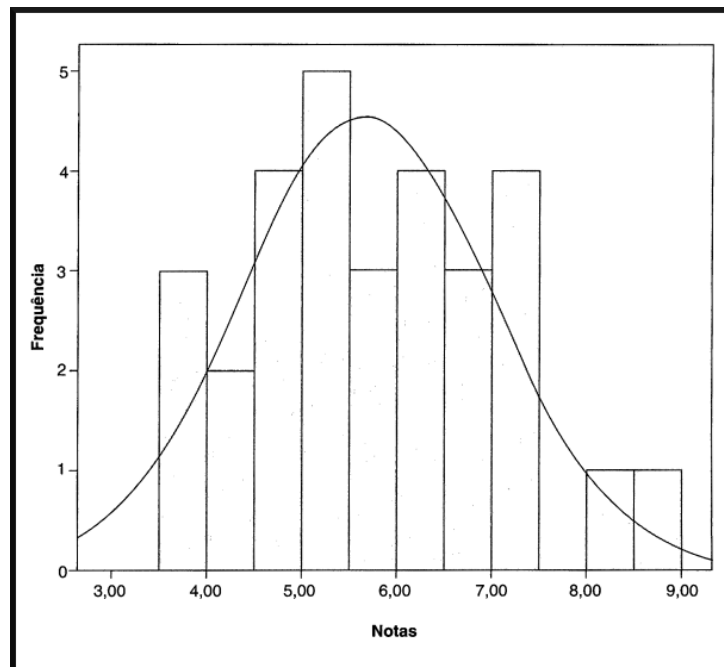
de diferentes formatos, na parte a) como sendo $R = 3.2$ e $r = 1.2$ que com o cálculo obtemos $\rho \approx 2.67$ tendo como valor final da métrica $\frac{\rho}{2} \approx 1.33$. Já na parte b) é apresentado um triângulo equilátero com $R = 1.8$ e $r = 0.9$ obtendo então $\rho = 2$ tendo como valor final da métrica $\frac{\rho}{2} = 1$, demonstrando que para triângulos equiláteros essa métrica sempre será definida como 1.

$$\rho = \frac{R}{r} \quad (5)$$

2.4.2 Gráficos para analisar a razão de aspecto dos triângulos da Malha

Dadas essas afirmações sobre a qualidade das malhas e a razão de aspecto dos triângulos, entende-se que a aplicação da métrica $\frac{\rho}{2}$ em uma malha se dá pelo cálculo desse valor em cada elemento. A quantidade de triângulos de uma malha facilmente passa de milhares e pode ultrapassar milhões, então avaliar a qualidade geral desses elementos se torna inviável. A Estatística Descritiva Univariada traz, por meio da análise da distribuição dos dados, uma forma de avaliar o comportamento sintetizado de um conjunto de dados (FÁVERO; BELFIORE, 2017, p. 21), então é uma ferramenta importante para auxiliar na avaliação da qualidade geral de malhas. Em seu trabalho Junior (2008, p. 99) já usa essa ferramenta através da geração de um histograma para fazer a comparação visual da tendência de distribuição dos dados.

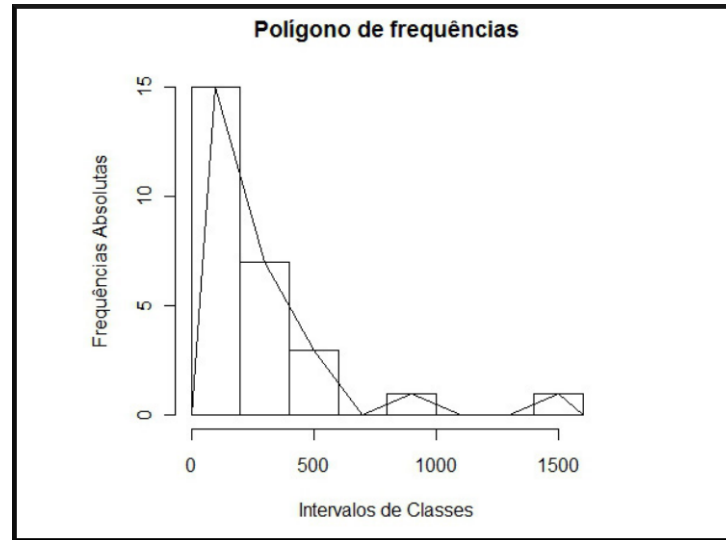
Figura 10 – Exemplo de histograma com notas de alunos.



Fonte: Adaptado de Fávero e Belfiore (2017)

Um histograma é um gráfico de barras bidimensional que apresenta a frequência com que determinados valores ocorrem (FÁVERO; BELFIORE, 2017, p. 32), como ilustrado na Figura 10 onde é exibida a quantidade de notas de alunos de uma turma agrupadas de 0,5 em 0,5.

Figura 11 – Exemplo de histograma com o polígono de frequências.



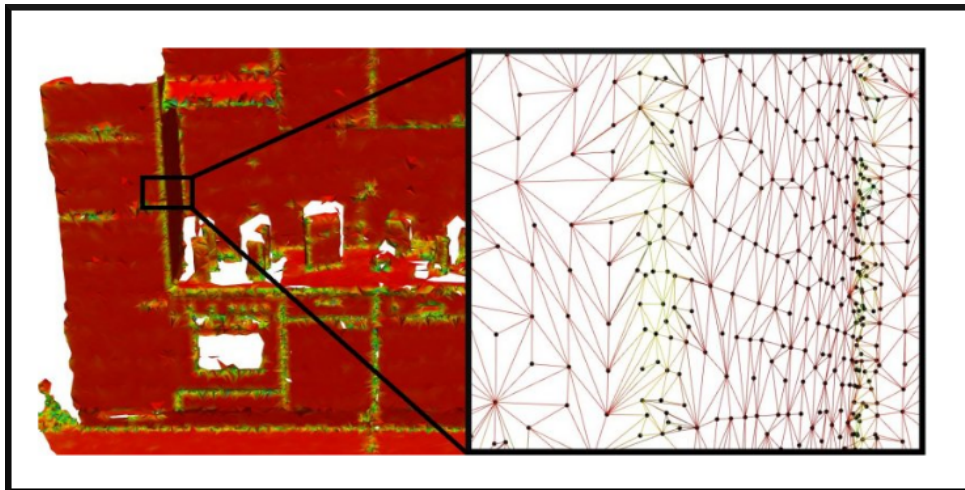
Fonte: Adaptado de Domingues e Domingues (2010)

No eixo horizontal são exibidas as classes, intervalos definidos com base nos valores a serem observados que agrupam ocorrências de valores próximos, já no eixo vertical é apresentada a quantidade de ocorrências onde é possível observar que houveram duas ocorrências entre o valor 4 e 4,5 e também nenhuma ocorrência de notas entre 7,5 e 8. Nesse exemplo foi traçada uma curva normal para exibir a tendência de ocorrência dos valores, mas nem sempre os dados estarão distribuídos dessa forma, então pode ser gerada uma linha conectada aos pontos médios dos topos de cada barra do histograma, como ilustrado na Figura 11.

3 TRABALHOS RELACIONADOS

Esse capítulo trata da discussão sobre como tem sido abordados os problemas da reconstrução de superfícies pela geração de malhas triangulares, e a comparação da qualidade de malhas geradas com o uso de métricas de razão de aspecto.

Figura 12 – Triangulação de uma superfície com a densidade adaptativa de vértices, onde é preciso maior detalhamento se tem mais vértices. O algoritmo GP3 também se adapta nessas situações.

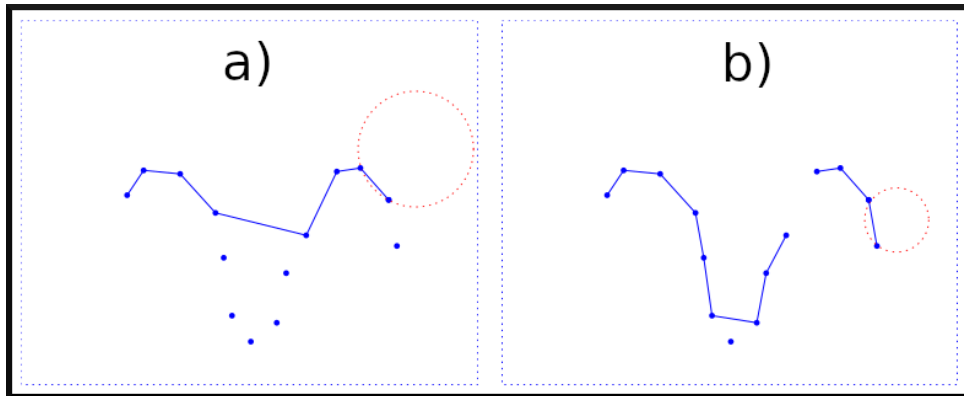


Fonte: Marton, Rusu e Beetz (2009)

A Figura 12 apresenta o resultado da triangulação de uma superfície pelo método GP3 (MARTON; RUSU; BEETZ, 2009) introduzido na Seção 2.3.2, onde as arestas com coloração verde indicam uma maior curvatura estimada da superfície e as arestas vermelhas indicam uma parte com menor curvatura estimada, de acordo com o trabalho anterior de Rusu *et al.* (2007) que estima a curvatura da superfície de acordo com a distribuição dos pontos e seus vizinhos no espaço 3D. No desenvolvimento do trabalho (MARTON; RUSU; BEETZ, 2009) que propôs o algoritmo GP3, foram apresentados o tempo de execução do algoritmo e a quantidade de triângulos gerados, porém não foram feitas comparações da qualidade das malhas através da qualidade de seus triângulos.

O algoritmo de reconstrução BPA (BERNARDINI *et al.*, 1999) introduzido na seção 2.3.1 possui uma limitação conhecida com a definição do tamanho da bola, que ao ser definida muito grande, pode prejudicar o detalhamento da malha como exemplificado na parte a) da Figura 13 ou se definida muito pequena pode trazer o aparecimento de buracos na malha como exemplificado na parte b) da Figura 13 (DIGNE, 2014). Além disso, por ser um algoritmo clássico de reconstrução, o seu tempo de execução pode ser otimizado e para isso, o trabalho de Digne (2014) desenvolve uma técnica de paralelização do método de reconstrução BPA onde se obtiveram ganhos na performance desse método e também foi sugerida a execução do método com diferentes valores do raio da bola pivotante a fim de encontrar a que melhor se adapta ao modelo a ser reconstruído.

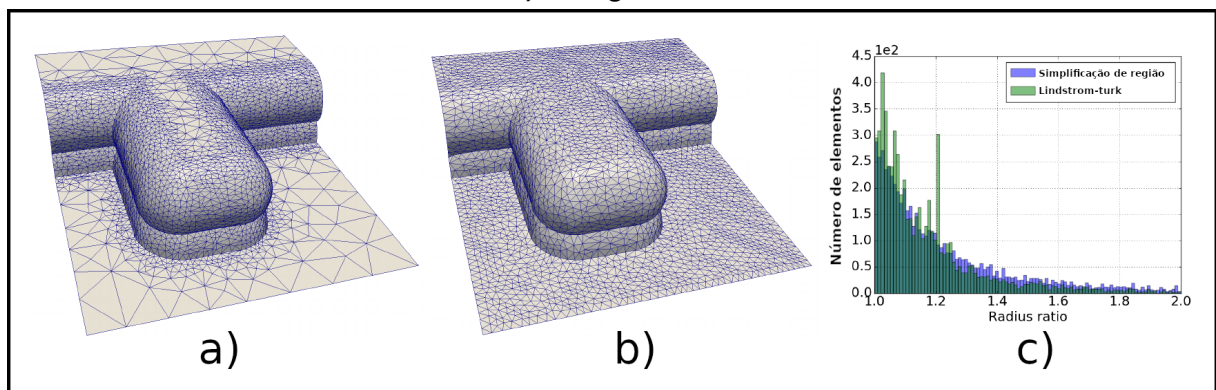
Figura 13 – Limitações do algoritmo BPA: a) Bola pivotante muito grande causada perda de detalhes, b) Bola muito pequena causa buracos na malha.



Fonte: Adaptado de Digne (2014)

O trabalho de (LENZ, 2019) propõe um método de simplificação de superfícies com foco na redução de elementos das malhas e a preservação de características geométricas. A geração de malhas por esse método visa ter triângulos melhores distribuídos e mais próximos do equilátero, a fim de otimizar a malha para as etapas de pós-processamento em simulações da fabricação de semicondutores. A simplificação de uma superfície é diferente da sua reconstrução, sendo que o problema da reconstrução visa, através de um conjunto de pontos, obter um conjunto de elementos geométricos que representam uma superfície, já o problema da simplificação tem como requisito esse conjunto de elementos geométricos previamente definido, onde o processamento feito visa alterar e/ou excluir elementos a fim de se obter uma melhora na qualidade, definida pela regularidade dos elementos da malha. Esse trabalho (LENZ, 2019) usa a distribuição de frequências dos valores da métrica *radius-ratio* (PÉBAY; BAKER, 2003) nos elementos triangulares da malha para a comparação da qualidade entre os métodos de simplificação de superfícies abordados.

