

RECONSTRUÇÃO 3D A PARTIR DE IMAGENS E NUVENS DE PONTOS DO SENSOR LIDAR: UMA ANÁLISE COMPARATIVA

Gabriel Castellani de Oliveira, Aurélio Faustino Hoppe – Orientador

Curso de Bacharel em Ciência da Computação
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

gabrielcastellani@furb.br, aureliof@furb.br

Resumo: O processo de reconstrução 3D é um ramo da visão computacional que permite a reconstrução tridimensional de objetos reais. A reconstrução 3D tornou-se uma área bastante estudada e importante para os dias atuais pois permite a visualização e o realce de estruturas volumétricas relevantes. Este artigo discute sobre a reconstrução de superfícies em 3D para objetos reais e, como o tipo de entrada dos algoritmos pode afetar o resultado da reconstrução. Para isso, utilizou-se diferentes algoritmos de reconstrução 3D, um realiza a reconstrução com base em imagem fotográfica enquanto o outro realiza a reconstrução a partir de nuvens de pontos obtidas através da tecnologia LiDAR. Para validar e comparar os resultados obtidos, utilizou-se diferentes objetos em cenários distintos para testar a performance e a qualidade da reconstrução. Conclui-se que os algoritmos de reconstrução com base em nuvens de pontos, Poisson e Ball pivoting, apresentam resultados melhores em relação aos que utilizam imagens 2D.

Palavras-chave: Reconstrução 3D. LIDAR. Nuvem de pontos. Imagens. Algoritmos de Poisson e Ball pivoting.

1 INTRODUÇÃO

Com o surgimento das técnicas de visão computacional, diversas atividades que antes eram feitas através de modelos matemáticos ou por ação manual, como por exemplo, a identificação de suspeitos em um lugar com fluxo de pessoas, contagem de células em uma placa experimental e a identificação de regiões defeituosas em plantações, puderam se tornar automáticas e mais exatas com o auxílio de tecnologias de visão computacional (COSTA, 2021, p. 16). Segundo José e Lopes (2007), a reconstrução 3D é um ramo da visão computacional que tem como objetivo prover informações tridimensionais de ambientes reais, podendo ser do seu volume completo, incluindo seu interior ou somente da superfície.

A grande aplicabilidade da reconstrução 3D a torna uma área bastante pesquisada, importante e promissora. Existem diversos desafios nesta área dentre eles um que se destaca é o custo alto dos aparatos necessários para a pesquisa muitas vezes inacessíveis para pesquisadores independentes e até para algumas universidades (JOSÉ; LOPES, 2007, p. 2).

Segundo José (2008), a aquisição geométrica 3D de um ambiente, objeto ou corpo podem ser obtidas através de varreduras laser, sonares, tomografias, ressonância magnética e fotografias. Além disso, o autor comenta que os sistemas que utilizam a varredura a laser, medem a distância dos objetos através de cálculos entre o atraso do envio do sinal luminoso com a recepção do mesmo sinal pelo sensor. Já os sistemas baseados em fotografias, realizam a reconstrução 3D a partir de uma imagem ou a partir de múltiplas imagens em diferentes ângulos. Segundo José (2008), caso utilize-se múltiplas imagens para realizar a reconstrução 3D de um objeto, marcações de pontos de referência são necessários para identificar as partes em comum para o mesmo objeto em imagens diferentes.

Para Fantí (2016), a digitalização de um objeto, consiste em adquirir as coordenadas cartesianas tridimensionais dos pontos que fazem referência à superfície do objeto. Dessa forma, resultando em uma nuvem de pontos que não necessariamente serão organizados. Segundo Fantí (2016), a reconstrução de superfícies em 3D a partir de nuvens de pontos é um problema difícil, que gera muitos problemas caso os dados obtidos estejam incompletos, com ruídos ou esparsos. De acordo com Laura *et al.* (2018), o projeto *Michelangelo Digital* foi executado no museu *Galleria dell'Accademia*, onde foram obtidas as nuvens de pontos dos objetos através de um *scanner* de alta resolução. A partir das informações capturadas, aplicou-se algoritmos de visão computacional que permitiu a reconstrução 3D de algumas esculturas de *Michelangelo*, considerado como um dos maiores criadores da história da arte. Desta forma, Costa (2021) comenta que o processamento de imagens passa a ter duas grandes categorias, a análise automática de objetos em uma cena por computadores e o aprimoramento das informações já presentes na cena para uma melhor percepção humana.

Neste contexto, a reconstrução tridimensional de um objeto, possibilita a interação do usuário com a estrutura reconstruída, permitindo a rotação e a escala do objeto reconstruído tridimensionalmente (KLENT; INFANTOSI, 2011). Segundo Benalcazar *et al.* (2020), o resultado de uma reconstrução 3D contém informações de profundidade e cor, possibilitando uma visualização mais detalhada do objeto reconstruído. Mescheder *et al.* (2019) mencionam que as abordagens baseadas em aprendizagem para reconstrução 3D se popularizaram mais que os algoritmos estéreos *multi-view* tradicionais, pois conseguem codificar informações prévias sobre o espaço das formas 3D, assim, resolvendo problemas de ambiguidades nas imagens de entrada. Além disso, os autores comentam que os modelos generativos

alcançaram sucessos notáveis na geração de imagens de alta resolução para um domínio 2D, porém, para um domínio 3D ainda não foi alcançado.

Diante desse contexto, este trabalho apresenta um comparativo entre dois algoritmos que possuem o mesmo objetivo, reconstruir superfícies em 3D para objetos reais. Para isso, utilizou-se um algoritmo que realiza a reconstrução 3D a partir de imagens capturadas através de uma câmera fotográfica e algoritmos que realizam a reconstrução 3D a partir de nuvens de pontos obtidas através do sensor Light Detection and Ranging (LiDAR).

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção está separada em 4 subseções. A subseção 2.1 descreve sobre a reconstrução de superfícies em 3D. Na subseção 2.2, discorre sobre a reconstrução e a visualização volumétrica 3D para objetos. Já a subseção 2.3, descreve sobre o funcionamento da tecnologia LiDAR e sua aplicação nas áreas de pesquisa. Por fim, a subseção 2.4 apresenta os trabalhos correlatos em relação ao tema pesquisado.

2.1 RECONSTRUÇÃO DE SUPERFÍCIES EM 3D

Segundo Khatamian e Arabnia (2016), a reconstrução de superfícies em 3D, consiste em um processo que visa reconstruir modelos tridimensionais de objetos reais. Esse processo de reconstrução, necessita de dados de entrada que são fornecidos através de dispositivos de scanner 3D. Geralmente os dados obtidos através do scanner 3D, estão na forma de pontos que ficam espalhados em um espaço 3D. Além disso, Khatamian e Arabnia (2016) comentam que os pontos podem ser não estruturados ou estruturados, sendo a quantidade de informações o diferencial entre as duas categorias. Quando um conjunto de pontos não é estruturado, as únicas informações fornecidas são as coordenadas de cada ponto. Já os conjuntos de pontos estruturados, contém informações geométricas ou topológicas adicionais, além das coordenadas dos pontos.

O processo de reconstrução de superfícies em 3D, necessita de um método apropriado para poder criar um modelo 3D de um objeto real que possa ser visualizável. Geralmente, as superfícies reconstruídas para os objetos, estão em um formato triangularizado, pois os métodos de reconstrução conectam os pontos em triângulos, compartilhando as suas arestas e vértices (KHATAMIAN; ARABNIA, 2016). Antes de iniciar o processo de reconstrução de superfície em 3D, Khatamian e Arabnia (2016) comentam da necessidade de realizar pré-processamentos nos dados de entrada, pois os dados podem conter ruídos ou até mesmo pontos indesejáveis. Além disso, os autores destacam que o resultado da reconstrução da superfície pode mudar drasticamente quando os dados de entrada forem aplicados em diferentes métodos de reconstrução.

As representações das superfícies geradas através dos métodos de reconstrução, podem ser classificadas como explícita ou implícita. Uma superfície explícita é uma prescrição da localização precisa da superfície que geralmente é representada no formato triangularizado. Além disso, existem dois tipos de superfícies explícitas, a paramétricas e trianguladas. Uma superfície paramétrica é a deformação de um modelo primitivo e são topologicamente limitadas pelo modelo inicial, em outras palavras, superfícies mais detalhadas não são facilmente representáveis. Já as superfícies trianguladas, são a versão mais intuitiva para representar uma superfície. Esse tipo de superfície é construído por triângulos conectados a partir dos pontos de entrada. Para esse tipo de reconstrução, métodos para calcular a distância do ponto com o seu vizinho mais próximo no conjunto de pontos são utilizados para construir a conectividade entre os pontos. Já a superfície implícita é definida por uma função cujo um de seus isocontornos é uma estimativa próxima para representar os dados de entrada. No caso, a reconstrução de uma superfície implícita consiste em um processo para encontrar uma função que se ajuste aos dados de entrada. A representação dessa superfície precisa se pós-processada para poder ser visualizada (KHATAMIAN; ARABNIA, 2016).

2.2 RECONSTRUÇÃO E VISUALIZAÇÃO VOLUMÉTRICA 3D

Segundo Silva (2003), o avanço da tecnologia computacional proporcionou o uso de técnicas de Computação Gráfica para representar graficamente dados ou fenômenos naturais. Além disso, a autora comenta que a área da Visualização Científica, auxilia a solução de problemas através de técnicas de visualização de dados, assim, facilitando o entendimento de um problema a partir de representações visuais dos dados relacionados. A autora também ressalta que o avanço da tecnologia, proporcionou para a área da computação, um melhoramento da interação homem-computador.

Diante desse contexto, Silva (2003) comenta que a Visualização Volumétrica (VV) é empregada para explorar a visualização de dados 3D. A autora explica que a ideia da VV é sumarizar informações de um conjunto de dados volumétricos em uma única imagem, de uma forma que possa ser possível visualizar essa imagem como um objeto real. Além disso, Silva (2003) comenta que a Visualização Volumétrica deveria gerar imagens que apresentem fidelidade com os dados originais, permitir a manipulação da representação 3D e gerar a representação em um tempo razoavelmente rápido.

Szeliski (2019) comenta que, ao utilizar técnicas confiáveis, é possível calcular com precisão e gerar um modelo 3D de um ambiente a partir de várias imagens sobrepostas, ou até mesmo, criar modelos de superfícies 3D densas e

precisas. Já Silva (2003), comenta que a interpolação entre imagens de dados 2D correspondentes a um objeto real, gera uma representação de um dado 3D.