Figura 14 – Resultados da simplificação da superfície: a) Método da simplificação por região, b) Método de Lindstrom-Turk e c) Histograma dos valores da métrica *radius-ratio*.



Fonte: Adaptado de Lenz (2019)

A Figura 14 apresenta a diferença entre duas malhas triangulares geradas por dois métodos de simplificação. Na parte a) da Figura é apresentada a malha gerada pelo método de

Lenz (2019) que possui uma quantidade menor de triângulos, já na parte b) da figura é apresentada a malha gerada pelo método de Lindstrom-Turk (LINDSTROM; TURK, 1998), com uma maior quantidade de triângulos. A parte c) da Figura apresenta o histograma da métrica *radius-ratio* (PÉBAY; BAKER, 2003) aplicada nessas malhas, onde é possível perceber que a malha a) gerada pelo método de simplificação de região de Lenz (2019) possui menos elementos considerados melhores se comparada com a malha b) gerada pelo método de Lindstrom-Turk. Essa afirmação é feita com base na quantidade de elementos com o valor de métrica, sendo que quanto mais próximo de 1 é esse valor, mais próximo do triângulo equilátero e portanto, melhor (LENZ, 2019).

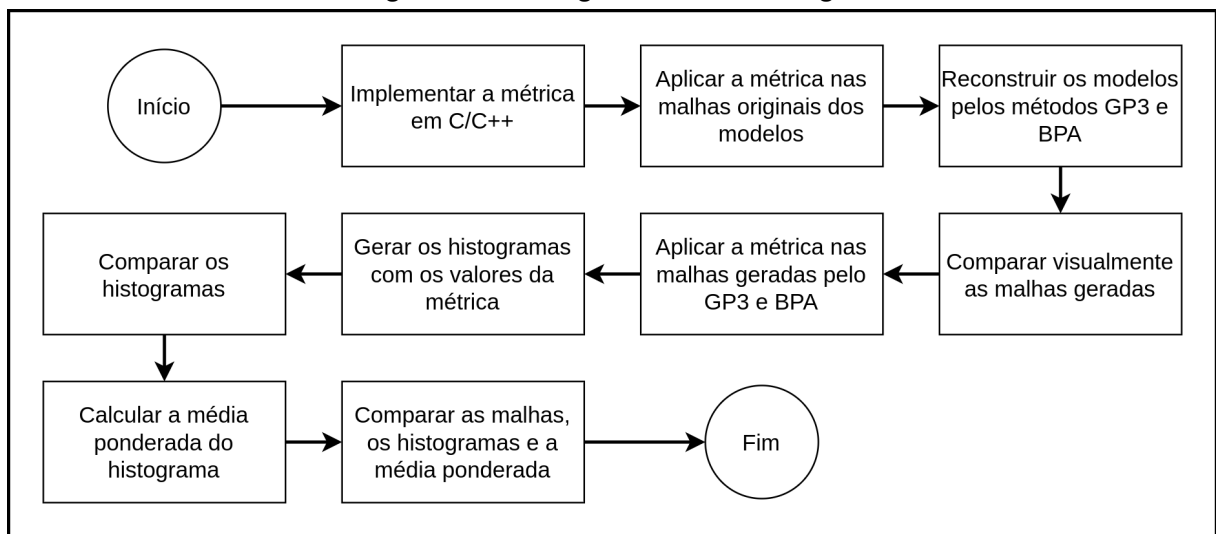
Berger *et al.* (2017) apresentou algoritmos de reconstrução de superfície por meio da perspectiva de *priors*, para lidar com as imperfeições na nuvem de pontos, buscando informações precisas sobre as dimensões da forma a ser reconstruída, sobre o campo de reconstrução de superfície e fornecendo uma categorização em relação aos *priors* e as imperfeições de dados na reconstrução da superfície. Ao considerar uma visão holística da reconstrução de superfícies, destacam-se as semelhanças entre diversas técnicas de reconstrução e propõem orientações para trabalhos futuros para a reconstrução de superfícies (BERGER *et al.*, 2017).

Em seu trabalho Junior (2008, p. 99-100) sugere que a razão de aspecto dos triângulos da malha pode ser usada para determinar sua qualidade. Para malhas consideradas boas, a média da razão de aspecto de todos os triângulos da malha deve estar próxima de 1 e o desvio padrão dessa medida deve ser usado para mensurar a variação da mesma.

4 DESENVOLVIMENTO

Este capítulo trata da metodologia abordada para o desenvolvimento deste trabalho, com os passos apresentados na Figura 15, onde inicialmente foi realizada a implementação da métrica *radius-ratio* (PÉBAY; BAKER, 2003) na linguagem C/C++ e na sequência foi realizada a aplicação dessa métrica nas malhas originais dos modelos Bunny e Dragon, a fim de se obter uma referência quanto ao valor ideal da razão de aspecto dos triângulos das malhas. Os métodos GP3 e BPA foram utilizados para a geração de novas malhas triangulares, que foram comparadas visualmente com as malhas originais.

Figura 15 – Fluxograma da metodologia.



Fonte: Autoria própria

Depois foi feita a aplicação da métrica *radius-ratio* nas malhas geradas pelos métodos de reconstrução e, com os valores obtidos foram gerados histogramas da sua distribuição de frequências. A partir desses histogramas foram feitas comparações a fim de verificar qual malha gerou mais triângulos com a razão de aspecto mais próxima de 1, ou seja, triângulos mais semelhantes ao triângulo equilátero. Além dos gráficos de histograma gerados, em sequência foi calculada a sua média ponderada a fim de se obter a qualidade da malha atribuída a um único valor numérico. Por fim foram realizadas comparações entre as malhas originais dos modelos, as malhas geradas pelos métodos de reconstrução, os histogramas de qualidade dessas malhas e a média ponderada desses histogramas.

4.1 Malhas Triangulares: Arquivos utilizados

Entre os diversos formatos de arquivo onde as nuvens de pontos e também as malhas triangulares podem ser armazenados, nesse trabalho foi utilizados o PLY, um formato criado por

Turk (1994) na Universidade Stanford e amplamente compatível com softwares de computação gráfica.

O formato de arquivo PLY é compatível com a codificação binária ou ASCII e possui capacidade para armazenar pontos tridimensionais, faces triangulares, normais dos pontos e normais das faces, definida na Seção 2.2.3, além de elementos personalizados como cores e iluminação (TURK, 1994).

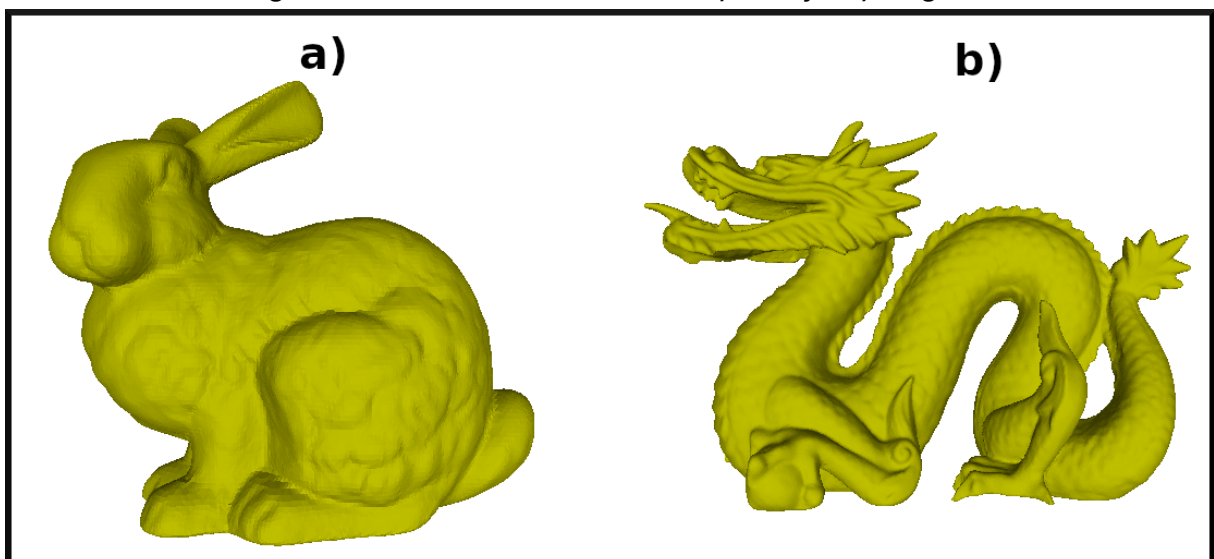
O PLY é composto por um cabeçalho onde é definida a estrutura de todos os elementos presentes no arquivo. No caso do elemento ponto são definidas as suas propriedades x , y , z e caso existam, as suas normais n_x , n_y e n_z . Já para o elemento face são definidas a quantidade de pontos da face e os seus respectivos índices.

Em seguida é definido o corpo do arquivo onde constam todos os elementos gráficos no formato definido no cabeçalho, no caso dos arquivos utilizados neste trabalho é definida uma lista de pontos e uma lista de faces.

O formato 3D encontrado nos arquivos do dataset da universidade de Osnabruck é definido na codificação ASCII, em texto legível com cada linha no formato x , y , z e temperatura em graus centígrados (NÜCHTER; LINGEMANN, 2016).

Os *datasets* dos modelos utilizados para o desenvolvimento deste trabalho já contém a malha tridimensional com os vértices e triângulos, então para gerar outras malhas com os métodos de reconstrução, somente os vértices foram passados como entrada para a execução como nuvens de pontos. A malha que já estava presente nos arquivos dos modelos foi chamada de malha original e foi a base para a comparação com as demais malhas geradas.

Figura 16 – Modelos tridimensionais a) Bunny e b) Dragon.



Fonte: Autoria própria

O modelo Bunny possui 35947 pontos, uma malha triangular com 69451 triângulos e foi ilustrado com o uso do software Meshlab (RANZUGLIA *et al.*, 2013) na Figura 16 a), já o modelo Dragon conta com 437645 pontos, 871414 triângulos e é apresentado na Figura 16 b).

Ambos os modelos têm as coordenadas dos seus pontos normalizadas nos valores de 0 a 1 e não possuem nenhuma informação sobre a unidade de medida em que elas são representadas.

4.2 Geração de malhas com os métodos de reconstrução

A aplicação dos métodos de reconstrução de superfície nas nuvens de pontos foi feita com a execução de módulos específicos já implementados em bibliotecas públicas nas linguagens de programação C/C++ e Python, sendo elas PyMeshlab (MUNTONI; CIGNONI, 2021) para o método descrito na seção 4.2.1 e *Point Cloud Library* (RUSU; COUSINS, 2011) para o descrito na seção 4.2.2.

4.2.1 Ball-Pivoting Algorithm

A implementação do método de reconstrução de (BERNARDINI *et al.*, 1999) encontrada no módulo PyMeshlab (MUNTONI; CIGNONI, 2021) para a linguagem de programação Python aceita como entrada um arquivo PLY contendo a nuvem de pontos com as coordenadas X , Y e Z que representa os objetos escaneados.

Os parâmetros necessários para a execução do método são apresentados na Tabela 1, sendo o *ballradius* responsável por definir o raio da bola pivotante em valor percentual com base na diagonal D definida na seção 2.2.2, o valor desse parâmetro limita o tamanho máximo das arestas dos triângulos da malha e tem relação direta com a quantidade de elementos gerados na reconstrução. O parâmetro *clustering* é definido em percentual da bola pivotante e considerado para limitar a menor distância permitida entre dois pontos presentes na malha. Já o parâmetro *creasethr* é o ângulo máximo entre a uma nova aresta a ser inserida na malha e a aresta em que a bola pivotante se encontra. Por fim, o parâmetro *deletefaces* é definido como verdadeiro quando se pretende remover todas as faces já presentes no arquivo de entrada.

Tabela 1 – Parâmetros do método de reconstrução *Ball-Pivoting Algorithm*

Parâmetro	Detalhe
ballradius	Tamanho da bola pivotante (em % de D)
clustering	Distância para mesclar pontos próximos
creasethr	Ângulo mínimo dos triângulos gerados
deletefaces	Desconsiderar os triângulos do arquivo de entrada

Fonte: Autoria própria

Para a geração dos resultados foram seguidos os passos definidos no tutorial do Py-Meshlab¹ sobre como usar essa implementação do método *Surface Reconstruction: Ball Pivoting* (BERNARDINI *et al.*, 1999), com os *scripts* para executar a reconstrução desenvolvidos na

¹ https://pymeshlab.readthedocs.io/en/latest/filter_list.html

linguagem de programação Python. Os quatro passos abaixo definem a sequência implementada para a geração dos resultados com esse método.

1. Criação de um objeto da classe MeshSet do PyMeshLab;
2. Definição do caminho para o arquivo de entrada e carregamento da nuvem de pontos na memória do computador;
3. Chamada ao método de reconstrução *Surface Reconstruction: Ball Pivoting* realizando a passagem dos parâmetros necessários;
4. Exportação da malha gerada (triângulos) para um arquivo do formato PLY.

Com a execução desses passos obtiveram-se as malhas triangulares reconstruídas. A variação na calibração dos parâmetros será detalhada na Seção 5.2.1.