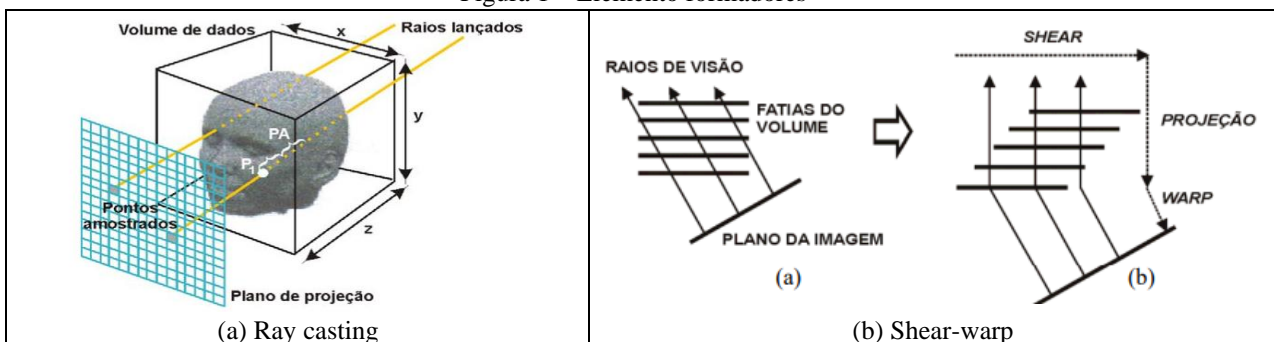
A representação de um dado 3D pode ser categorizada em (i) representações baseadas em *voxel*: são as representações mais comuns para tarefas 3D, (ii) representações baseadas em pontos: são representações dadas por uma nuvem de pontos 3D e (iii) representações baseadas em malha: aplicadas para classificações 3D discriminativas ou para tarefas de segmentação (MESCHEDER *et al.*, 2019).

Segundo Silva (2003), a representação de um dado 3D não necessariamente precisa ser gerada a partir da interpolação de dados 2D, existe também a possibilidade de gerar uma representação a partir de modelos geométricos representados por uma lista de polígonos. A autora comenta que o modelo geométrico é convertido ao espaço discreto 3D, conhecido como voxelização, assim, gerando um conjunto de dados 3D.

Silva (2003) explica que a Visualização Volumétrica possui três principais algoritmos. O primeiro deles é conhecido como Splatting e, utiliza a técnica *object-order*, que inicialmente realiza a projeção dos *voxels* em direção ao plano da imagem, classifica o valor da intensidade de cada *voxel* a partir das funções de transferência de cor e opacidade, e por fim, o algoritmo ilumina os *voxels* de acordo com o vetor de gradiente obtido através das funções de classificação (SILVA, 2003).

Já o segundo algoritmo é conhecido com Ray Casting, esse algoritmo lança raios que calculam os pontos de intersecção entre o raio lançado e o volume de dados, esses raios são disparados do observador em direção a cena. Os pontos obtidos na intersecção, serão utilizados para a amostragem e a classificação. Os elementos básicos formadores desse algoritmo, são o volume dos dados, o plano de projeção, o raio traçado e os pontos de amostragem (SIVA, 2003), conforme representado na Figura 1 item (a).

Figura 1 – Elemento formadores



Fonte: Silva (2003).

Por fim, Silva (2003) comenta que o último algoritmo é conhecido como sheerp-warp, que basicamente mescla as abordagens mencionadas anteriormente, empregando tanto o algoritmo de Splatting quanto o algoritmo de Ray casting. Ainda segundo a autora, na implementação do algoritmo Shear-warp, são usadas estruturas do tipo Run Length Encoding (RLE) para agilizar o processamento dos *voxels* nas fatias e pixels na imagem intermediária, conforme ilustra a Figura 1 item (b). Com o uso desta codificação, os *voxels* transparentes e os pixels já opacos podem ser facilmente ignorados. Geralmente, o algoritmo de Shear-warp é de 5 a 10 vezes mais rápido em comparação ao Ray casting.

2.3 LIDAR

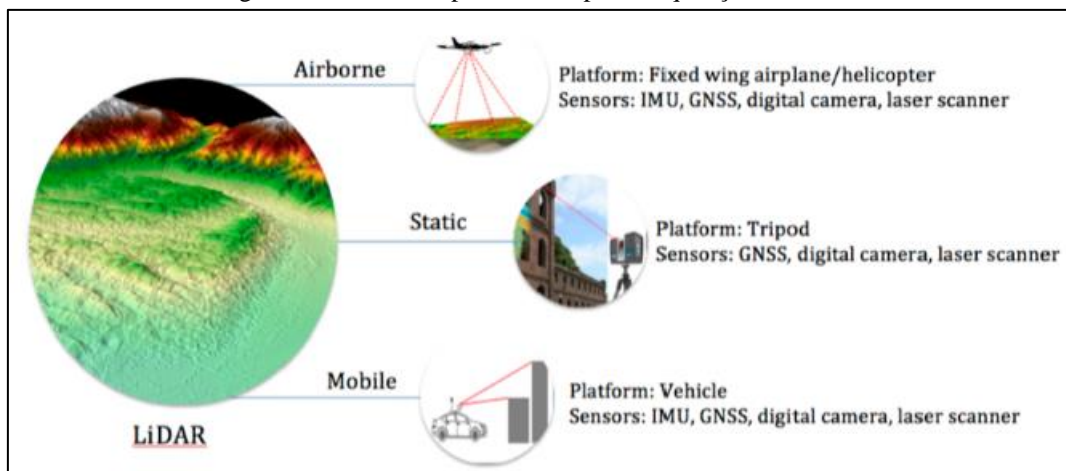
O Light Detection and Ranging (LiDAR), é um termo estabelecido para esta nova tecnologia de sensoriamento remoto. No português, utiliza-se a terminologia de Sistema de Varredura a Laser ou Perfilamento a Laser. Além disso, a nomenclatura Laser Detection and Ranging (LaDAR) também é utilizado, porém, com maior frequência em aplicações militares (GIONGO *et al.*, 2010).

Durante as décadas de 60 e 70, diversas experiências foram realizadas com a tecnologia LiDAR para demonstrar a eficiência e a qualidade da utilização do laser em sensoriamento remoto. Durante o passar dos anos, o LiDAR veio demonstrando um grande potencial em diversas áreas de pesquisa, desde a avaliação de regiões com risco de inundações, até o planejamento urbano de uma cidade (GIONGO *et al.*, 2010, p. 231). Wang, Watkins e Xie (2020), comentam que nos últimos anos, a tecnologia LiDAR ganhou um grande destaque pela sua utilização em veículos autônomos. Com a utilização dessa tecnologia, os veículos autônomos conseguem detectar e prever obstáculos, reconhecer e rastrear objeto, localização e mapeamento simultâneos. Já para a área agrícola, os robôs e aviões agrícolas utilizam a tecnologia LiDAR para ajudar na classificação de espécies de plantas e na análise do seu estado de crescimento.

O LiDAR é uma tecnologia utilizada para medir a distância entre o sensor com a superfície dos objetos. Essa tecnologia funciona como um emissor de pulsos laser que emite com frequência, pulsos em direção aos objetos. Quando o pulso atinge a superfície de um objeto, o mesmo é refletido para o seu emissor. A partir do tempo decorrido entre a

emissão e a captação, o LiDAR calcula uma estimativa da distância do sensor com o objeto (SILVA, 2011, p. 10). Além disso, Giongo *et al.* (2010) comentam que a sua utilização pode ser tanto através de uma plataforma aérea quanto de uma plataforma terrestre, conforme exemplifica a Figura 2.

Figura 2 – Diferentes plataformas para a aquisição dos dados



Fonte: Muhadi *et al.* (2020).

Segundo Giongo *et al.* (2010), o sensor LiDAR possui uma fonte de energia própria, no caso, o laser. Esse laser emite uma radiação eletromagnética que se torna sensível às perturbações atmosféricas, por essa razão, é considerado um método direto na captura dos dados. Além disso, os autores ressaltam uma vantagem na utilização dos sensores de varreduras a laser em comparação com os clássicos sensores passivos, como por exemplo, as imagens de satélites. Os *scanners* a laser não precisam do sol como uma fonte de iluminação, por esse motivo, a análise dos dados não é prejudicada por sombras causadas por outros objetos. Além disso, os autores comentam que os pulsos laser conseguem penetrar em pequenas aberturas no dossel da floresta, assim, identificando o relevo dessas áreas sem problemas.

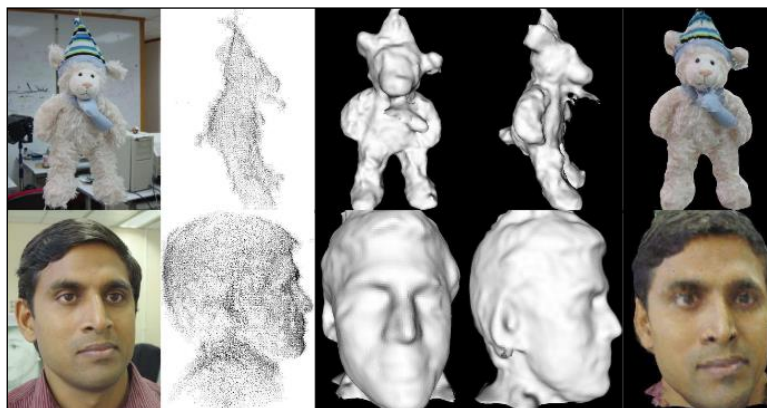
2.4 TRABALHOS CORRELATOS

Nesta seção serão apresentados trabalhos com características semelhantes aos principais objetivos do estudo proposto. A subseção 2.4.1 apresenta o desenvolvimento de uma abordagem para realizar a reconstrução de superfícies, através da propagação de dados estéreo 3D e imagens 2D (ZENG *et al.*, 2004). A subseção 2.4.2 aborda uma rede de ocupação para a reconstrução de objeto 3D (MESCHENDER *et al.*, 2019). Por fim, a subseção 2.4.3 apresentará uma abordagem para reconstruir objetos com base em uma única imagem de entrada (FAN; SU; GUIBAS, 2017).

2.4.1 Surface Reconstruction by propagating 3D stereo data in multiple 2D images

Zeng *et al.* (2004) construíram um *framework* para reconstrução de superfície através da propagação de dados estéreo 3D utilizando várias imagens 2D. O *framework* analisa o ciclo de propagação da superfície, explorando a integração dos pontos estéreo e as imagens calibradas. Segundo os autores, inicialmente instancia-se uma lista de pontos estéreo 3D que serão utilizados para realizar a propagação da superfície. Estes pontos são computados a partir de um conjunto de imagens de entrada. Para cada objeto, os autores coletaram em torno de 30 imagens ao redor do objeto a ser reconstruído, conforme exemplificado na Figura 3.

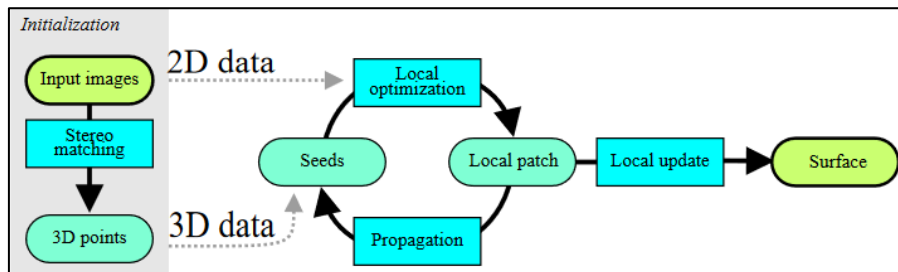
Figura 3 – Conjunto de entrada



Fonte: Zeng *et al.* (2004).

Zeng *et al.* (2004) destacam que o ciclo de propagação da superfície realiza correções da superfície com base nos pontos 3D obtidos. A Figura 4 demonstra que este ciclo possui três etapas: (i) seleção do ponto 3D para a propagação, (ii) geração de um remendo a partir do ponto escolhido e (iii) geração de novos pontos a partir do remendo criado. Segundo os autores, essas etapas serão executadas para cada ponto da lista, fazendo com que a superfície cresça até atingir o resultado desejado. O ciclo de propagação termina quando não existirem mais pontos disponíveis a serem propagados.