4.2.2 Greedy Projection Triangulation

A implementação do método de reconstrução GP3 de Marton, Rusu e Beetz (2009) encontrada na biblioteca *Point Cloud Library* (PCL) (RUSU; COUSINS, 2011) também é compatível com arquivos do formato PLY. Os parâmetros necessários para a execução desse método de reconstrução são definidos na Tabela 2, sendo eles *radius* que limita o tamanho máximo dos lados dos triângulos, *maxAngle* para limitar o ângulo interno máximo do triângulo, *minAngle* para limitar o ângulo interno mínimo do triângulo, *mult* e *n* são necessários para definir a quantidade de pontos vizinhos selecionados para a criação do plano de projeção usado para realizar a conexão de novos pontos à malha gerada (MARTON; RUSU; BEETZ, 2009, p. 3218).

Tabela 2 – Parâmetros do método de reconstrução *Greedy Projection Triangulation*

Parâmetro	Detalhe
radius	Lado máximo dos triângulos gerados
maxAngle	Ângulo máximo dos triângulos gerados
minAngle	Ângulo mínimo dos triângulos gerados
mult	Multiplicador
k	Número de pontos vizinhos para estimação da normal
n	número máximo de vizinhos a conectar
maxSurfaceAngle (MSAngle)	Ângulo máximo da superfície
normalConsistent	Consistência obrigatória da normal

Fonte: Autoria própria

Já o parâmetro *k* é necessário para o pré-processamento dos vetores normais, conceito definido na Seção 2.2.3, de todos os pontos da malha, uma vez que esse é um requisito obrigatório desse método. Já o parâmetro *maxSurfaceAngle* é configurado para lidar com cantos vivos ou onde dois lados da superfície estão muito próximos um do outro, onde a comparação

entre os pontos é feita pelas suas normais, cuja angulação não deve ser maior que o valor desse parâmetro. O parâmetro *normalConsistent* serve para informar ao algoritmo se as normais dos pontos são consistentes, pois nem todos os métodos de estimação de normais podem garanti-las orientadas de forma consistente.

A implementação para a execução desse método de reconstrução foi feita na linguagem de programação C/C++ seguindo o tutorial da PCL sobre como usar o método *Fast triangulation of unordered point clouds*² (MARTON; RUSU; BEETZ, 2009). Os nove passos a seguir apresentam a sequência de implementação para se obter os resultados da reconstrução.

1. Abertura do arquivo da nuvem de pontos para leitura;
2. Carregamento dos pontos X, Y, Z do arquivo na memória do computador em um objeto da classe `pcl::PointCloud<pcl::PointXYZ>`;
3. Instanciação de uma árvore de busca KD-Tree, usada para a estimação das normais dos pontos, que é um requisito para a aplicação desse método;
4. Chamada do método padrão da PCL para estimação das normais dos pontos;
 - a) O parâmetro *k* é usado para definir a quantidade de pontos vizinhos usados para a estimação das normais;
5. Instanciação de uma árvore de busca KD-Tree, usada para a execução da reconstrução;
6. Instanciação de um objeto da classe `GreedyProjectionTriangulation`;
7. Definição dos parâmetros da reconstrução no objeto criado no item anterior;
8. Execução do método de reconstrução presente no objeto da classe `GreedyProjectionTriangulation` criado anteriormente;
9. Exportação da malha gerada (triângulos) para um arquivo do formato PLY.

4.3 Aplicação da métrica nas malhas geradas

Com os arquivos de malha gerados, seguiu-se para a aplicação da métrica de qualidade. Não foi encontrada uma implementação da métrica $\frac{\rho}{2}$ que satisfizesse as necessidades do presente trabalho, então o autor realizou a implementação da mesma na linguagem de programação C/C++ a fim de se obter em um único arquivo de texto, o valor da razão de aspecto de todos os triângulos.

² https://pcl.readthedocs.io/projects/tutorials/en/master/greedy_projection.html#greedy-triangulation

O primeiro passo da implementação foi a abertura do arquivo de malha para leitura com o auxílio da biblioteca PCL, dada sua compatibilidade com o formato de arquivo PLY. Os dados da malha foram carregados na memória do computador em um objeto da classe PolygonMesh presente na PCL. Em seguida, criou-se um arquivo de texto comum, chamado arquivo de saída, para armazenar os valores da métrica obtidos. Então, foi realizada a iteração entre todos os triângulos da malha e para cada triângulo foram executados os passos descritos abaixo:

1. Cálculo do valor da métrica Radius-ratio, definida por (PÉBAY; BAKER, 2003) e implementada pelo autor;
2. Gravação do valor numérico gerado pela métrica como uma nova linha no arquivo de saída;

Ao final da execução, obteve-se o arquivo de saída com o valor da métrica de todos os triângulos da malha no formato ilustrado na Figura 17.

Figura 17 – Primeiras nove linhas do arquivo gerado pela aplicação da métrica nos triângulos de uma malha.

```
1 1.299898
2 1.014488
3 2.842416
4 1.396719
5 10.505877
6 3.984649
7 1.565110
8 1.886657
9 1.204460
```

Fonte: Autoria própria

5 RESULTADOS

Nesta seção são apresentados os resultados obtidos com a execução dos métodos de reconstrução e a aplicação da métrica nas malhas geradas, assim como a comparação entre elas e a determinação de sua qualidade.

5.1 Ambiente de Trabalho

Os resultados obtidos foram gerados em um computador com as configurações de *hardware* e sistema operacional apresentadas na Tabela 3.

Tabela 3 – Configurações da máquina utilizada

Componente	Configuração
Processador	Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz
Memória RAM	8 GB
Arquitetura	64 bits
Placa de vídeo	NVIDIA GeForce MX250 (GP108M) 2 GB
Sistema Operacional	Linux Mint 20.2 (Uma)

Fonte: Autoria própria

Alguns *softwares* foram usados para a implementação e execução das ferramentas, estes são detalhados na Tabela 4. O Compilador GNU C++ foi a ferramenta utilizada para a geração do programa executável implementado para o cálculo da métrica nas malhas. A utilização da biblioteca PCL se deu para a execução do método GP3 e para a leitura e gravação dos arquivos no formato PLY.

Tabela 4 – Ferramentas utilizadas

Ferramenta	Versão
Compilador GNU C++	9.4.0
Point Cloud Library	1.10.0
Python	3.8.10
Meshlab	2021.10

Fonte: Autoria própria

A linguagem de programação Python foi utilizada para a execução do método de reconstrução BPA já disponível no módulo PyMeshLab e para a geração dos histogramas, com o uso do módulo Matplotlib. Já o *Meshlab* foi utilizado para a visualização das malhas e geração das figuras 18, 19, 20 e 21.

5.2 Calibrações dos métodos de reconstrução

Para que fosse possível determinar qual reconstrução gerada pelos métodos seria usada na comparação dos valores de métrica, foi preciso gerar diversas malhas com calibrações distintas dos parâmetros de cada método de reconstrução, apresentadas na Seção 5.2.1 e na Seção 5.2.2. A variação na calibração dos parâmetros dos métodos foi feita com o objetivo de se obter malhas mais próximas às dos modelos originais apresentados na Seção 4.1, tanto no aspecto visual das malhas, quanto no comportamento dos gráficos da métrica aplicada. As malhas com maior quantidade de triângulos foram consideradas melhores, pois os métodos utilizados apresentaram a presença de buracos na malha, prejudicando assim a sua semelhança com a malha original.

5.2.1 Ball-Pivoting Algorithm

Pelas definições do método BPA apresentadas na Seção 4.2.1, é possível concluir que para se obter boas malhas o parâmetro do raio da bola pivotante deve ser variado até encontrar o que melhor se adapta à distribuição dos pontos amostrados do modelo. Por isso inicialmente foram fixados os valores dos parâmetros *Clustering Radius* e *Crease Treshold* conforme apresentado na Tabela 5, onde na coluna *Crease Treshold* foram definidos os valores de 90° e 180° para esse parâmetro que limita o ângulo máximo de uma nova aresta a ser inserida na malha. Esses valores foram selecionados pois, uma vez que aumentou-se acima de 180° ou foi diminuído abaixo de 90°, a quantidade de triângulos foi prejudicada, ocasionando a aparição de buracos nas malhas geradas. Na coluna *Clustering Radius* da Tabela 5 foram definidos os valores 1%, 5% e 20% para esse parâmetro, pois foi observado que em valores maiores que 20% a quantidade de triângulos decaiu e também aumentou a ocorrência de buracos nas malhas geradas.

Tabela 5 – Parâmetros fixados durante a variação do *Ball radius*

Nome	Clustering Radius	Crease Treshold
Mesh0	20%	180°
Mesh1	20%	90°
Mesh2	5%	180°
Mesh3	1%	180°

Fonte: Autoria própria (2022).

Em seguida variou-se o valor do parâmetro *Ball radius* de 0% a 10% da diagonal D, onde foi possível perceber que em valores maiores que 10% não se obtiveram maiores quantidade de triângulos, então esse foi o valor máximo estabelecido para esse parâmetro nas calibrações consideradas. A Tabela 6 apresenta a quantidade de triângulos obtida nas reconstruções do modelo Bunny, de acordo com a variação do parâmetro *Ball Radius* apresentada na primeira

coluna desta tabela, onde as demais colunas representam uma linha da Tabela 5. Então, para exemplificar, considera-se que há quatro calibrações diferentes na última linha da Tabela 6, todas elas com o raio da bola pivotante definido em 10,00% da diagonal D e na coluna Mesh0 tem-se os parâmetros *Clustering Radius* definido como 20% e *Crease Treshold* em 180° e a quantidade de triângulos é 208, da mesma forma a coluna Mesh1 tem os parâmetros *Clustering Radius* definido como 20% e *Crease Treshold* em 90° e a quantidade de triângulos também 208, já a coluna Mesh2 tem os parâmetros *Clustering Radius* definido como 5% e *Crease Treshold* em 180° com 30680 triângulos e por fim, a última coluna Mesh3 tem os parâmetros *Clustering Radius* definido como 1% e *Crease Treshold* em 180° com 67318 triângulos.

Tabela 6 – Quantidade de triângulos gerados com o *Ball-Pivoting Algorithm* no modelo *Bunny*

Ball Radius	Mesh0	Mesh1	Mesh2	Mesh3
0,01%	0	0	0	0
0,05%	0	0	0	0
0,10%	0	0	0	0
0,15%	16	16	14	14
0,20%	57	57	58	55
0,30%	2238	2233	2264	2270
0,40%	64898	64891	64917	64917
0,50%	70480	70500	70525	70529
0,60%	71229	71227	71314	71320
0,70%	71293	71280	71523	71531
0,80%	70934	70952	71476	71484
0,90%	70528	70457	71348	71356
1,00%	70140	70034	71329	71337
1,50%	67967	67792	70706	70716
2,00%	60081	60208	70016	70029
2,50%	34158	34470	69387	69514
3,00%	29064	30047	68765	69061
3,50%	21020	21155	68029	68713
4,00%	13681	13698	67256	68450
4,50%	11802	11796	66467	68390
5,00%	7110	7103	65676	67679
5,50%	5708	5689	65556	67849
6,00%	5251	5256	65076	68087
6,50%	4963	4952	64488	67487
7,00%	4652	4642	64349	67552
7,50%	4212	4213	63508	67658
8,00%	3525	3519	57343	67774
8,50%	1868	1868	39823	67483
9,00%	1039	1037	34003	67406
9,50%	455	455	31685	67424
10,00%	208	208	30680	67318

Fonte: Autoria própria

Ainda na Tabela 6 é possível perceber que no caso das calibrações da Mesh0 a maior quantidade de triângulos gerados para esse modelo foi com o parâmetro *ballradius* variando en-

tre 0,4% e 2%, comportamento que se repete para as calibrações da Mesh1. Já nas calibrações da Mesh2, foi possível perceber que a quantidade de triângulos gerados também foi alta em outros valores do *ballradius*, com 63508 triângulos gerados no valor 7,5% desse parâmetro, assim como na Mesh3, com 68390 triângulos na linha 4,5%. Mas com o parâmetro *ballradius* definido em 0,7% obtiveram-se as maiores quantidades de triângulos gerados para as calibrações Mesh0, Mesh1, Mesh2 e Mesh3, com a geração de 71293, 71280, 71523 e 71531 triângulos respectivamente, sendo essas calibrações consideradas as melhores por ter uma maior quantidade de triângulos.

Tabela 7 – Quantidade de triângulos gerados com o *Ball-Pivoting Algorithm* no modelo *Dragon*

Ball Radius	Mesh0	Mesh1	Mesh2	Mesh3
0,01%	13911	13722	16830	17635
0,05%	25211	24590	36975	40929
0,10%	321379	321519	384298	407943
0,15%	569906	569682	656298	701664
0,20%	652234	652362	741870	799559
0,30%	657588	657220	758381	837284
0,40%	614535	614277	739300	829422
0,50%	575663	576201	724683	818882
0,60%	541595	543229	711056	808201
0,70%	398494	403701	698142	797084
0,80%	312897	317555	687070	788068
0,90%	267810	269934	676188	780442
1,00%	195822	195979	665526	772601
1,50%	86056	85857	605261	738158
2,00%	51457	51311	549539	713166
2,50%	26966	26833	504742	696052
3,00%	13296	13241	310825	681783
3,50%	6491	6456	240931	670926
4,00%	2871	2854	166500	659370
4,50%	1198	1193	135453	647832
5,00%	526	524	118231	636985
5,50%	244	244	94880	626183
6,00%	130	130	73608	614230
6,50%	105	105	62591	602455
7,00%	62	62	56775	589485
7,50%	53	53	51042	576857
8,00%	52	52	44770	564998
8,50%	45	45	37898	553656
9,00%	31	31	32879	542355
9,50%	31	31	28162	532770
10,00%	0	0	23056	521535

Fonte: Autoria própria

A Tabela 7 apresenta a quantidade de triângulos obtida nas reconstruções do modelo Dragon, de acordo com a variação do parâmetro *Ball Radius* em cada uma das calibrações apresentadas na Tabela 5. Para esse modelo foi possível perceber a ocorrência de malhas

com maiores quantidades de triângulos com o parâmetro *Ball Radius* definido em 0,3% com a geração de 657588 triângulos para a calibração da Mesh0, 657220 na Mesh1, 758381 na Mesh2 e 837284 na Mesh3.

Tabela 8 – Calibrações do método de reconstrução *Ball-Pivoting Algorithm* analisadas

Modelo	Ball radius	Clustering Radius	Crease Treshold	Triângulos
Bunny (Mesh0)	0,70%	20%	180°	71293
Bunny (Mesh2)	0,70%	5%	180°	71523
Bunny (Mesh3)	0,70%	1%	180°	71531
Dragon (Mesh0)	0,30%	20%	180°	657588
Dragon (Mesh2)	0,30%	5%	180°	758381
Dragon (Mesh3)	0,30%	1%	180°	837284

Fonte: Autoria própria

A Tabela 8 traz de forma detalhada as calibrações e respectivas quantidades de triângulos das malhas que tiveram melhores resultados na execução da reconstrução com o método BPA, destacadas em negrito nas tabelas 6 e 7, estas foram as malhas selecionadas para realizar as comparações entre as calibrações do próprio método, assim como a comparação com as malhas geradas pelo método GP3.

5.2.2 Greedy Projection Triangulation

Neste método, ao variar os parâmetros ângulo máximo do triângulo, ângulo mínimo do triângulo, raio e multiplicador foi percebida uma grande variação nos aspectos visuais das malhas geradas, apresentada na Seção 5.3, assim como foi percebida a variação na quantidade de triângulos gerados. O mesmo não aconteceu quando foram variados os parâmetros de ângulo máximo da superfície, número máximo de vizinhos a conectar e número de vizinhos para a estimação da normal.

Tabela 9 – Calibrações do método de reconstrução *Greedy Projection Triangulation* analisadas

Modelo	r (% de D)	maxAng	minAng	mult	k	n	MSA	triangulos
Bunny c)	100	180	1	2,75	20	100	180	71778
Bunny b)	100	90	30	2,75	20	100	180	70534
Bunny d)	11,99	45	0	2,75	20	100	270	68073
Dragon c)	100	180	1	2,75	20	100	180	848925
Dragon d)	11,24	45	0	2,75	20	100	270	806589
Dragon b)	100	90	30	25	20	100	180	756869

Fonte: Autoria própria

Cada linha da Tabela 9 corresponde a uma calibração do método GP3 definida a partir dos seus parâmetros e a última coluna apresenta a quantidade de triângulos gerados, sendo que os registros da tabela estão ordenados do maior para o menor valor dessa coluna em cada modelo.

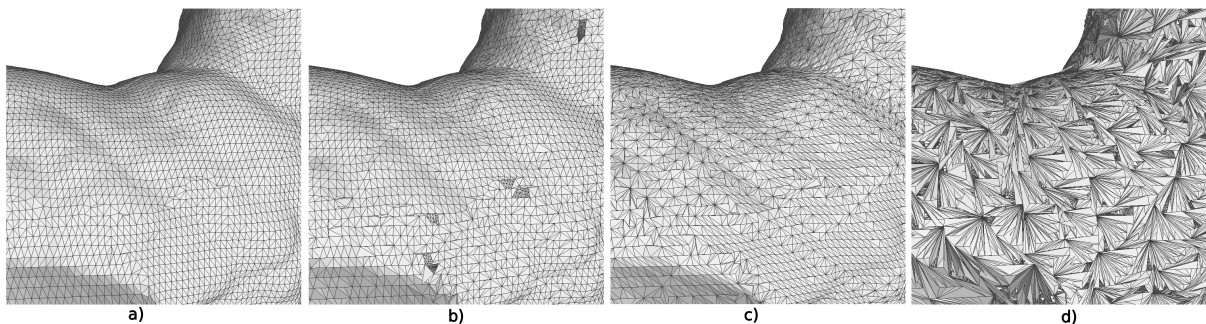
5.3 Características das malhas geradas

A fim de se ter uma noção inicial de como os métodos de reconstrução se comportaram, foram feitas algumas observações nas características visuais das malhas. Um dos problemas encontrados nas malhas de saída geradas pelos métodos, foi a presença de buracos. Também foi percebido que, conforme variaram-se os parâmetros, variou também a presença de triângulos com pior aspecto visual, que são os triângulos com a razão de aspecto distante de 1, ou seja, triângulos muito diferentes do equilátero.

Com o método *Greedy Projection Triangulation*, obtiveram-se os melhores resultados com o parâmetro r definido em 100% da diagonal D , então variaram-se os outros parâmetros em busca da malha com maior quantidade de triângulos e o aspecto visual mais semelhante à malha original.

Com a Figura 18 foi realizada a comparação visual entre as diferentes malhas do modelo Bunny abordadas durante o desenvolvimento do presente trabalho, sendo que em a) é apresentado um recorte da malha original onde é possível observar que não existe a presença de buracos na superfície. Na parte b) da Figura 18 é exibido o mesmo recorte na malha gerada pelo método GP3 com a calibração Bunny b) apresentada na segunda linha da Tabela 9, que tem um total de 70534 triângulos. É possível observar que, apesar dessa malha ter o aspecto dos triângulos semelhantes ao da malha original, o que é considerado bom, há a ocorrência de buracos na malha pelo tom escurecido na superfície do modelo ilustrado, causado pela ausência de triângulos nesses locais que destoam na comparação com a malha original, o que é considerado ruim.

Figura 18 – Malhas do modelo Bunny para comparar visualmente o algoritmo GP3: a) malha original, b) malha similar à original, porém com a presença de buracos, c) malha sem buracos, mas com aspecto visual diferente da malha original, d) malha muito diferente da original e com a presença de buracos.



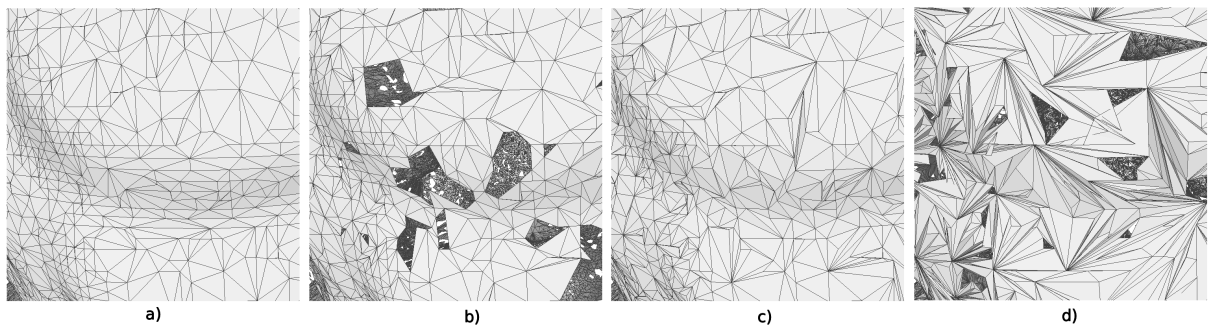
Fonte: Autoria própria

A malha apresentada na parte c) da Figura 18 corresponde ao recorte na malha gerada pelo método GP3 com a calibração Bunny c) apresentada na primeira linha da Tabela 9, que tem mais triângulos, um total de 70534 sem a ocorrência de buracos na malha, o que é considerado bom, porém o aspecto desses triângulos apresenta uma maior diferença visual se comparado com a malha original, da parte a) da figura, o que é ruim dado que a malha original foi conside-

rada a ideal. Por fim, na parte d) da Figura 18 é exibida a malha gerada pela calibração Bunny d) do método GP3 listada na terceira linha da Tabela 9. As características visuais dessa malha possuem grande diferença em relação à malha original, contendo triângulos com aspecto mais alongado e também a presença de buracos na malha, o que a torna a pior malha dentre as quatro apresentadas na Figura 18.

A Figura 19 foi usada para realizar a comparação visual entre as diferentes malhas do modelo Dragon analisadas durante o desenvolvimento do presente trabalho, sendo que em a) é apresentado um recorte da malha original onde é possível observar que não existe a presença de buracos na superfície. Na parte b) da Figura 19 é exibido o mesmo recorte na malha gerada pelo método GP3 com a calibração Dragon b) apresentada na sexta linha da Tabela 9, que tem um total de 756869 triângulos. É possível observar que há a ocorrência significativa de buracos na malha destacados pelo tom escurecido na superfície do modelo ilustrado, causado pela ausência de triângulos nesses locais que fazem com que essa malha seja considerada ruim em comparação com a malha original.

Figura 19 – Malhas do modelo Dragon para comparar visualmente o algoritmo GP3: a) malha original, b) malha similar à original, porém com a presença de buracos, c) malha sem buracos, mas com aspecto visual diferente da malha original, d) malha muito diferente da original e com a presença de buracos.



Fonte: Autoria própria

A malha apresentada na parte c) da Figura 19 corresponde ao recorte na malha gerada pelo método GP3 com a calibração Dragon c) apresentada na quarta linha da Tabela 9, que tem mais triângulos, um total de 70534 sem a ocorrência de buracos na malha, o que é considerado bom, porém o aspecto desses triângulos apresenta uma maior diferença visual se comparado com a malha original, da parte a) da figura, o que é ruim dado que a malha original foi considerada a ideal. Por fim, na parte d) da Figura 19 é exibida a malha gerada pela calibração Dragon d) do método GP3 listada na quinta linha da Tabela 9. As características visuais dessa malha possuem grande diferença em relação à malha original, contendo triângulos com aspecto mais alongado e também a presença de buracos na malha, o que a torna a pior malha dentre as quatro apresentadas na Figura 19.

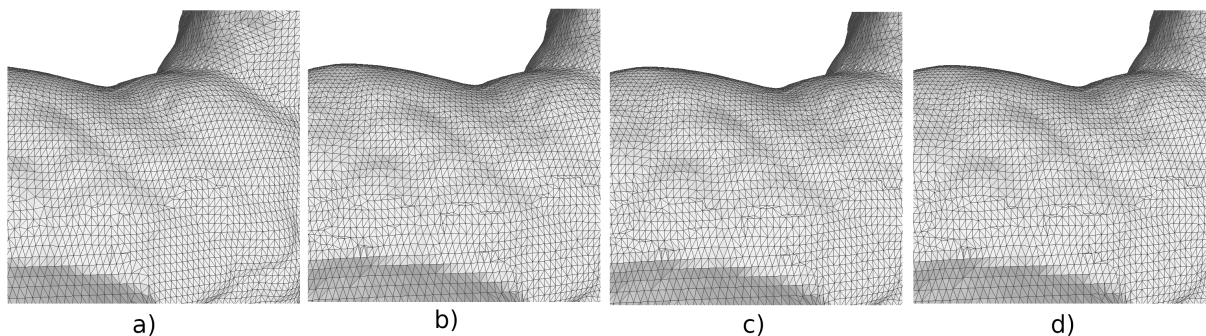
Pelas observações realizadas nas malhas geradas pelo método GP3, foi possível observar que, ao permitir que o método gerasse triângulos com ângulos próximos de 0° ou maiores

que 90°, a qualidade visual da malha gerada piorou, uma vez que aumentou a ocorrência de triângulos com razão de aspecto distante do equilátero se comparado com a malha original.

Para a reconstrução com o método BPA, foram consideradas as calibrações dos parâmetros apresentadas na Tabela 8 e de um modo geral, houve pouca diferença visual das malhas geradas quando comparadas com a malha original dos modelos, porém, ainda assim foi possível notar a presença de buracos em alguns resultados, o que é considerado um ponto negativo.

A Figura 20 apresenta as malhas do modelo Bunny analisadas, onde na parte a) consta o recorte da malha original, que possui 69451 triângulos. Na parte b) da figura é apresentado o recorte da malha gerada pela calibração Mesh0_0.70% com os valores dos parâmetros apresentados na primeira linha da Tabela 8, que possui 71293 triângulos. Na sequência a parte c) apresenta a malha gerada pela calibração Mesh2_0.70% com os valores dos parâmetros apresentados na segunda linha da Tabela 8, que possui 71523 triângulos. Por fim, na parte d) da Figura 20 é apresentada a malha gerada pela calibração Mesh3_0.70% com os valores dos parâmetros apresentados na terceira linha da Tabela 8, que possui 71531 triângulos.