Figura 4 – Estrutura da abordagem



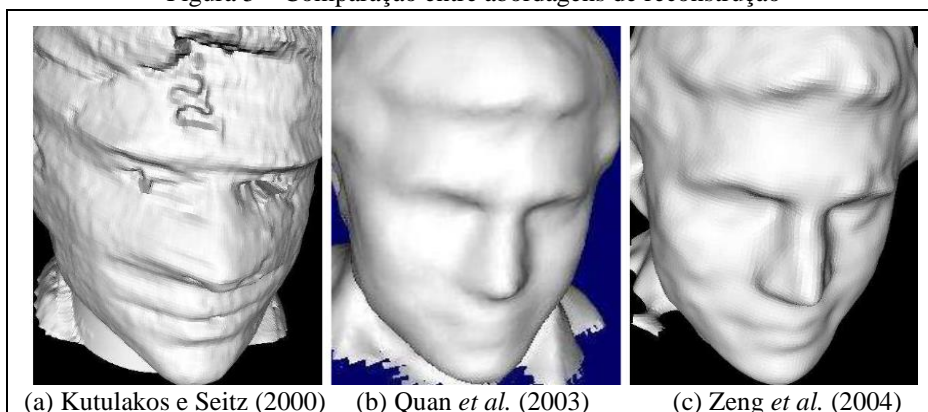
Fonte: Zeng *et al.* (2004).

Segundo Zeng *et al.* (2004), a primeira etapa da propagação seleciona o melhor ponto da lista para servir como referência para o remendo. Para isso, os autores utilizam o critério π para avaliar se o ponto escolhido é a melhor opção para a propagação, no qual os pontos iniciais da lista são considerados como pontos confiáveis. A partir do ponto escolhido, a segunda etapa da propagação gera um remendo ao redor do ponto escolhido anteriormente, dessa forma, o algoritmo estende a superfície existente. Por fim, a última etapa gera novos pontos a partir do remendo anterior, assim, garantindo a propagação da superfície. Os autores comentam que a localização de um novo ponto é determinada pelos seguintes aspectos: (i) qualidade do remendo, (ii) qualidade de correspondência, (iii) regularidade da superfície e (iv) eficiência da propagação. Além disso, os autores comentam que a quantidade de novos pontos é inspirada na configuração de malha triangular, onde a distância angular entre dois pontos vizinhos fica entre os valores apresentados na Equação 1.

$$d = \left[\frac{2\pi}{5}, \frac{2\pi}{7} \right] \quad (1)$$

Zeng *et al.* (2004) aplicaram o *framework* em três conjuntos de entrada. No primeiro conjunto tem-se a representação de um brinquedo, no qual os autores examinaram a exatidão e a robustez da reconstrução. Os dois últimos conjuntos são representações de rostos humanos. Percebe-se a partir da Figura 5 que o primeiro rosto possui mais textura do que o segundo. Os autores comentam que o resultado da reconstrução da superfície obteve para os dois conjuntos praticamente a mesma qualidade.

Figura 5 – Comparação entre abordagens de reconstrução



Fonte: Zeng *et al.* (2004).

Zeng *et al.* (2004) também realizaram um comparativo entre o método de escultura espacial (KUTULAKOS; SEITZ, 2000) que utiliza apenas informações de imagens 2D para a reconstrução da superfície, o método de conjunto de níveis (QUAN *et al.*, 2003) e a abordagem desenvolvida. Os três métodos utilizam informações 3D e 2D para a reconstrução, e o resultado dessa comparação pode ser observado na Figura 5. Os autores destacam que o método de escultura espacial (KUTULAKOS; SEITZ, 2000) perde muitos detalhes do rosto, por conta de usar apenas dados 2D para a reconstrução. Já o resultado do método de conjunto de níveis (QUAN *et al.*, 2003) utiliza as mesmas informações de entrada que a abordagem desenvolvida pelos autores. Porém, (QUAN *et al.*, 2003) aplicam um método de definição de nível que suaviza a superfície, perdendo detalhes geométricos. Contudo, o método desenvolvido por Zeng *et al.* (2004) conseguiu gerar um resultado sem perder os detalhes e as características do objeto de entrada, assim, gerando a superfície com precisão e robustez.

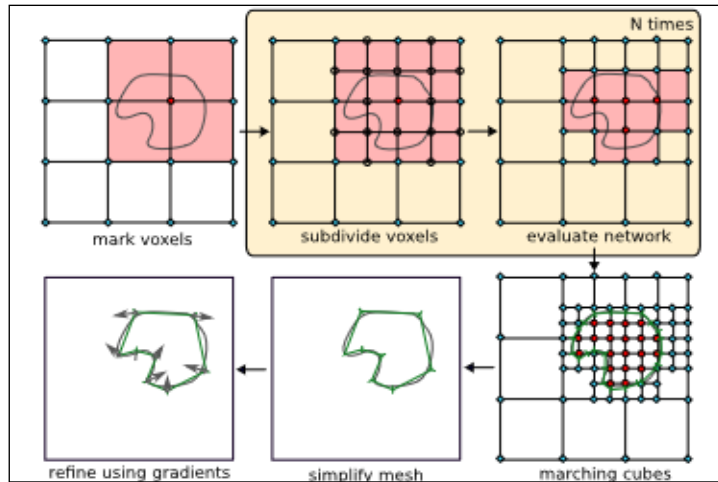
2.4.2 Occupancy Networks: Learning 3D reconstruction in function space

Mescheder *et al.* (2019) propuseram uma abordagem que não necessita de um modelo do objeto de entrada como referência. Tal abordagem utiliza aprendizado profundo para obter uma representação mais expressiva, podendo ser integrado com um *pipeline* de aprendizado de ponta a ponta. Além disso, os autores desenvolveram uma função de ocupação 3D contínua para reconstrução de objetos 3D, sendo equivalente a uma rede neural para classificação binária, mas com o objetivo de decidir implicitamente as fronteiras das superfícies de um objeto.

Segundo Mescheder *et al.* (2019), a função de ocupação do objeto 3D é denotada pelo resultado da função $o : R^3 \rightarrow \{0, 1\}$, no qual os autores aproximam essa função com uma rede neural que atribui para cada local $\rho \in R^3$ uma probabilidade de ocupação entre 0 e 1. Além disso, os autores enfatizam a necessidade de condicioná-la na entrada quando forem aplicar para reconstruções de objetos 3D com base em observações (imagem, nuvem de pontos etc.). Para realizar esse processo de condicionamento, Mescheder *et al.* (2019) utilizaram uma equivalência funcional simples: representada por uma função que recebe um par $(p, x) \in R^3 \times X$ como entrada e produz um número real como saída. Essa representação pode ser parametrizada por uma rede neural f_θ , que recebe um par (p, x) como entrada e gera uma probabilidade de ocupação. Os autores chamam a rede $f_\theta : R^3 \times X \rightarrow [0, 1]$ de rede de ocupação.

Mescheder *et al.* (2019) também desenvolveram um algoritmo hierárquico Multiresolution IsoSurface Extraction (MISE), que utiliza a rede de ocupação para extrair uma isosuperfície correspondente a uma observação. Inicialmente esse algoritmo discretiza o espaço volumétrico e avalia a rede de ocupação $f_\theta(p, x)$ para todos os p da grade, marcando todos os pontos da grade como ocupados se o resultado da ocupação for maior ou igual a 0 ou 1. Já os demais pontos que não entraram nessa condição são marcados como desocupados. Em seguida, o algoritmo marca como ativo todos os *voxels* que possuem pelo menos dois pontos adjacentes com diferentes previsões de ocupação. Neste mesmo momento, os *voxels* ativos são subdivididos em 8 *subvoxels*, gerando novos pontos na grade de ocupação que posteriormente serão avaliados. Por fim, o algoritmo repete essas etapas até alcançar a resolução desejada. Essa estrutura pode ser observada na Figura 6, os pontos ocupados estão marcados com círculos vermelhos, os pontos desocupados em diamantes cianos, os *voxels* ativos em vermelho claro e os novos pontos marcados com círculos vazios.

Figura 6 – Algoritmo MISE



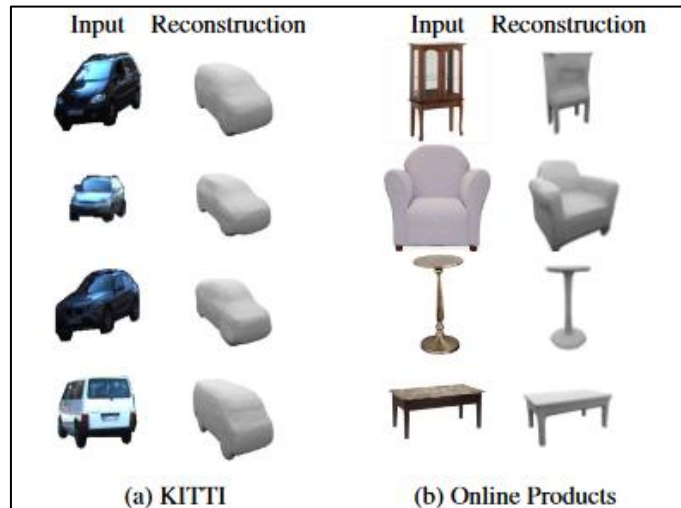
Fonte: Mescheder *et al.* (2019).

Após obter a malha resultante das operações descritas acima, Mescheder *et al.* (2019) apontam que a malha passa por um processo de extração, que inicialmente se aplica o algoritmo Marching cubes (LORENSEN; CLINE, 1987) que cria modelos triangulares de superfícies constantes a partir de dados 3D e, em seguida, a malha extraída passa pelo algoritmo Fast-Quadric-Mesh-Simplification (GARLAND; HECKBERT, 1998) que simplifica a superfícies poligonais. Além disso, os autores também utilizaram informações de gradientes de primeira e segunda ordem para realizar um refinamento da malha de saída. Para isso, os autores aplicaram pontos p_k aleatórios em cada face da malha de saída e minimizaram a perda através da equação descrita na Equação 2. A função $n(p_k)$ denota o vetor normal da malha em p_k e λ é definido como $\lambda = 0.01$. Para a minimização do segundo termo da equação, Mescheder *et al.* (2019) utilizaram informações de gradiente de segunda ordem que pode ser aplicada usando Double-Backpropagation (DRUCKER; CUN, 1992). Os autores ressaltam que essa última etapa remove os artefatos da discretização do algoritmo Marching cubes (LORENSEN; CLINE, 1987) e que não seria possível apenas com uma representação baseada em *voxel*.

$$\sum_{k=0}^k (f_0(p_k, x) - t)^2 + \lambda \left\| \frac{\nabla_p f_0(p_k, x)}{\|\nabla_p f_0(p_k, x)\|} - n(p_k) \right\|^2 \quad (2)$$

Mescheder *et al.* (2019) aplicaram a rede de ocupação nos conjuntos de dados KITTI (GEIGER *et al.*, 2013) e Online Products (SONG *et al.*, 2016), para verificar o comportamento da rede com dados reais. Os autores ressaltam que para os dados KITTI (GEIGER *et al.*, 2013) foi necessário aplicar a máscara de instância (ALHAIJA *et al.*, 2018) para cortar as regiões pertencentes a um carro, aplicando em seguida a rede de ocupação sobre as imagens. Para o conjunto de dados do Online Products (SONG *et al.*, 2016), os autores aplicaram o mesmo modelo pré-treinado. Na Figura 7 pode-se ver o resultado da rede de ocupação para os conjuntos de dados KITTI (GEIGER *et al.*, 2013) e Online Products (SONG *et al.*, 2016).