Figura 20 – Malhas do modelo Bunny para comparar visualmente o algoritmo *Ball-Pivoting Algorithm*: a) malha original, b) calibração Mesh0_0.70%, c) calibração Mesh2_0.70%, d) calibração Mesh3_0.70%.



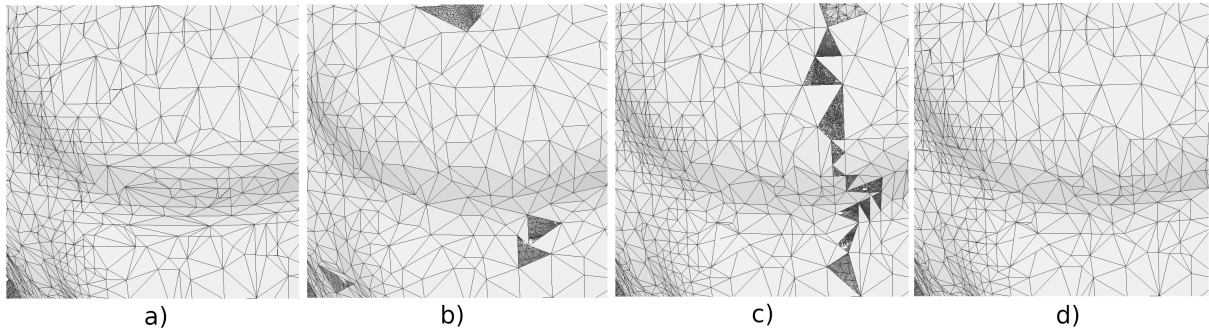
Fonte: Autoria própria

Pela observação das malhas apresentadas na Figura 20, que foram geradas pelo método de reconstrução BPA, foi possível concluir que a diferença visual entre essas malhas é pequena, uma vez que não há a presença de buracos e também o aspecto dos triângulos das malhas geradas se assemelha ao da malha original. Sendo assim, todas as malhas geradas pelo método BPA no modelo Bunny podem ser consideradas boas.

A Figura 21 apresenta as malhas do modelo Dragon analisadas, onde na parte a) consta o recorte da malha original, que possui 871414 triângulos. Na parte b) da figura é apresentado o recorte da malha gerada pela calibração Mesh0_0.30% com os valores dos parâmetros apresentados na quarta linha da Tabela 8, que possui 657588 triângulos. Na sequência a parte c) apresenta a malha gerada pela calibração Mesh2_0.30% com os valores dos parâmetros apresentados na quinta linha da Tabela 8, que possui 758381 triângulos. Por fim, na parte d) da

Figura 20 é apresentada a malha gerada pela calibração Mesh3_0.30% com os valores dos parâmetros apresentados na sexta linha da Tabela 8, que possui 837284 triângulos.

Figura 21 – Malhas do modelo Dragon para comparar visualmente o algoritmo *Greedy Projection Triangulation*: a) malha original, b) melhor malha gerada pelo método, c) malha de qualidade "média", d) Pior malha gerada pelo método.



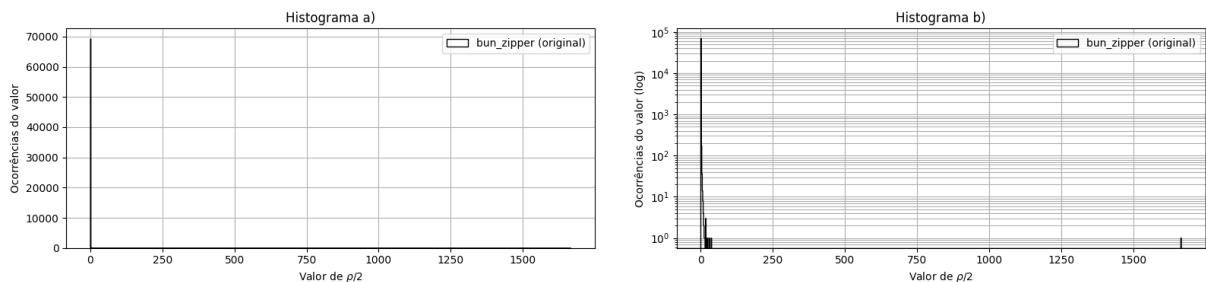
Fonte: Autoria própria

Pelas malhas apresentadas na Figura 21, foi possível observar que o modelo Dragon se comportou de maneira diferente na aplicação do método de reconstrução BPA, onde houve o surgimento de buracos nas malhas, como em b) e c), mas notou-se que o aspecto visual das malhas geradas pela reconstrução é semelhante ao da malha original, sendo assim, como em d) não houve a ocorrência de buracos na malha, esta pode ser considerada a melhor calibração desse método para o modelo Dragon.

5.4 Comparação dos valores obtidos da métrica

Para entender o comportamento da métrica a fim de realizar a comparação dos valores obtidos da sua aplicação em todos os triângulos das malhas originais dos modelos, foram gerados alguns histogramas.

Figura 22 – Histograma dos valores da métrica na malha original do modelo Bunny: a) Gráfico sem escala, b) Gráfico com escala logarítmica no eixo horizontal que apresenta a quantidade de ocorrências.



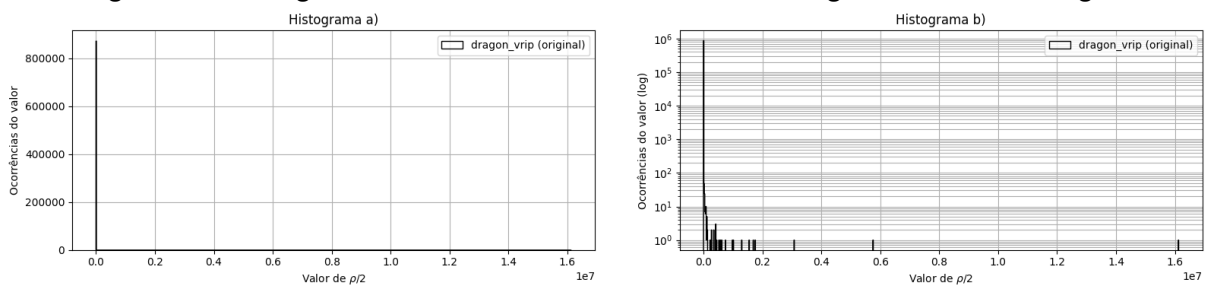
Fonte: Autoria própria

Nas figuras 22 e 23, o histograma apresentado em a) foi gerado com base nos valores da métrica agrupados em 1024 intervalos, onde foi possível perceber um comporta-

rítmico na distribuição dos valores. Então, em b) é exibido o histograma dos mesmos dados com escala logarítmica no eixo y.

O intervalo dos valores obtidos com a aplicação da métrica $\frac{\rho}{2}$ na malha original do modelo Bunny foi [1.000002, 1666.676847] e como é possível observar no Histograma b) da Figura 22, houve uma maior quantidade de ocorrências próximo do valor 0, e quanto maior o valor da métrica, menor é a sua quantidade de ocorrências na malha, por exemplo, o valor máximo do intervalo ocorre somente uma vez. A Figura 22 também mostra que os Histogramas estão com a sua visualização prejudicada pela presença de poucos valores muito distantes dos valores que ocorreram com mais frequência, indicando a presença de *outliers*.

Figura 23 – Histograma dos valores da métrica na malha original do modelo Dragon.



Fonte: Autoria própria

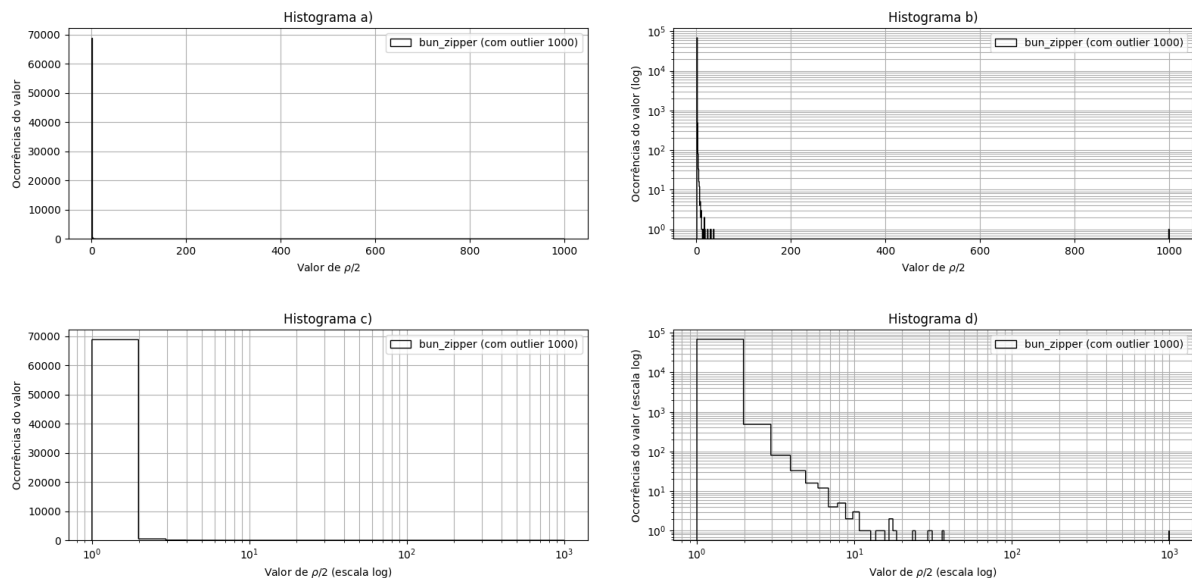
Pela aplicação da métrica $\frac{\rho}{2}$ na malha original do modelo Dragon foi obtido o intervalo dos valores sendo [1.0, 6128519.355163] e como é possível observar na Figura 23, por maior que tenha sido a concentração de ocorrências próximas do valor 0, houveram mais ocorrências de maior valor em comparação com o histograma da qualidade do modelo Bunny, apresentado na Figura 22.

Nos histogramas b) apresentados nas Figuras 22 e 23 foi possível perceber que a frequência de ocorrências tende a ser menor para os maiores valores da métrica e os mais próximos a 1 mostraram-se mais frequentes, o que é um comportamento esperado da métrica *radius-ratio* encontrada no trabalho de (PÉBAY; BAKER, 2003). Inicialmente tentou-se remover os valores discrepantes com o uso da ferramenta BoxPlot, porém não se mostrou eficaz por desconsiderar grande parte dos valores do conjunto: 9% para o modelo Bunny e 16% dos valores para o modelo Dragon. Observado o comportamento da distribuição da métrica no modelo Bunny, concluiu-se que existe um único valor que ultrapassa 1000. Já no modelo Dragon, existem mais ocorrências de triângulos com o valor de métrica que ultrapassam 1000, mas estas representam apenas 0,15% do total. Com base nessas observações, foi estabelecido um valor máximo (teto) durante a obtenção da métrica, fixado em 1000, para eliminar os valores discrepantes a fim de se obter os histogramas com base na tendência da maior parte dos dados.

Os histogramas a), b), c) e d) apresentados na Figura 24 foram gerados com os mesmos dados, porém com escalas diferentes. Esses dados representam os valores da métrica no modelo Bunny agora com o valor de teto, e é possível observar que no Histograma b) o comportamento do gráfico teve uma pequena alteração, se comparado com o Histograma b) da Figura

22, pois da mesma forma existe o único registro do valor máximo, mas agora definido como 1000.

Figura 24 – Histograma dos valores da métrica na malha original do modelo Bunny com o valor máximo definido em 1000. Em a) sem aplicação de escala, b) com escala logarítmica no eixo vertical, c) com escala logarítmica no eixo horizontal e d) com escala logarítmica nos dois eixos.



Fonte: Autoria própria

No Histograma c) da Figura 24 foi aplicada a escala logarítmica no eixo horizontal do gráfico, mas ainda assim a concentração das ocorrências dos valores se apresentou desproporcional nesse eixo com quase a totalidade no primeiro intervalo. Quando a escala logarítmica foi aplicada no eixo horizontal e no eixo vertical, conforme apresentado no Histograma d), o gráfico apresentou os valores da métrica no eixo horizontal e a quantidade de ocorrências no eixo vertical de forma mais distribuída, tendo então uma melhora no comportamento visual do gráfico se comparado com o Histograma a) da mesma figura, por ter uma característica visual que pode ser usada nas comparações entre outros gráficos gerados com essa mesma escala.

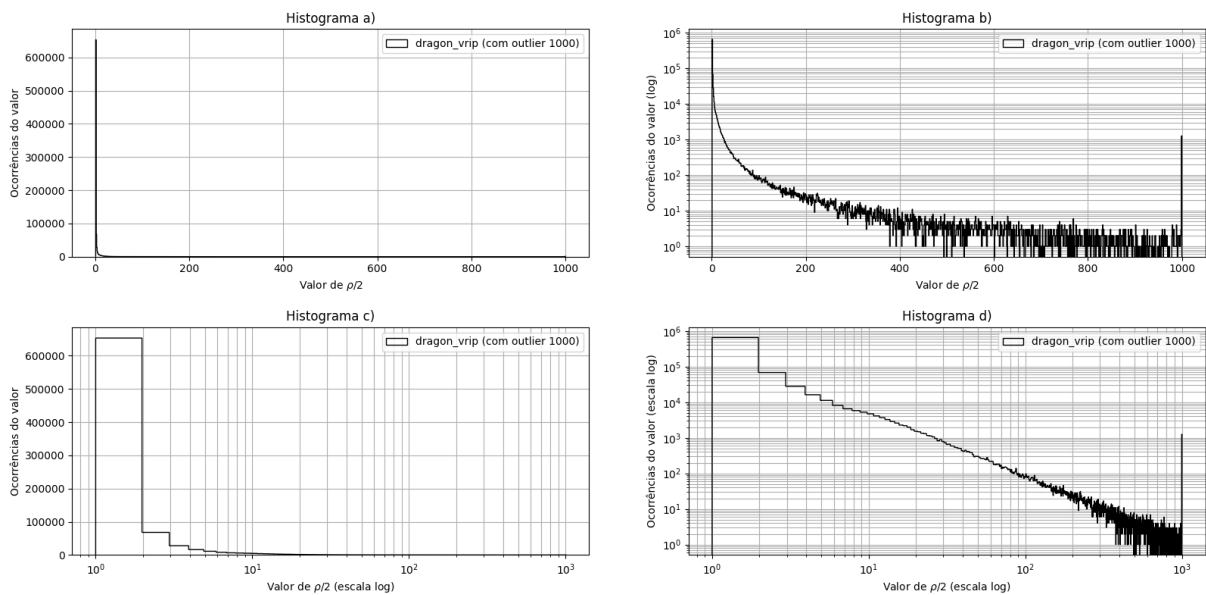
Ao observar o Histograma b), ilustrado na Figura 25, dos valores da métrica $\frac{\rho}{2}$ obtidos em sua aplicação no modelo Dragon com o valor de teto definido, é possível perceber que o comportamento do gráfico alterou de forma considerável em relação ao Histograma b) da Figura 23, com a quantidade de ocorrência dos valores mais distribuída e aparência semelhante a de uma curva, com a escala logarítmica no eixo vertical do gráfico. Também foi possível observar que houveram 1269 ocorrências do valor de teto 1000, correspondendo a aproximadamente 0,15% do total de 871414 triângulos desta malha.

Comparando os histogramas d) das figuras 24 e 25 notou-se que existe uma diferença na distribuição dos valores da métrica entre um modelo e outro, onde no caso do Bunny mesmo depois de aplicar a escala logarítmica nos dois eixos do gráfico, apresenta a ausência de valores

em um determinado intervalo, o que não ocorreu no Dragon, de onde foram obtidos ocorrências melhores distribuídas durante todo o intervalo de [0, 1000].

Os gráficos com maior ocorrência de valores mais próximos de 1 devem ser considerados melhores pelo comportamento da métrica radius-ratio para triângulos equiláteros (PÉBAY; BAKER, 2003), onde se obtém a melhor razão de aspecto com valor 1 (JUNIOR, 2008, p. 99-100).

Figura 25 – Histograma dos valores da métrica na malha original do modelo Dragon com o valor máximo definido em 1000. Em a) sem aplicação de escala, b) com escala logarítmica no eixo vertical, c) com escala logarítmica no eixo horizontal e d) com escala logarítmica nos dois eixos.



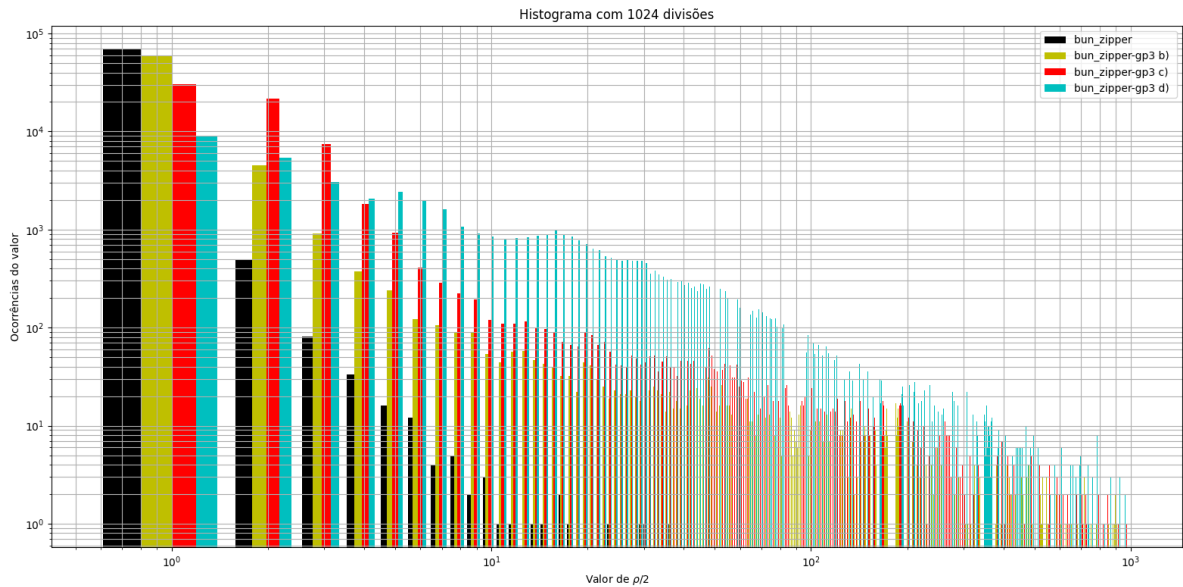
Fonte: Autoria própria

De acordo com as ilustrações dos histogramas apresentadas nesta seção, a aplicação da escala logarítmica nos eixos vertical e horizontal facilitou visualização de como os dados da métrica estão distribuídos de acordo com os triângulos da malha, então esse recurso foi utilizado para a geração das figuras dos histogramas e polígonos de frequências, que apresentam a comparação entre a aplicação da métrica na malha original dos modelos e as malhas reconstruídas pelos métodos Ball-Pivoting Algorithm e *Greedy Projection Triangulation*. São as Figuras 26, 27, 28, 29, 30, 31, 32 e 33.

A Figura 26 apresenta um histograma com 4096 barras lado a lado representando a distribuição de frequência das ocorrências dos valores de métrica, respectivamente a malha original (barras pretas), a calibração com aspecto visual similar à malha original, porém com a presença de buracos b) (barras verdes), a calibração da malha sem buracos, mas com aspecto visual diferente da malha original c) (barras vermelhas) e a calibração com pior aspecto visual d) (barras azuis) do método *Greedy-Projection Triangulation* no modelo Bunny. Observando o gráfico, que tem no eixo horizontal os valores da métrica obtidos e no eixo vertical a quantidade de ocorrências desses valores, é possível perceber que existe uma diferença na distribuição

dos valores da métrica que varia de acordo com as calibrações do método de reconstrução. A Figura 27 apresenta os mesmos valores em uma representação de linha equivalente ao topo de cada barra do histograma da Figura 26, o chamado polígono de frequências.

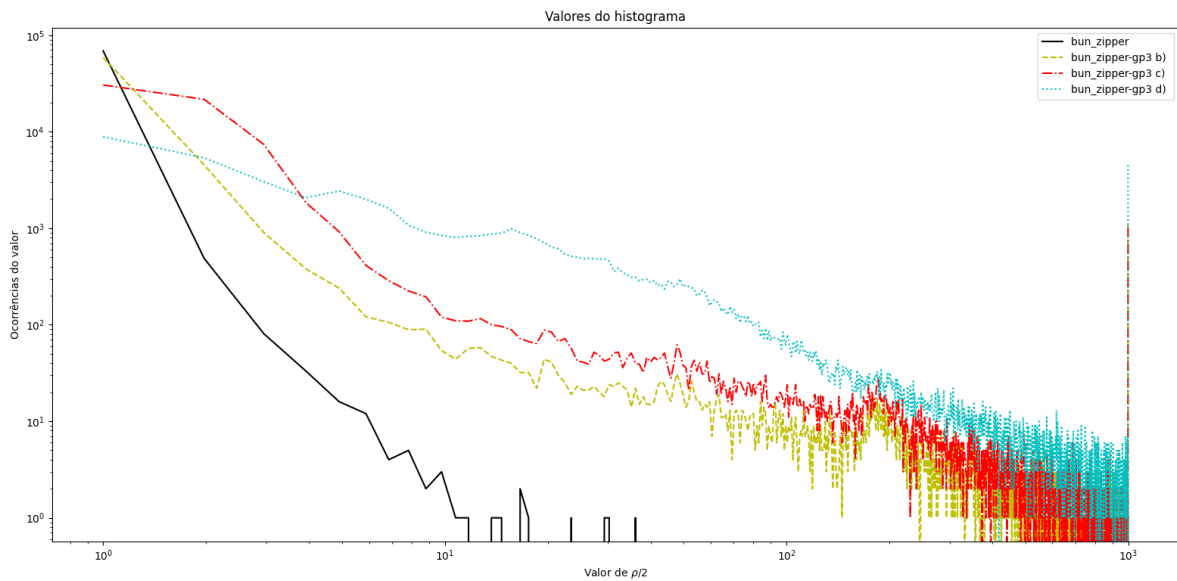
Figura 26 – Histograma dos valores da métrica na malha original e em três calibrações diferentes do método *Greedy Projection Triangulation* para o modelo Bunny.



Fonte: Autoria própria

Assumindo que os melhores valores de métrica são os mais próximos de 1, é possível observar a Figura 27 e concluir que a malha bun_zipper-gp3 d) teve a pior qualidade por ter a sua linha abaixo das demais malhas em valores da métrica próximos de 1.

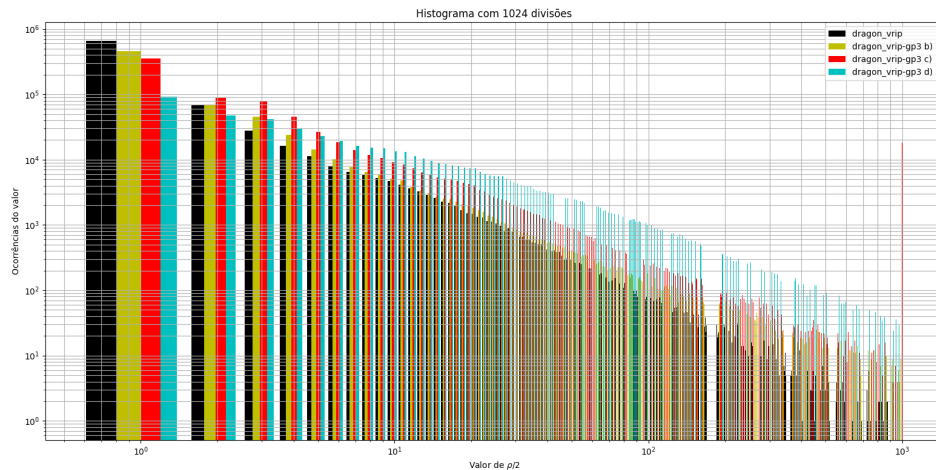
Figura 27 – Linha de valores do histograma na malha original e em três calibrações diferentes do método *Greedy Projection Triangulation* para o modelo Bunny.



Fonte: Autoria própria

Ainda na Figura 27, ao comparar a quantidade de ocorrências em maiores valores da métrica, a linha do gráfico para a malha bun_zipper-gp3 d) está acima das demais e ao avaliar a qualidade visual da malha desta calibração do método na Figura 18, também é possível observar a presença de muitos triângulos com uma grande diferença entre o tamanho de seus lados, ou seja, com a razão de aspecto mais distante de um triângulo equilátero.

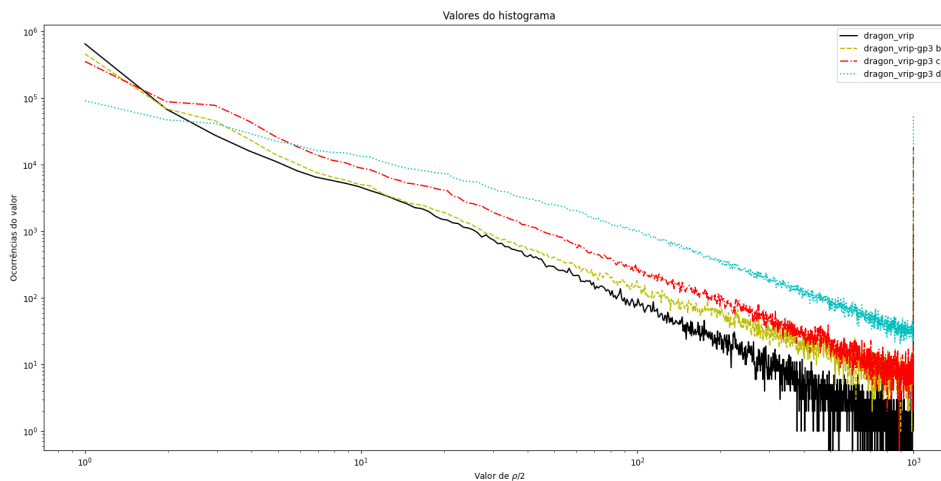
Figura 28 – Histograma dos valores da métrica na malha original e em três calibrações diferentes do método *Greedy Projection Triangulation* para o modelo Dragon.