Figura 7 – Resultados dos testes



Fonte: Mescheder *et al.* (2019).

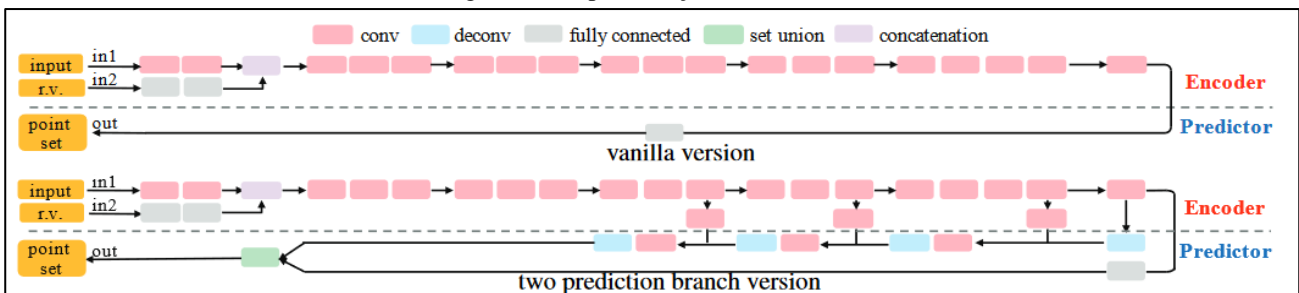
Mescheder *et al.* (2019) concluem que a rede de ocupação consegue gerar reconstruções semelhantes ao objeto de entrada, gerando malhas realistas de alta resolução. Além disso, os autores comentam que a partir dos experimentos realizados, foi possível observar que a rede de ocupação pode ser utilizada para os aprendizados supervisionados e os não supervisionados.

2.4.3 A point set generation network for 3D object reconstruction from a single image

Fan, Su e Guibas (2017) utilizaram redes generativas para realizar a reconstrução 3D com base em uma única imagem de entrada. Os autores destacam que a rede estima a profundidade das partes visíveis do objeto na imagem. Fan, Su e Guibas (2017) também ressaltam que a representação por nuvem de pontos não é a ideal para a reconstrução 3D, porém, possui uma estrutura simples e uniforme para o aprendizado.

De acordo com Fan, Su e Guibas (2017), construir uma rede generativa para uma representação em pontos é um processo desafiador, pois a representação é desordenada e possui ambiguidades. Para isso, os autores desenvolveram uma arquitetura composta por um gerador de conjunto de pontos, função de perda para comparação do conjunto de pontos e a modelagem da incerteza da verdade fundamental (partes invisíveis do objeto), conforme exibe a Figura 8.

Figura 8 – Representação da estrutura



Fonte: Fan, Su e Guibas (2017).

Segundo Fan, Su e Guibas (2017), a primeira parte da arquitetura consiste em uma rede para representar estruturas e estatísticas dos dados geométricos, tendo dois estágios:

- codificador: mapeia a imagem de entrada e um vetor aleatório em um espaço de incorporação;
- preditor: emite uma matriz $N \times 3$, onde cada linha contém as coordenadas de um ponto.

Fan, Su e Guibas (2017) apontam que o *design* do ramo preditor foi modificado para suportar superfícies grandes e lisas, pois a estrutura do *vanilla version* não faz uso completo das estatísticas geométricas. Esse novo *design* possui dois ramos preditores paralelos, sendo um totalmente conectado e o outro de deconvolução. Segundo os autores, o ramo totalmente conectado continua prevendo pontos N_1 e o ramo de deconvolução prevê uma imagem com três canais de tamanho $H \times W$, e os valores de cada pixel representam a coordenada de um ponto.

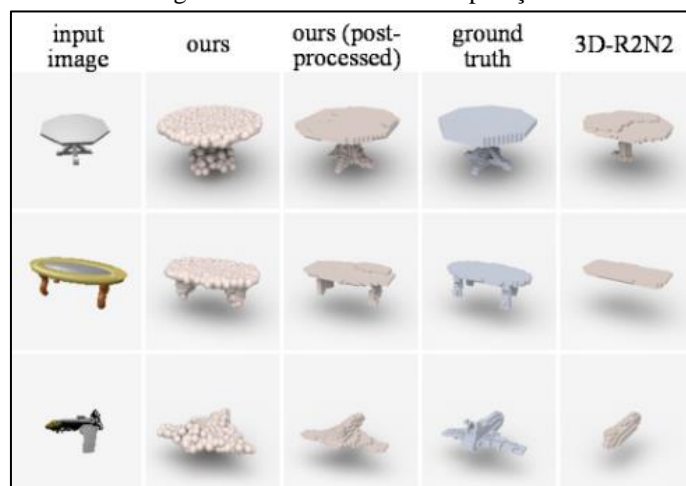
Fan, Su e Guibas (2017) descrevem que a segunda parte da arquitetura é uma função de perda, que serve para comparar a nuvem de pontos prevista com a parte invisível do objeto gerado. Além disso, os autores listaram três condições para definir uma distância adequada para conectar a uma rede neural: (i) diferenciável em relação às localizações dos pontos, (ii) eficiente para computar e (iii) robusto para um número pequeno de pontos discrepantes. Com base nessas condições, os autores propuseram duas funções:

- a) chamfer distance: para cada ponto, a função localiza o vizinho mais próximo no outro conjunto e soma as distâncias ao quadrado;
- b) earth Mover's distance: o cálculo da função em si é muito custoso para um aprendizado profundo. Para isso, os autores implementaram um esquema de aproximação desenvolvida originalmente por Bertsekas (1985).

De acordo com Fan, Su e Guibas (2017), a terceira parte da arquitetura é responsável por realizar a reconstrução da parte invisível do objeto, nessa parte da arquitetura os autores encontraram problemas de regressão e ambiguidades. Para modelar a melhor representação do objeto oculto, os autores aplicaram o método Min-of-N que calcula o mínimo de n distâncias.

Nos experimentos, Fan, Su e Guibas (2017) utilizaram o conjunto de dados ShapeNet (CHANG *et al.*, 2015) que possui um grande volume de modelos de objetos 3D, nesse conjunto de dados os autores preferiram escolher imagens que possuem uma iluminação simples. Em cada modelo, os autores normalizam o hemisfério delimitador e alinham seu plano base. Em seguida, o modelo da imagem 2D é renderizado utilizando a fórmula de sombreado de Blinn-Phong. Fan, Su e Guibas (2017) compararam a abordagem desenvolvida com o 3D-R2N2 (CHOY *et al.*, 2016) que utiliza aprendizado profundo para reconstruir representações 3D a partir de imagens com visualização única. As comparações podem ser visualizadas na Figura 9.

Figura 9 – Resultados da comparação



Fonte: Fan, Su e Guibas (2017).

Com base nos resultados obtidos, Fan, Su e Guibas (2017) relatam que a abordagem 3R-R2N2 (CHOY *et al.*, 2016) geralmente não consegue reconstruir as características finas de um objeto, gerando uma reconstrução incompleta do objeto de entrada. Já a abordagem dos autores, conseguiu reconstruir o objeto com suas características finas e as partes ocultas da imagem a partir de uma visão única.

3 DESCRIÇÃO DO PROCESSO DE RECONSTRUÇÃO 3D

A implementação da reconstrução de superfícies em 3D pode ser sumarizada a partir da Figura 10, no qual dividiu-se o processo em três partes. A seção 3.1 descreve a montagem de um ambiente controlável para a aquisição das imagens 2D através da câmera fotográfica e das nuvens de pontos através do sensor LiDAR. A seção 3.2 discorre sobre os tratamentos que foram necessários tanto nas imagens 2D quanto nas nuvens de pontos para serem utilizáveis nos algoritmos de reconstrução 3D. Por fim, a seção 3.3, apresenta os algoritmos utilizados para reconstruir as imagens e as nuvens de pontos em um objeto tridimensional.

Figura 10 – Sumarização da implementação



Fonte: elaborado pelo autor.

3.1 PREPARAÇÃO DO AMBIENTE E CAPTURA DOS OBJETOS

Inicialmente, preparou-se um ambiente controlado de captura dos objetos. Os objetos a serem capturados foram posicionados no centro de uma mesa de madeira com aproximadamente 78 centímetros de comprimento e 38 centímetros de largura. A mesa foi revestida com uma toalha branca para esconder o aspecto amadeirado e as deformidades. Ao redor da mesa, posicionou-se um Led Yongnuo YN9.000 e dois Led Yongnuo YN600 para emitir uma luminosidade no ambiente, ambos os equipamentos podem ter a sua intensidade de luminosidade configurável e controlável através do próprio dispositivo (Figura 11 item A). O equipamento Led Yongnuo YN9.000 foi direcionado para o centro da mesa, onde o objeto a ser capturado estava posicionado. Já os equipamentos Led Yongnuo YN600, um foi direcionado para o teto (Figura 11 item B), enquanto o outro foi direcionado para a parede (Figura 11 item C). Dessa forma, criou-se um ambiente que a sua intensidade de luminosidade poderia ser controlada. Após a preparação da mesa e o posicionamento dos equipamentos de luminosidade, adicionou-se alguns marcadores para informar as distâncias que os objetos deveriam ser capturados. Para isso, utilizou-se um pedacinho de massinha de modelar para posicionar em distâncias de 20, 40 e 80 centímetros do objeto (Figura 11 item D). Ao lado do objeto, também se colocou uma moeda de 10 centavos para se ter a referência do tamanho do objeto em comparação ao da moeda.

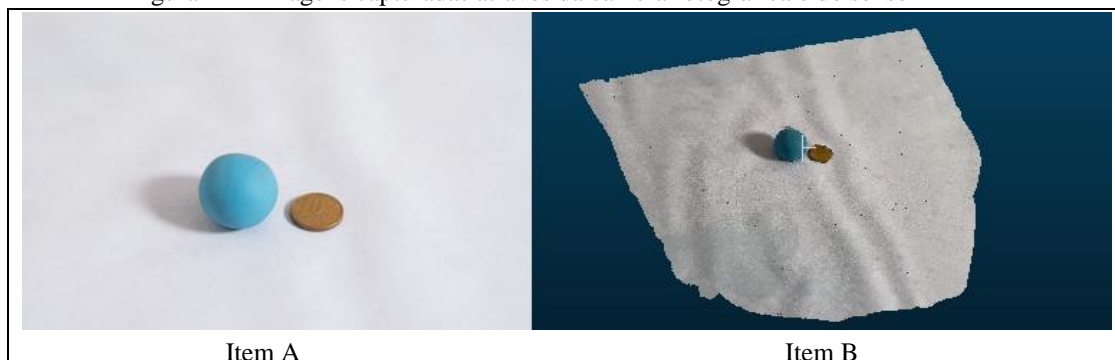
Figura 11 – Preparação do ambiente controlado para realizar a captura dos objetos



Fonte: elaborado pelo autor.

Após a montagem do ambiente, iniciou-se o processo de captura das imagens a serem utilizadas na reconstrução da superfície em 3D. Para realizar esse processo, utilizou-se uma câmera Sony A6600 com uma lente Sony 24mm ZEISS 1.8 para a aquisição das imagens 2D e um iPhone Pro Max 12 com o sensor LiDAR para a captura das nuvens de pontos. Para cada objeto posicionado no centro da mesa, foram realizadas três capturas com a câmera fotográfica e três capturas utilizando o sensor LiDAR, todas as capturas respeitaram as distâncias marcadas anteriormente com as massinhas de modelar (Figura 11 item D). A Figura 12 demonstra um exemplo das capturas realizadas através da câmera fotográfica (Figura 12 item A) e do sensor LiDAR (Figura 12 item B).

Figura 12 – Imagens capturadas através da câmera fotográfica e do sensor LiDAR



Fonte: elaborado pelo autor.

Destaca-se que o objeto utilizado na Figura 12, consiste em uma bolinha de massinha de modelar na cor azul. No lado direito dessa bolinha, tem uma moeda de 10 centavos que foi posicionada para ser utilizada como um objeto de comparação, ambos os objetos (bolinha e moeda) foram posicionados no centro da mesa envelopada com uma toalha branca. Além disso, a Figura 12 exibe dois formatos de visualização para o mesmo objeto. No lado esquerdo, tem-se uma visualização no formato de imagem que foi capturada através da câmera fotográfica Sony A6600. Já no lado direito, uma visualização no formato de nuvem de pontos, essa malha de pontos foi capturada através do sensor LiDAR do iPhone Pro Max 12. Ambas as capturas, foram obtidas com uma distância de 20 centímetros do objeto (bolinha de massinha de modelar).

3.2 TRATAMENTOS EM IMAGENS E NUVENS DE PONTOS

Durante a utilização dos algoritmos de reconstrução de superfícies em 3D, constatou-se um alto consumo da memória RAM, e por consequência, o computador não tinha uma quantidade disponível de memória para que o processamento das imagens fosse concluído. Além disso, identificou-se que, quanto maior for a qualidade da resolução das imagens, mais processamento e memória os algoritmos utilizaram para realizar a reconstrução da superfície em 3D. Vale ressaltar que durante os testes, aplicou-se o algoritmo em uma imagem em 4k, sendo o processamento realizado por um computador com 24GB de memória RAM, o que resultou na utilização de 100% da memória RAM e a finalização do algoritmo informando que a máquina não tinha memória suficiente para finalizar a operação (Tabela 1). Por esse motivo, antes de iniciar o processo de reconstrução de superfícies em 3D, realizou-se um redimensionamento nas imagens capturadas através da câmera fotográfica para ficarem em uma resolução 800x800. Esse processo se tornou necessário para que o algoritmo funcionasse sem a ocorrência de problemas durante o seu processamento e para garantir que não houvesse um grande consumo dos recursos computacionais.

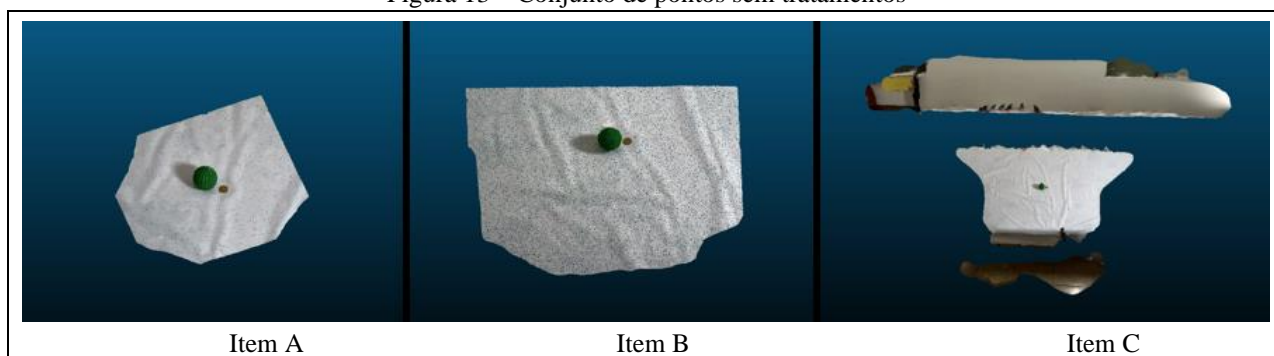
Tabela 1 – Consumo de memória RAM considerando a resolução da imagem

Resolução	Consumo base do computador (RAM)	Consumo da aplicação (RAM)	Tempo de processamento
4k	3GB	21GB	40 segundos (Erro)
800x800	3GB	1,5GB	74 segundos

Fonte: elaborado pelo autor.

Além do redimensionamento realizado nas imagens, identificou-se problemas ao utilizar as nuvens de pontos no algoritmo de reconstrução de superfície em 3D para um conjunto de pontos. Inicialmente, o arquivo original gerado pela tecnologia LiDAR contém apenas as informações dos pontos (x, y, z) e das cores (r, g, b), que podem ser visualizadas através dos itens A, B e C da Figura 13. Portanto, para todas as nuvens de pontos que foram capturadas através da tecnologia LiDAR com distâncias de 20 (Figura 13 item A), 40 (Figura 13 item B) e 80 centímetros (Figura 13 item C), foi necessário aplicar uma normalização nos conjuntos de pontos, pois o algoritmo utilizado para a reconstrução da superfície em 3D, necessita das normalizações dos pontos além das informações já presentes no arquivo para conseguir recriar a superfície para o objeto em 3D. Porém, antes de aplicar a normalização para os conjuntos de pontos capturados, foi necessário aplicar mais um tratamento para as nuvens de pontos adquiridas com distâncias de 80 centímetros (Figura 13 item C). Como o objeto a ser capturado estava bem distante da câmera, a tecnologia LiDAR não conseguiu focar somente no objeto no centro da mesa, acabou pegando outros objetos que estavam presentes no ambiente, mesmo tendo criado um ambiente controlado com apenas os objetos que seriam utilizados para a captura das imagens.

Figura 13 – Conjunto de pontos sem tratamentos

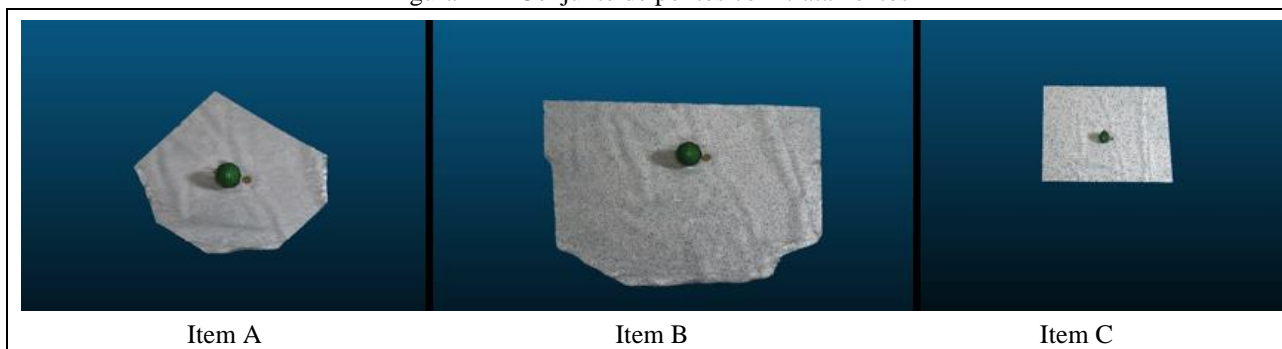


Fonte: elaborado pelo autor.

A Figura 13 apresenta a sequência de nuvens de pontos capturadas em distâncias de 20, 40 e 80 centímetros respectivamente. Em cada nuvem de pontos, têm-se apenas informações dos pontos e das cores. Na última nuvem de pontos, pode-se visualizar a quantidade de pontos indesejáveis que foram obtidos juntamente com o objeto no centro da mesa. Já na Figura 14, têm-se a mesma sequência de nuvens de pontos, porém, além dos pontos e cores possui as normalizações. Além disso, na última nuvem de pontos, pode-se verificar o resultado do processo de segmentação.

Para realizar a normalização e segmentação, utilizou-se a ferramenta *CloudCompare* que possui algumas funções para trabalhar com um conjunto de pontos. Entre as funções disponíveis pela ferramenta, encontra-se a função para realizar a segmentação sob um conjunto de pontos e uma função para computar a normalização do conjunto de pontos. Na primeira etapa do tratamento, realizou-se a segmentação dos pontos relevantes/área de interesse para posteriormente serem utilizados para a normalização. Após aplicar a segmentação para os conjuntos de pontos capturados a 80 centímetros de distância, realizou-se a normalização. O resultado desse processamento foi salvo em um arquivo texto contendo as informações dos pontos (x, y, z), cores (r, g, b) e das normalizações. Na Figura 13 é possível visualizar o conjunto de pontos sem os tratamentos e na Figura 14 é possível visualizar o mesmo conjunto de pontos com os tratamentos aplicados.

Figura 14 – Conjunto de pontos com tratamentos



Fonte: elaborado pelo autor.

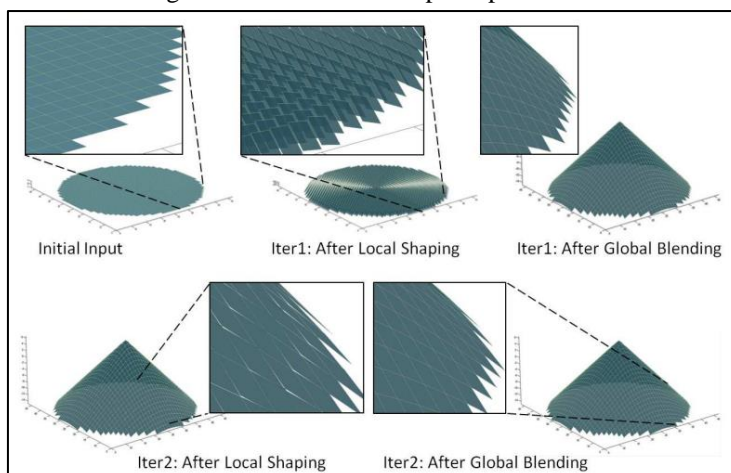
3.3 RECONSTRUÇÃO DA SUPERFÍCIE EM 3D

Esta seção discorre sobre os algoritmos utilizados para realizar a reconstrução da superfície em 3D para os objetos capturadas. Inicialmente, na subseção 3.3.1 será abordado o algoritmo de reconstrução 3D para as imagens que foram capturadas através da câmera fotográfica. Posteriormente, na subseção 3.3.2 serão apresentados dois algoritmos de reconstrução 3D para as nuvens de pontos capturadas através do sensor LiDAR.

3.3.1 Reconstrução da superfície em 3D a partir de imagens 2D

Para reconstruir superfícies em 3D a partir de imagens capturadas através de uma câmera fotográfica, aplicou-se o algoritmo desenvolvido por Xie *et al.* (2014) através do ambiente de desenvolvimento Matlab. Segundo Xie *et al.* (2014), o algoritmo pode reconstruir superfícies a partir de gradientes. O método desenvolvido, formulou-se sob uma estrutura do processamento de geometria discreta. Além disso, os autores comentam que o problema de reconstrução contínua, resolveu-se utilizando sequência de etapas de otimização de mínimos quadros. Segundo Xie *et al.* (2014), a superfície a ser reconstruída é calculada em um modelo de malha composto por várias facetas (Figura 15), onde cada faceta faz correspondência a uma amostra no campo de gradiente.