Fonte: Autoria própria

A Figura 28 apresenta o histograma com 4096 barras lado a lado representando a distribuição de frequência das ocorrências dos valores de métrica, respectivamente a malha original (barras pretas), e as calibrações b) (barras verdes), c) (barras vermelhas) e d) (barras azuis) do método *Greedy-Projection Triangulation* no modelo Dragon.

Figura 29 – Linha de valores do histograma na malha original e em três calibrações diferentes do método *Greedy Projection Triangulation* para o modelo Dragon.

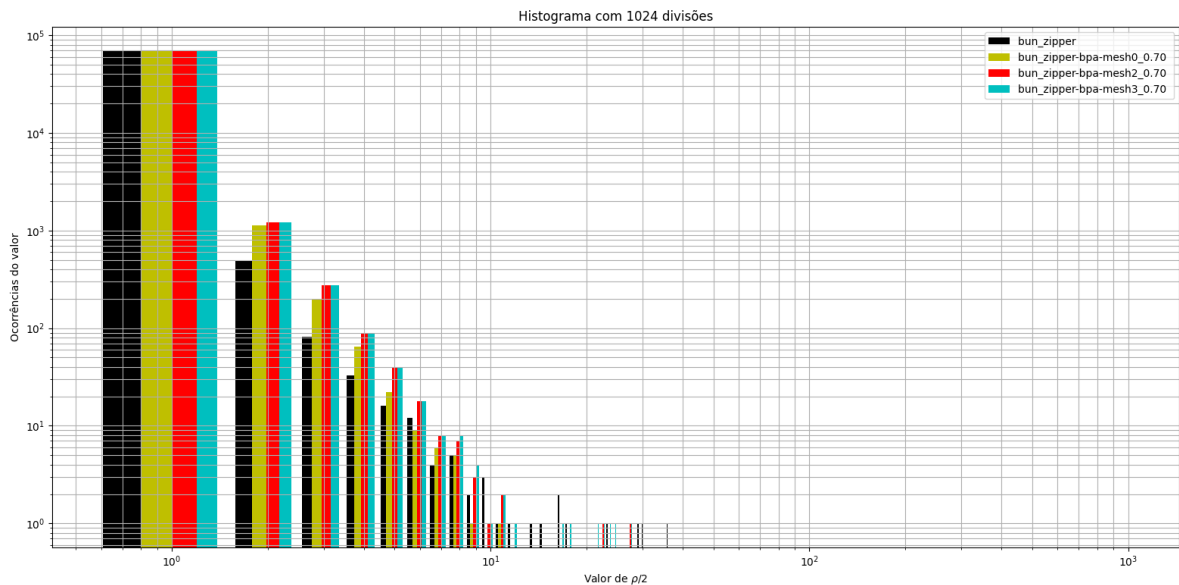


Fonte: Autoria própria

A Figura 29 apresenta os mesmos valores em uma representação de linha equivalente ao topo de cada barra do histograma da Figura 28. Analisando essas duas figuras é possível perceber que também existe a diferença na distribuição dos valores da métrica que varia de acordo com as calibrações do método de reconstrução e, além disso, que nesse modelo a malha original se mostrou melhor por ter menos ocorrências de valores maiores da métrica, assim como mais ocorrências dos valores menores da métrica.

A Figura 30 por sua vez, apresenta o histograma com 4096 barras lado a lado representando a distribuição de frequência das ocorrências dos valores de métrica, respectivamente a malha original (barras pretas), a calibração Mesh0_0.70% em b) (barras verdes), a calibração Mesh2_0.70% em c) (barras vermelhas) e a calibração Mesh3_0.70% em d) (barras azuis) do método *Ball-Pivoting Algorithm* no modelo Bunny. Analisando o gráfico é possível perceber que também existe a diferença na distribuição dos valores da métrica que varia de acordo com as calibrações do método de reconstrução.

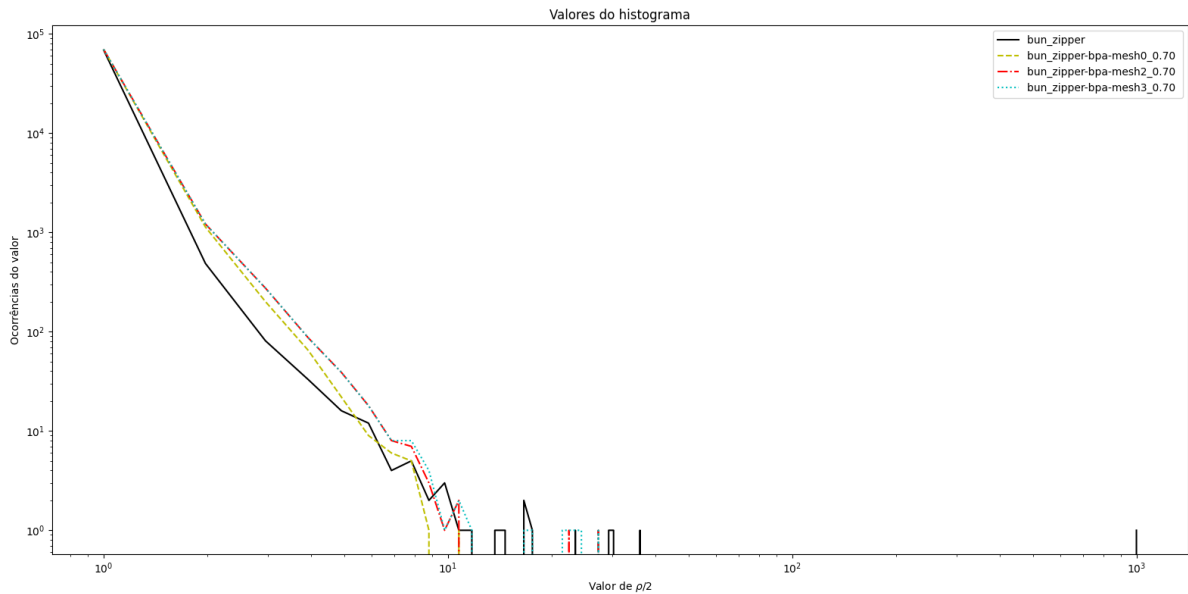
Figura 30 – Histograma dos valores da métrica na malha original e em três calibrações diferentes do método *Ball-Pivoting Algorithm* para o modelo Bunny.



Fonte: Autoria própria

A Figura 31 apresenta os mesmos valores em uma representação de linha equivalente ao topo de cada barra do histograma da Figura 30. Ao analisar essas duas outras figuras, é possível perceber que nas colunas do histograma com maiores valores da métrica, a coluna preta possui uma menor quantidade de ocorrências, sendo assim é possível concluir que a malha original tem melhor qualidade em relação às demais calibrações do método BPA no modelo Bunny. Já a calibração considerada pior nesse modelo é a Mesh3_0.70%, pois nos maiores valores de métrica encontrados, esta possui também a maior quantidade de ocorrências, e a segunda melhor malha, de acordo com a métrica, é a malha Mesh0_0.70% que possui uma quantidade menor de triângulos com maiores valores de métrica comparada à malha Mesh3_0.70%.

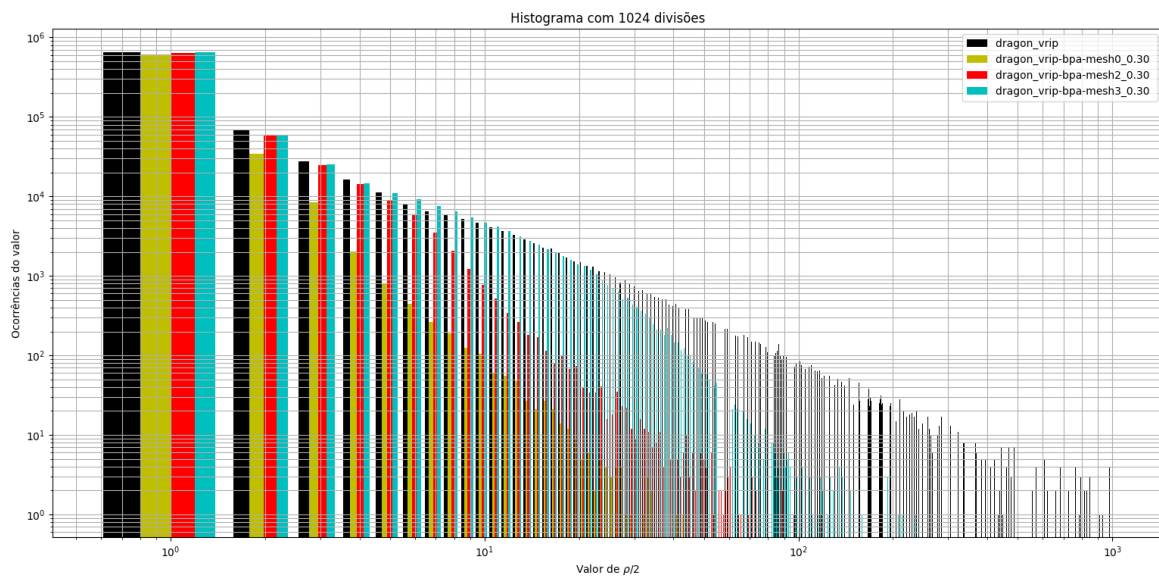
Figura 31 – Linha de valores do histograma na malha original e em três calibrações diferentes do método *Ball-Pivoting Algorithm* para o modelo Bunny.



Fonte: Autoria própria

A Figura 32 apresenta o histograma com 4096 barras lado a lado representando a distribuição de frequência das ocorrências dos valores de métrica, respectivamente a malha original (barras pretas), a calibração Mesh0_0.30% em b) (barras verdes), a calibração Mesh2_0.30% em c) (barras vermelhas) e a calibração Mesh3_0.30% em d) (barras azuis) do método *Ball-Pivoting Algorithm* no modelo Dragon.

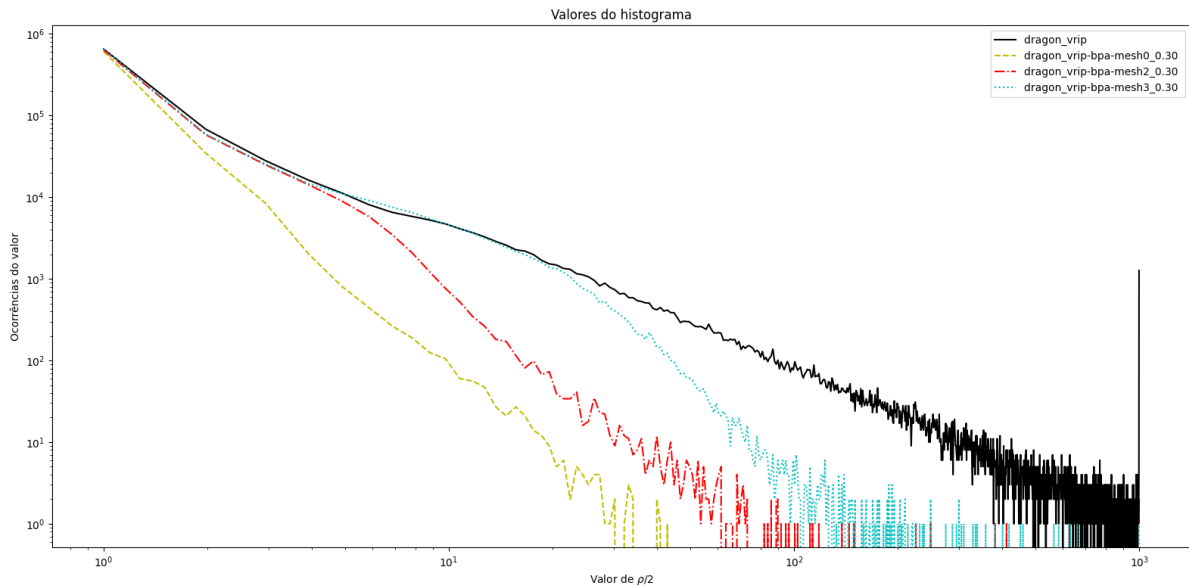
Figura 32 – Histograma dos valores da métrica na malha original e em três calibrações diferentes do método *Ball-Pivoting Algorithm* para o modelo Dragon.



Fonte: Autoria própria

Analisando o gráfico é possível perceber que da mesma forma que os demais resultados, existe a diferença na distribuição dos valores da métrica que varia de acordo com as calibrações do método de reconstrução. A Figura 33 apresenta os mesmos valores em uma representação de linha equivalente ao topo de cada barra do histograma da Figura 32. Nas reconstruções realizadas com o método BPA no modelo Dragon, foi possível perceber uma alteração no comportamento em relação aos demais resultados, sendo que a malha original, representada pela linha preta da Figura 33, contém mais ocorrências de valores maiores da métrica, então conclui-se que a malha original desse modelo possui mais triângulos distantes do equilátero, ou seja, piores.