Figura 15 – Malha M composta por facetas



Fonte: Xie *et al.* (2014)

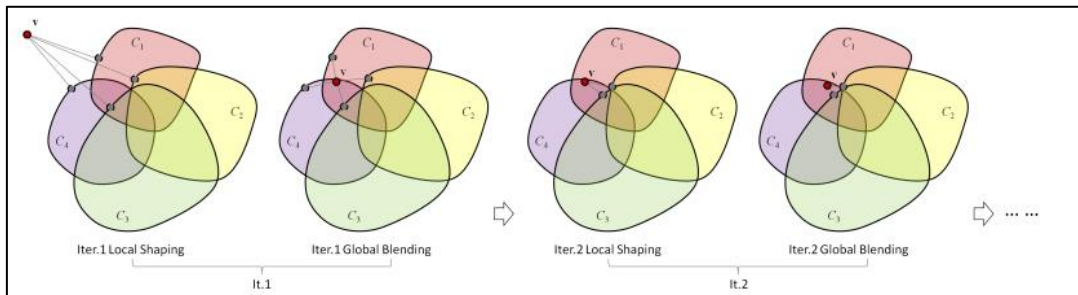
Após a conversão dos gradientes nas amostras para vetores normais, Xie *et al.* (2014) aplicaram um método de Processamento de Geometria Discreta (PGD) com o intuito de deformar a malha M e conduzir as facetas pelos vetores de normais. A deformação da malha utilizando o método PGD é calculada iterativamente e, para cada iteração, utilizou-se a estratégia local/global que consiste em duas etapas. A primeira etapa chama-se modelagem local e essa etapa

determina a posição e a orientação que cada faceta. De acordo com Xie *et al.* (2014), os vértices de uma faceta quadrangular $f_{i,j}$ são projetados com a sua normal $n_{i,j}$. A projeção de um vértice ao longo do eixo Z de um plano, pode ser obtido através da Equação 3.

$$p_{i,j}(v_{k,l}) = c_{i,j}^z - \frac{[n_{i,j}^x(k-i)h + n_{i,j}^y(l-j)h]}{n_{i,j}^z} \quad (3)$$

O resultado da primeira etapa, gera quatro facetas adjacentes que darão quatro posições alvo para o vértice. Caso as orientações alvo de cada faceta for uma região viável, a etapa de modelagem local projeta a posição atual do vértice para os pontos mais próximos das regiões viáveis. Por fim, a segunda etapa chama-se mistura global e, é encarregada para determinar as posições comprometidas dentre as quatro posições projetadas anteriormente, resultando em uma nova forma da superfície da malha, conforme mostra a Figura 16.

Figura 16 – Modelagem local e mistura global



Fonte: Xie *et al.* (2014)

A partir das imagens capturadas através da câmera fotográfica, aplicou-se o algoritmo de Xie *et al.* (2014). Como o algoritmo foi desenvolvido no Matlab, a execução do algoritmo ocorreu na máquina local. Por essa razão, o algoritmo consumiu muita memória RAM para realizar o processamento de reconstrução 3D para imagens no formato 800x800. Conforme mencionado anteriormente na etapa de tratamento, quando maior for a qualidade das imagens a ser processadas, mais recursos computacionais como memória, serão utilizados pelo algoritmo para realizar a reconstrução 3D. Na Figura 17 demonstra-se o resultado obtido através do algoritmo de Xie *et al.* (2014).

Figura 17 – Reconstrução da superfície utilizando algoritmo de Xie et al. (2014)



Fonte: elaborado pelo autor.

No lado esquerdo da Figura 17, exibe-se a captura realizada através da câmera fotográfica, contendo uma bolinha de massinha de modelar e uma moeda de 10 centavos. Já no lado direito da Figura 17, exibe-se a superfície reconstruída para a captura do lado esquerdo. Diante da reconstrução realizada, pode-se verificar que o algoritmo conseguiu reconstruir as regiões que contém os objetos, porém, não conseguiu reconstruir o volume de cada objeto e nem os detalhes contidos na moeda.

3.3.2 Reconstrução da superfície em 3D do sensor LiDAR

Para a reconstrução da superfície em 3D a partir das nuvens de pontos do sensor LiDAR, desenvolveu-se um algoritmo utilizando a linguagem Python, através do ambiente de desenvolvimento Google Colab. Na construção do algoritmo, utilizou-se a biblioteca *Open3D* que permite trabalhar e processar dados 3D. Essa biblioteca possui uma vasta documentação e exemplos de como aplicar algoritmos de reconstrução 3D sobre um conjunto de pontos. Além disso, ela também possui muitos algoritmos de reconstrução 3D, como por exemplo, o algoritmo de Ball pivoting e o algoritmo de Poisson. Ambos foram utilizados para a reconstrução das superfícies em 3D para as nuvens de pontos obtidas com o sensor LiDAR. O Quadro 1 demonstra as etapas realizadas juntamente com o algoritmo de Ball pivoting.

Quadro 1 – Descrição dos processos realizados com o algoritmo Ball pivoting

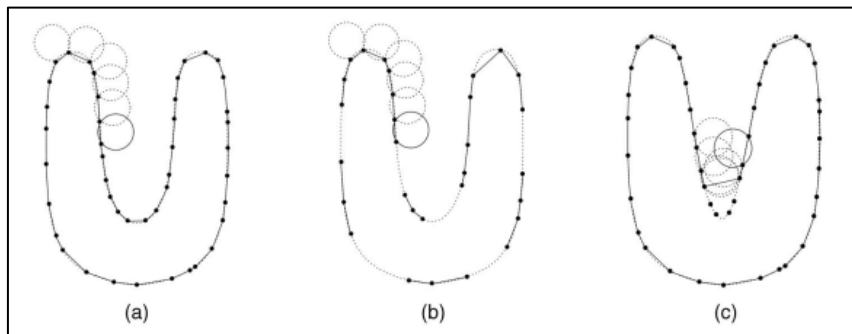
Nº	Tarefa
01	Criação de uma geometria com base nas nuvens de pontos
02	Calcular o raio do círculo que será utilizado na etapa 3
03	Aplicação do algoritmo Ball pivoting
04	Simplificação da malha usando métricas de erro quadrático
05	Tratamentos na malha resultante

Fonte: elaborado pelo autor.

Inicialmente, criou-se um objeto PointCloud com base nas informações presentes no arquivo de texto gerado através do processo de tratamento das nuvens de pontos. Esse objeto necessita de duas informações importantes para a sua construção, os pontos (x, y, z) e as normalizações. Por esse motivo, o pré-processamento realizado anteriormente para computar as normalizações das nuvens de pontos, se tornou um processo essencial para que o conjunto de pontos se torne utilizável pelo algoritmo.

Na segunda etapa, calcula-se o raio do círculo que será utilizado no algoritmo de Ball pivoting. Inicialmente, utilizou-se uma função disponibilizada através do objeto criado anteriormente que calcula as distâncias dos pontos com o seu vizinho mais próximo na nuvem de pontos, resultando em uma lista de distâncias. A partir da lista de distâncias, aplicou-se uma média aritmética para retornar a média dos elementos presentes na lista de distâncias. Após obter a média dos elementos, o valor foi multiplicado três vezes, dessa forma, o raio do círculo foi obtido. Na terceira etapa, aplicou-se o algoritmo de Ball pivoting usando como parâmetro o raio do círculo obtido através da etapa 2. Basicamente, o algoritmo utiliza o método desenvolvido por Bernardini *et al.* (1999) que encontra uma malha triangular que interpola um conjunto desordenado de pontos, conforme exemplificado na Figura 18. O algoritmo reconstrói a superfície utilizando um círculo com um determinado raio, esse círculo percorre o conjunto de pontos e conecta-os com uma aresta (a). Quando a densidade do ponto é muito baixa, o círculo não consegue conectar os pontos, deixando buracos na superfície (b). Além disso, alguns pontos podem não ser alcançados quando o raio do círculo for maior que a curvatura do conjunto de pontos (c).

Figura 18 – Processo de reconstrução da superfície pelo algoritmo de Ball pivoting



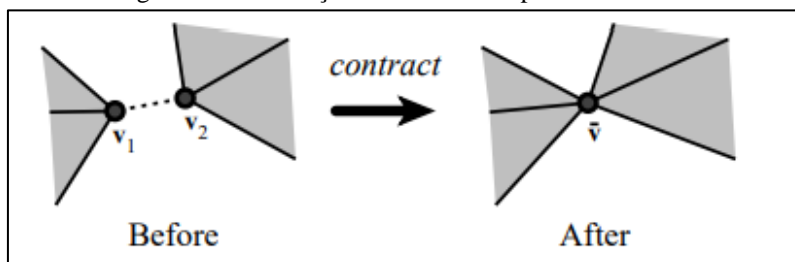
Fonte: Bernardini *et al.* (1999).

Na quarta etapa, aplica-se uma função para simplificar a malha resultante usando métricas de erro quadrático de Garland e Heckbert (1997), que produz rapidamente aproximações em modelo poligonais (Equação 4).

$$E_i = \frac{1}{|X_n| + |X_i|} \left(\sum_{v \in X_n} d^2(v, M_i) + \sum_{v \in X_i} d^2(v, M_n) \right) \quad (4)$$

Segundo Garland e Heckbert (1997), o algoritmo utiliza contrações iterativas de pares de vértices para simplificar modelos e manter aproximações de erro de superfície usando matrizes quádrico (Figura 19). Além disso, os autores comentam que o algoritmo consegue unificar regiões do modelo poligonal que antes não eram conectadas.

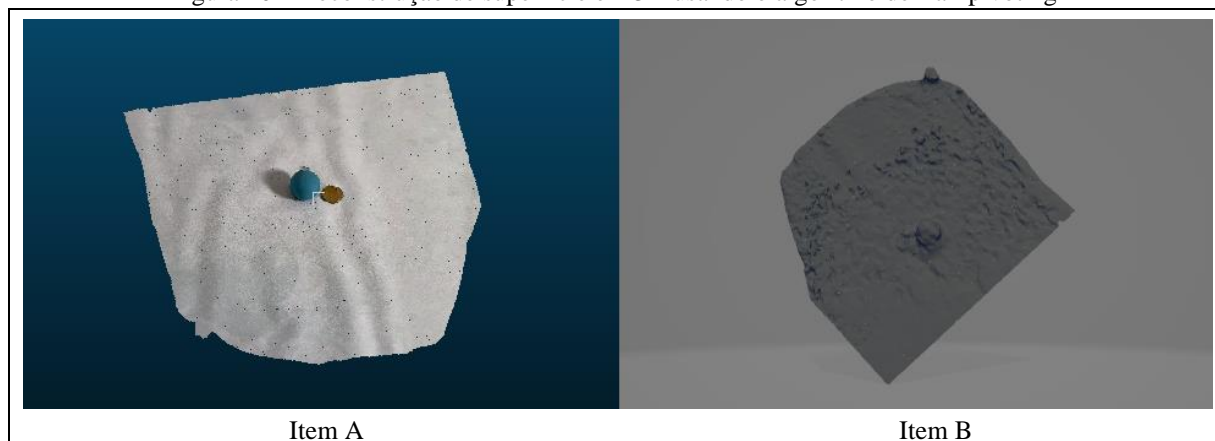
Figura 19 – Contrações iterativas de pares de vértices



Fonte: Garland e Heckbert (1997).

A malha resultante, gerada após a aplicação das métricas de erro quádrico, podem conter algumas imperfeições. Na etapa 5, aplicou-se quatro tratamentos sobre a malha resultante. O primeiro para remover os triângulos que referenciavam o mesmo vértice várias vezes no mesmo triângulo. O segundo para remover os triângulos duplicados, ou seja, os triângulos que referenciavam os mesmos vértices e possuem a mesma orientação. Já o terceiro tratamento, remove os vértices que possuem a mesma coordenada. Dessa forma, manteve-se apenas um único vértice por posição. Por fim, o último tratamento remove todas as arestas não múltiplas. A Figura 20 demonstra o resultado das etapas descritas, na Figura 20 item (a) encontra-se a nuvem de pontos geradas pelo sensor LiDAR e na Figura 20 item (b), a reconstrução da nuvem de pontos utilizando o algoritmo de Ball pivoting. Através dessa figura, pode-se perceber que o algoritmo conseguiu reconstruir a superfície juntamente com o objeto localizado no centro da superfície. Porém, a moeda de 10 centavos que foi posicionada para comparativo, não foi reconstruída pelo algoritmo.

Figura 20 – Reconstrução de superfície em 3D usando o algoritmo de Ball pivoting



Fonte: elaborado pelo autor.

Além do algoritmo de Ball pivoting, aplicou-se também o algoritmo de Poisson para os mesmos conjuntos de pontos. No Quadro 2 constam as etapas realizadas juntamente com o algoritmo de Poisson para reconstruir a superfície 3D para as nuvens de pontos obtidas através do sensor LiDAR.

Quadro 2 – Descrição dos processos realizados com o algoritmo de Poisson

Nº	Tarefa
01	Criação de uma geometria com base nas nuvens de pontos
02	Aplicação do algoritmo de Poisson
03	Capturar os pontos relevantes a partir de uma caixa delimitadora

Fonte: elaborado pelo autor.

Na primeira etapa, criou-se um objeto PointCloud com base nas informações presentes no arquivo de texto gerado através do processo de tratamento das nuvens de pontos. Na segunda etapa, aplicou-se o algoritmo de Poisson desenvolvido por Kazhdan e Hoppe (2013) sobre os conjuntos de pontos. Esse algoritmo resolve problemas com a otimização regularizado ao obter uma superfície lisa para o objeto. Diferente do algoritmo de Ball pivoting demonstrado anteriormente, o algoritmo de Poisson produz resultados mais suaves por conta de os pontos da malha já serem os vértices da malha triangular resultante. Além disso, o algoritmo de Poisson dispõe de um parâmetro para informar a profundidade do detalhamento, quanto maior for esse valor, mais detalhado será a superfície reconstruída. Vale ressaltar que, quanto maior for o valor da profundidade do detalhamento, mais memória RAM e processamento o algoritmo consumirá. A Tabela 2 demonstra o consumo de memória RAM durante a execução do algoritmo de Poisson.

Tabela 2 – Consumo de memória RAM do algoritmo de Poisson

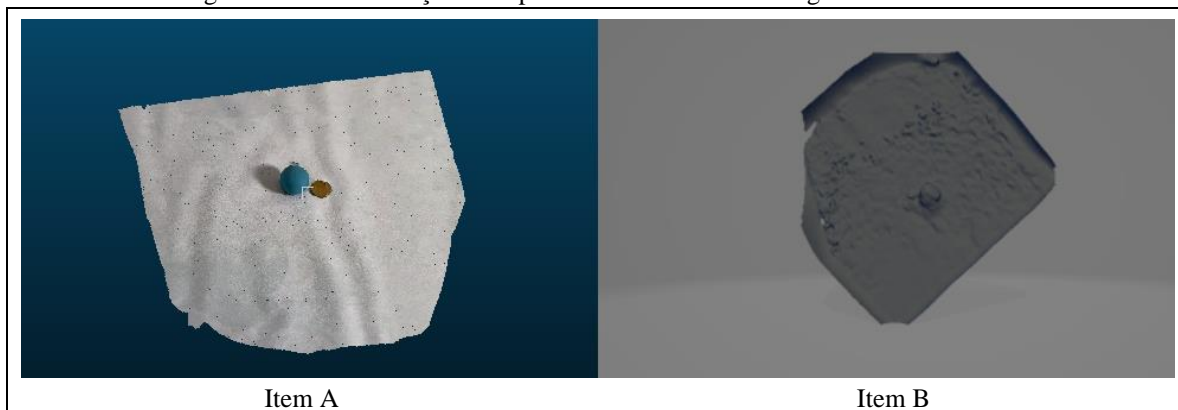
Profundidade	Consumo do algoritmo (RAM)	Tempo de processamento
8 (Padrão)	1GB	18 segundos
12	1.3GB	36 segundos
15	13GB	89 segundos
20	11.3GB	977 segundos

Fonte: elaborado pelo autor.

Por fim, na última etapa, aplicou-se um recorte para capturar apenas os pontos relevantes da malha resultante do algoritmo de Poisson. Durante a reconstrução da superfície, o algoritmo criou uma caixa delimitadora maior que a região que contém a nuvem de pontos obtidas pelo sensor LiDAR. Nas partes da caixa delimitadora que possuía a ausência de pontos, o algoritmo adicionou pontos para deixar a superfície lisa. Por esse motivo, aplicou-se um recorte para tentar obter apenas os pontos inseridos pelo LiDAR. A Figura 21 demonstra o resultado das etapas descritas, na Figura 21 item

(a) encontra-se a nuvem de pontos geradas pelo sensor LiDAR e na Figura 21 item (b), a reconstrução da nuvem de pontos utilizando o algoritmo de Poisson.

Figura 21 – Reconstrução da superfície em 3D usando o algoritmo de Poisson



Fonte: elaborado pelo autor.

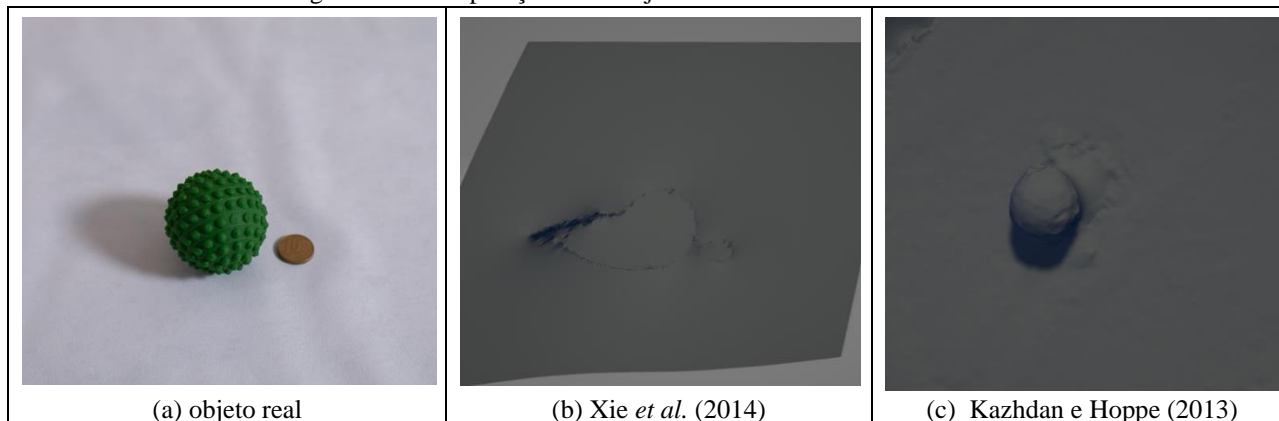
Percebe-se que nas reconstruções obtidas através dos algoritmos de Ball pivoting e Poisson, a malha de pontos de entrada foi reconstruída por completa. Podendo visualizar a bolinha de massinha no centro de cada reconstrução, tanto o algoritmo de Ball pivoting quanto o algoritmo de Poisson, não foram capaz de reconstruir a moeda de 10 centavos que se localizava ao lado do objeto de foco (bolinha de massinha). Durante os testes realizados com os dois algoritmos utilizando todos os recursos computacionais, constatou-se que os objetos pequenos ou pequenos detalhes contidos na nuvem de pontos, não são facilmente exibidos, podendo ser um problema quando o foco da reconstrução é gerar mínimos detalhes.

4 COMPARAÇÃO DE RESULTADOS DOS ALGORITMOS DE RECONSTRUÇÃO 3D

Essa seção tem como objetivo comparar as reconstruções de superfícies em 3D resultantes do algoritmo desenvolvido por Xie *et al.* (2014) e do algoritmo desenvolvido por Kazhdan e Hoppe (2013). Vale relembrar que, o algoritmo de Xie *et al.* (2014), reconstrói a superfícies em 3D a partir de imagens. Já o algoritmo de Kazhdan e Hoppe (2013), realiza a reconstrução da superfície em 3D a partir de uma nuvem de pontos. Para realizar a comparação entre os resultados obtidos através dos algoritmos, utilizou-se uma bola cravo antiestresse como objeto de foco para a reconstrução e para ambos os algoritmos, utilizou-se capturas em distâncias de 20, 40 e 80 centímetros, onde cada distância foi separada em uma etapa de comparação.

Inicialmente, aplicou-se os algoritmos de reconstrução de superfícies em 3D para o objeto captura a 20 centímetros de distância. Na Figura 22 item (a), exibe-se o objeto real que foi utilizado nos algoritmos de reconstrução 3D. Na Figura 22 item (b), demonstra-se a superfície reconstruída para o objeto através do algoritmo desenvolvido por Xie *et al.* (2014). Percebe-se que o algoritmo conseguiu pegar as bordas da bolinha e as bordas da moeda de 10 centavos, porém, não se percebe os detalhes dos objetos e nem os seus volumes. Além disso, percebe-se que para a bolinha, gerou-se uma deformidade no lado direito em comparação com o seu lado esquerdo. Já a Figura 22 item (c), exibe a superfície reconstruída a partir do algoritmo desenvolvido por Kazhdan e Hoppe (2013). Através dessa figura, visualiza-se a bolinha reconstruída com volume e com alguns detalhes na sua superfície, porém, a superfície da moeda de 10 centavos não foi recriada em 3D.

Figura 22 – Comparação entre objeto de 20 centímetros de distância

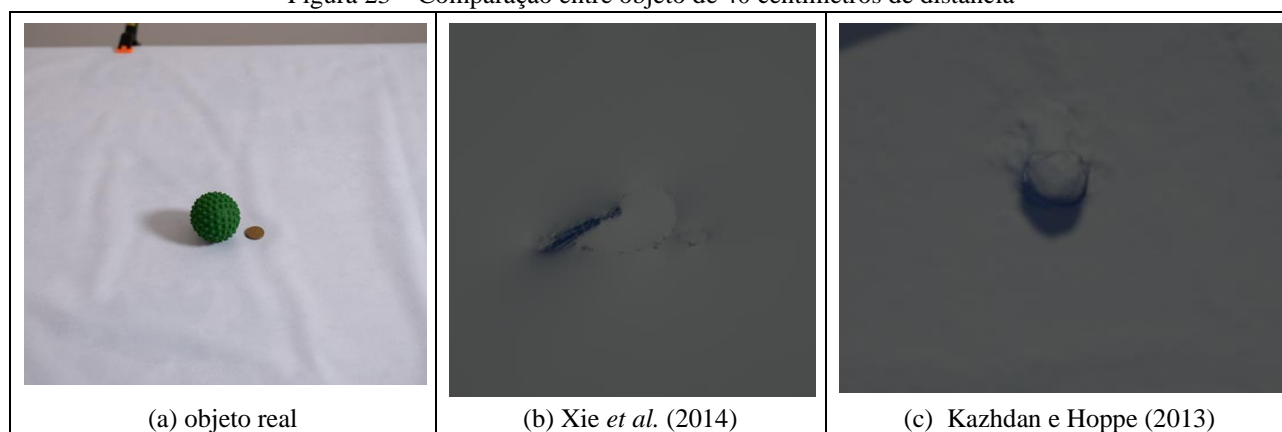


Fonte: elaborado pelo auto.