Figura 33 – Linha de valores do histograma na malha original e em três calibrações diferentes do método *Ball-Pivoting Algorithm* para o modelo Dragon.



Fonte: Autoria própria

Observando o comportamento gráfico da distribuição dos valores da métrica, foi estabelecido um coeficiente único da qualidade da malha como uma média ponderada onde os elementos da média são as frequências F com que cada intervalo do valor da métrica ocorrem e pesos são os próprios valores V da métrica que foram agrupados pela geração do histograma.

$$Q = \frac{F_1 \cdot V_1 + F_2 \cdot V_2 + F_3 \cdot V_3 + \dots + F_{1024} \cdot V_{1024}}{V_1 + V_2 + V_3 + \dots + V_{1024}} \quad (6)$$

Foi realizado o cálculo do coeficiente pela Equação 6 nas malhas geradas a fim de se obter um valor único que represente a qualidade de cada uma. A Tabela 10 apresenta os valores obtidos pela aplicação da métrica na malha original, nas calibrações analisadas do método *Greedy Projection Triangulation* e nas calibrações analisadas do método *Ball-Pivoting Algorithm* para o modelo Bunny.

Segundo a Tabela 10, as reconstruções feitas com o modelo Bunny e o método *Ball-Pivoting Algorithm* se mostraram melhores que as feitas com o *Greedy Projection Triangulation*, porém não foram melhores que a malha original.

Tabela 10 – Valor do coeficiente Q calculado para a malha original do modelo *Bunny* e as malhas reconstruídas pelos métodos *Greedy Projection Triangulation* e *Ball-Pivoting Algorithm*

Malha	Valor de Q
bun_zipper	0.13985790599594736
bun_zipper-gp3 b)	4.130088184111106
bun_zipper-gp3 c)	4.806644575022583
bun_zipper-gp3 d)	16.730630279412125
bun_zipper-bpa mesh0_0.7	0.14296034996343476
bun_zipper-bpa mesh2_0.7	0.14440036391818856
bun_zipper-bpa mesh3_0.7	0.14465797370222783

Fonte: Autoria própria

A Tabela 11 apresenta os valores obtidos no cálculo do coeficiente Q da Equação 6 pela aplicação da métrica na malha original, nas calibrações analisadas do método *Greedy Projection Triangulation* e nas calibrações analisadas do método *Ball-Pivoting Algorithm* para o modelo Dragon. Foi possível observar que, assim como na qualidade visual e no comportamento dos gráficos das malhas, as reconstruções do método *Greedy Projection Triangulation* se revelaram piores em comparação à malha do modelo original, sendo a malha dragon_vrip-gp3 d) a pior delas.

Ainda na Tabela 11 é possível perceber que as reconstruções do método *Ball-Pivoting Algorithm* se mostraram melhores até que a própria malha original. Apesar de as malhas geradas por esse ter uma quantidade menor de triângulos em comparação com a malha original, estes parecem ser de melhor qualidade.

Tabela 11 – Valor do coeficiente Q calculado para a malha original do modelo *Dragon* e as malhas reconstruídas pelos métodos *Greedy Projection Triangulation* e *Ball-Pivoting Algorithm*

Malha	Valor de Q
dragon_vrip	12.153957701380582
dragon_vrip-gp3 b)	54.601626852371766
dragon_vrip-gp3 c)	67.54863345812176
dragon_vrip-gp3 d)	216.74982992778192
dragon_vrip-bpa mesh0_0.3	1.4230100812358097
dragon_vrip-bpa mesh2_0.3	2.072432157265448
dragon_vrip-bpa mesh3_0.3	3.8604747065043576

Fonte: Autoria própria

6 CONCLUSÃO

Este trabalho sugeriu um método de análise e comparação da qualidade de malhas triangulares com base em uma métrica existente na literatura que quantifica a razão de aspecto de cada triângulo de forma independente e individual. Em trabalhos relacionados, obteve-se a informação de que malhas com uma maior quantidade de triângulos equiláteros são processadas mais facilmente por algoritmos de distorção e transformação, então seguiu-se a premissa de que quanto mais triângulos com o aspecto próximo ao do equilátero, melhor é a malha.

Para obter diferentes malhas a fim de determinar quais dentre elas seriam as melhores, foram utilizados dois métodos de reconstrução presentes na literatura e implementados em bibliotecas de *software* públicas, onde a entrada esperada para esses métodos são nuvens de pontos, e a saída gerada são malhas triangulares.

As nuvens de pontos utilizadas foram obtidas de modelos do repositório da Universidade de Stanford disponíveis publicamente. Uma vez que esses modelos já possuem sua malha gerada, esta foi chamada neste trabalho de malha original e usada como referência para comparar com as demais malhas geradas pelos métodos de reconstrução.

Foram estabelecidas diversas calibrações dos métodos de reconstrução, nas quais se observou principalmente a variação na quantidade de triângulos gerados, e nas características visuais das malhas, onde foram identificados buracos sem a presença de elementos e em outros casos, elementos gerados com a razão de aspecto discrepante de triângulos equiláteros. Por isso foram selecionadas as malhas com a maior quantidade de triângulos para a aplicação da métrica.

Para a comparação da qualidade das malhas analisadas, foram gerados histogramas que visaram quantificar a ocorrência e a variação dos valores da métrica obtidos na malha original e nas malhas geradas pela execução dos métodos de reconstrução analisados. A distribuição das ocorrências de valores da métrica se apresentou logarítmica, tornando necessária a aplicação de escala nos eixos horizontal e vertical dos gráficos a fim de se obter uma diferença visual significativa entre os gráficos.

Com os gráficos visualmente distintos, foram plotados em uma mesma figura os histogramas das malhas geradas por diferentes calibrações juntamente com a malha original, com isso foi possível observar que a distribuição dos valores da métrica difere de acordo com os aspectos visuais das malhas, tendo mais ocorrências próximas de 1 as malhas que aparentaram ser melhores e mais ocorrências distantes de 1 as malhas que se demonstraram piores em relação à razão de aspecto de seus triângulos.

Também sugeriu-se o cálculo de um coeficiente único para quantificar a qualidade das malhas com base na distribuição e quantidade de ocorrências nos valores da métrica obtidos. Esse coeficiente se mostrou condizente com a comparação das características visuais da malha e também com a comparação realizada entre os histogramas, onde um valor menor representa uma malha com melhor qualidade e um valor maior, malhas de qualidade inferior.

Por fim, foi feita a comparação dos valores do coeficiente proposto para as diferentes malhas geradas de um mesmo modelo e concluiu-se que o método *Ball-Pivoting Algorithm* apesar de gerar menos triângulos em alguns casos, tende a gerar malhas com melhor qualidade em comparação com o método *Greedy-Projection Triangulation*.

6.1 Trabalhos Futuros

Considerando que os resultados obtidos neste trabalho foram satisfatórios para os métodos de reconstrução utilizados e comparados, sugere-se como estudo futuro a replicação destes resultados nos métodos mais recentes presentes na literatura para a reconstrução de malhas triangulares a partir de nuvens de pontos. Como o método proposto por Digne (2014), que paraleliza a execução do método *Ball-Pivoting Algorithm* visando uma melhor performance, pode apresentar resultados interessantes para a reconstrução de *datasets* maiores como de ambientes urbanos.

Ainda, sugere-se como tema para trabalhos futuros a realização da aplicação de diferentes métricas a fim de observar o comportamento da distribuição dos valores em outras características das malhas além da razão de aspecto dos seus triângulos. Pébay e Baker (2003) definem em seu trabalho outras métricas como a *edge-ratio* τ que possui um valor de aproximadamente 2 para os piores triângulos.

REFERÊNCIAS

- ALVES, G. d. O. A. **A arte rupestre como expressão comunicativa da cultura**. 2006. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2006.
- ANTON, H.; RORRES, C. **Álgebra Linear com aplicações**. 10. ed. Porto Alegre: Brookman, 2012.
- BARBOSA, J. L. M. Geometria euclidiana plana. coleção do professor de matemática. **Rio de Janeiro: Sociedade Brasileira de Matemática**, 2006.
- BATAGELO, H.; MARQUES, B. **Computação Gráfica: Espaço Euclidiano**. 2021. Disponível em: <https://www.brunodorta.com.br/cg/>. Acesso em: 29 nov. 2022.
- BERGER, M. *et al.* A survey of surface reconstruction from point clouds. *In: WILEY ONLINE LIBRARY. Computer Graphics Forum*. [S.l.], 2017. v. 36, n. 1, p. 301–329.
- BERNARDINI, F. *et al.* The ball-pivoting algorithm for surface reconstruction. **IEEE transactions on visualization and computer graphics**, IEEE, v. 5, n. 4, p. 349–359, 1999.
- CURLESS, B.; LEVOY, M.; TURK, G. **The Stanford 3D Scanning Repository**. 1994. Site do laboratório de computação gráfica da Universidade Stanford. Disponível em: <https://graphics.stanford.edu/data/3Dscanrep/>. Acesso em: 14 fev. 2022.
- DEMBSKI, F. *et al.* Urban digital twins for smart cities and citizens: The case study of herrenberg, germany. **Sustainability**, v. 12, n. 6, 2020. ISSN 2071-1050. Disponível em: <https://www.mdpi.com/2071-1050/12/6/2307>.
- DIETRICH, C. A. **Grupos de Arestas: Uma Nova Abordagem para Entender a Qualidade da Malha Gerada pelo Marching Cubes e suas Variantes**. 2008. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, 2008.
- DIGNE, J. An analysis and implementation of a parallel ball pivoting algorithm. **Image Processing On Line**, v. 4, p. 149–168, 2014.
- DOMINGUES, M.; DOMINGUES, J. Estatística exploratória. **Universidade Federal Rural de Pernambuco**, 2010.
- FÁVERO, L. P.; BELFIORE, P. **Manual de análise de dados: estatística e modelagem multivariada com Excel®, SPSS® e Stata®**. [S.l.]: Elsevier Brasil, 2017.
- GHALI, S. **Introduction to geometric computing**. [S.l.]: Springer Science & Business Media, 2008.
- GOIS, J. P. **Reconstrução de superfícies a partir de nuvens de pontos**. 2004. Tese (Doutorado) — Universidade de São Paulo, 2004.
- JAMEI, E. *et al.* Investigating the role of virtual reality in planning for sustainable smart cities. **Sustainability**, Multidisciplinary Digital Publishing Institute, v. 9, n. 11, p. 2006, 2017.
- JUNIOR, V. P. **Métodos implícitos para a reconstrução de superfícies a partir de nuvens de pontos**. 2008. Tese (Doutorado) — Universidade de São Paulo, 2008.
- LENZ, C. **Curvature based surface mesh simplification**. 2019. Tese (Doutorado) — Wien, 2019.

- LINDSTROM, P.; TURK, G. Fast and memory efficient polygonal simplification. *In: IEEE. Proceedings Visualization'98 (Cat. No. 98CB36276)*. [S.l.], 1998. p. 279–286.
- MARTON, Z. C.; RUSU, R. B.; BEETZ, M. On fast surface reconstruction methods for large and noisy point clouds. *In: IEEE. 2009 IEEE international conference on robotics and automation*. [S.l.], 2009. p. 3218–3223.
- MUNTONI, A.; CIGNONI, P. **PyMeshLab**. [S.l.]: Zenodo, 2021.
- NÜCHTER, A.; LINGEMANN, K. **Robotic 3D Scan Repository**. 2016. Site do laboratório de computação gráfica da Universidade Stanford. Disponível em: <http://kos.informatik.uni-osnabrueck.de/3Dscans/>. Acesso em: 14 fev. 2022.
- PÉBAY, P.; BAKER, T. Analysis of triangle quality measures. **Mathematics of Computation**, v. 72, n. 244, p. 1817–1839, 2003.
- RANZUGLIA, G. *et al.* Meshlab as a complete tool for the integration of photos and color with high resolution 3d geometry data. *In: CAA 2012 Conference Proceedings*. Pallas Publications - Amsterdam University Press (AUP), 2013. p. 406–416. Disponível em: <http://vcg.isti.cnr.it/Publications/2013/RCDGS13>.
- RUSU, R. B. *et al.* Towards 3d object maps for autonomous household robots. *In: IEEE. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2007. p. 3191–3198.
- RUSU, R. B.; COUSINS, S. 3d is here: Point cloud library (pcl). *In: IEEE. 2011 IEEE international conference on robotics and automation*. [S.l.], 2011. p. 1–4.
- SKIENA, S. S.; REVILLA, M. A. **Programming challenges: the programming contest training manual**. [S.l.]: Springer, 2003.
- TORI, R.; KIRNER, C.; SISCOOTTO, R. A. **Fundamentos e tecnologia de realidade virtual e aumentada**. [S.l.]: Editora SBC, 2006.
- TRAINA, A. J. M.; OLIVEIRA, M. C. F. de. Apostila de computação gráfica. **Instituto de Ciências Matemáticas e de Computação-Universidade de São Paulo**, 2006.
- TURK, G. **The PLY Polygon File Format**. Stanford University: Stanford, CA, 1994. Disponível em: <http://gamma.cs.unc.edu/POWERPLANT/papers/ply.pdf>. Acesso em: 29 mai. 2019.