Na segunda etapa da comparação, aplicou-se os algoritmos de reconstrução de superfície em 3D para o objeto capturado a 40 centímetros de distância. O objeto real que terá a sua superfície reconstruída em 3D a partir dos algoritmos, pode-se observado através da Figura 23 item A. Na Figura 23 item B, demonstra a superfície reconstruída para o objeto a partir do algoritmo de Xie *et al.* (2014). Através dessa figura, percebe-se que o resultado obtido se assemelha com o resultado obtido anteriormente para o objeto a 20 centímetros de distância (Figura 22 item B), sendo a distância em relação ao eixo Z a maior diferença entre os dois resultados.

Já a Figura 23 item C, exhibe a superfície reconstruída para o objeto a partir do algoritmo de Kazhdan e Hoppe (2013). Através dela, percebe-se que o algoritmo conseguiu reconstruir a superfície para o objeto informado, criando pequenos detalhes em sua superfície e modelando uma forma que se assemelha com o objeto real. Diferente do resultado obtido anteriormente para o objeto a 20 centímetros de distância (Figura 22 item C), a superfície resultante dessa etapa (Figura 23 item C), possui deformidade em torno do objeto reconstruído.

Figura 23 – Comparação entre objeto de 40 centímetros de distância

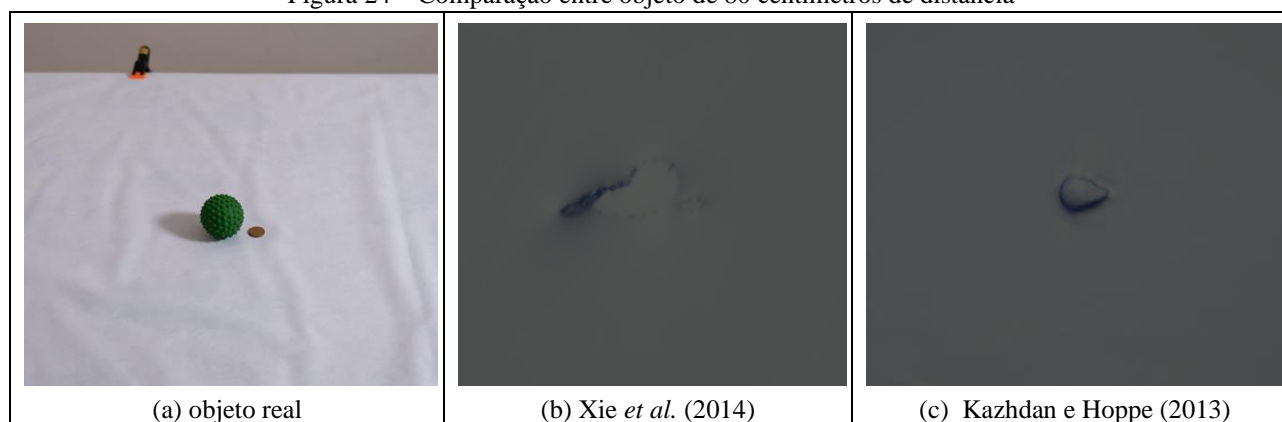


Fonte: elaborado pelo auto.

Para a terceira etapa da comparação, aplicou-se os mesmos algoritmos de reconstrução de superfície em 3D, porém, para objetos capturadas a 80 centímetros de distância. Igualmente nas comparações anteriores, a Figura 24 item (A) exhibe o objeto real que será utilizado nos algoritmos de reconstrução. Através da Figura 24 item (B), percebe-se que o algoritmo de Xie *et al.* (2014) reconstruiu uma superfície semelhante ao que foi apresentado na primeira (Figura 22 item B) e segunda (Figura 23 item B) etapa. Porém, percebe-se que a região onde a moeda de 10 centavos fica localizada, não está muito bem visível em comparação aos resultados obtidos anteriormente (Figura 22 item B e Figura 23 item B).

Já a Figura 24 item C, exhibe a superfície reconstruída através do algoritmo de Kazhdan e Hoppe (2013). Através dessa figura, percebe-se que o algoritmo teve dificuldade em reconstruir objetos em que a sua captura seja muito distante. Em comparação aos resultados que foram apresentados na Figura 22 item C e na Figura 23 item C, o objeto reconstruído nessa etapa não se assemelha a uma bolinha. A modelagem do objeto ficou toda deforma e não percebe-se nenhum detalhe em sua superfície.

Figura 24 – Comparação entre objeto de 80 centímetros de distância



Fonte: elaborado pelo auto.

5 CONCLUSÕES

Nos dias atuais, as técnicas de visão computacional vêm ganhando bastante espaço em diversas áreas de pesquisa. Além disso, o avanço das tecnologias móveis está permitindo a aquisição de dados para a reconstrução das superfícies

em 3D de uma maneira muito mais fácil e prática. Conforme descrito, o objetivo deste artigo era comparar algoritmos que realizam a reconstrução da superfície com diferentes tipos de entrada. Os dois principais algoritmos testados foram o de Xie *et al.* (2014) que utiliza imagens capturadas através de câmera fotográfica e o de Kazhdan e Hoppe (2013) que utiliza uma nuvem de pontos obtidas através de um scanner a laser, no caso desse trabalho, utilizou-se o Light Detection and Ranging (LiDAR).

A partir dos algoritmos utilizados, concluiu-se que para processar os algoritmos que reconstrói superfícies super detalhadas e com uma ótima qualidade, necessita-se de muitos recursos computacionais, tais como, uma boa quantidade de memória RAM e um bom processador. Nos testes realizados, identificou-se que para o algoritmo de Xie *et al.* (2014) quanto maior foi a resolução da imagem, mais memória RAM será consumido para reconstruir a superfície. Já o algoritmo de Kazhdan e Hoppe (2013), quando maior for o nível de detalhamento, mais memória RAM será consumida pelo algoritmo.

Além disso, a partir dos resultados obtidos através dos algoritmos, constatou-se que o algoritmo de reconstrução 3D a partir de imagens possui uma qualidade e uma reconstrução inferior ao do algoritmo de reconstrução 3D a partir de nuvens de pontos. Porém, identificou-se problemas ao reconstruir a superfície mais detalhada para os objetos que foram capturados com 80 centímetros de distância.

Como trabalhos futuros, sugere-se focar na reconstrução de superfícies em 3D a partir de nuvens de pontos, porém, com ênfase na reconstrução dos detalhes minimalista do objeto real. Dessa forma, a sua utilização pode ser redirecionada para reconstrução de imagens dermatológicas, possibilitando um melhor entendimento dos cânceres de pele pelo médico especialista.

REFERÊNCIAS

- ALHAIJA, Hassan A. *et al.* Augmented Reality Meets Computer Vision: efficient data generation for urban driving scenes. **International Journal Of Computer Vision**, [S.l.], v. 126, n. 9, p. 961-972, 7 mar. 2018.
- BENALCAZAR, Daniel P. *et al.* A 3D Iris Scanner From a Single Image Using Convolutional Neural Networks. **Ieee Access**, [S.l.], v. 8, p. 98584-98599, maio 2020.
- BERNARDINI, Fausto *et al.* The ball-pivoting algorithm for surface reconstruction. **IEEE transactions on visualization and computer graphics**, v. 5, n. 4, 1999, p. 349-359.
- BERTSEKAS, Dimitri. A distributed asynchronous relaxation algorithm for the assignment problem. In: 1985 24TH IEEE CONFERENCE ON DECISION AND CONTROL, 24., 1985, Fort Lauderdale, Fl, Usa. **1985 24th IEEE Conference on Decision and Control**. [S.L.]: Ieee, 1985. p. 1703-1704.
- CHANG, Angel X. *et al.* **ShapeNet**: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012, 2015.
- CHOY, Christopher B. *et al.* 3D-R2N2: a unified approach for single and multi-view 3d object reconstruction. **Computer Vision – Eccv 2016**, [S.l.], v. 1, n. 0, p. 628-644, out. 2016.
- COSTA, Guilherme Ramalho. **Reconstrução de um modelo 3D a partir de imagens térmicas 2D de uma mama via câmera infravermelha**. 2021. 87 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, 2021.
- DRUCKER, H.; CUN, Y. Le. Improving generalization performance using double backpropagation. **Ieee Transactions On Neural Networks**, [S.l.], v. 3, n. 6, p. 991-997, nov. 1992.
- LOURA, Daniel. de S. A.; OLIVEIRA, Yuri. de M. A. de; RAIMUNDO, Pedro. O.; AGÜERO, Karl. P. A. Reconstrução 3D de Objetos com Kinect e Câmera Digital. **Revista Eletrônica de Iniciação Científica em Computação**, [S. l.], v. 16, n. 6, 2018.
- FAN, Haoqiang; SU, Hao; GUIBAS, Leonidas J. A Point Set Generation Network for 3D Object Reconstruction From a Single Image. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7., 2017, Honolulu. **Proceedings...** Honolulu: Ieee, 2017. p. 605-613.
- FANTI, Carlos Eduardo. **Reconstrução de superfícies tridimensionais utilizando B-splines com peso associado à redução do número de pontos de controle**. 2016. 139 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, 2016.
- GARLAND, Michael.; HECKBERT, Paul S.. Simplifying surfaces with color and texture using quadric error metrics. In: VISUALIZATION '98, 98., 1998, Research Triangle Park, Nc, Usa. **Proceedings Visualization '98 (Cat. No.98CB36276)**. [S.L.]: Ieee, 1998. p. 263-269.
- GARLAND, Michael; HECKBERT, Paul S. Surface simplification using quadric error metrics. In: SIGGRAPH97: The 24th International Conference on Computer Graphics and Interactive Techniques, 24., 1997, New York. **Proceedings...** New York: ACM Press/Addison-Wesley Publishing Corporation, 1997. p. 209-2016.
- GEIGER, Andreas *et al.* Vision meets robotics: The KITTI dataset. **The International Journal Of Robotics Research**, [S.l.], v. 32, n. 11, p. 1231-1237, 23 ago. 2013.

- GIONGO, Marcos et al. LiDAR: princípios e aplicações florestais. **Pesquisa Florestal Brasileira**, [S.L.], v. 30, n. 63, p. 231-244, 28 out. 2010.
- JOSÉ, Marcelo A.; LOPES, Roseli D. Sistema de Reconstrução 3D de baixo custo. In: International Conference on Graphics Engineering for Arts and Design, 7., 2007, Curitiba. **Anais...** Curitiba: Graphica, 2007. p. 1-10.
- JOSÉ, Marcelo Archanjo. **Reconstrução tridimensional de baixo custo a partir de par de imagens estéreo**. 2008. 153 f. Dissertação (Mestrado) - Curso de Sistemas Eletrônicos, Escola Politécnica, São Paulo, 2008.
- KAZHDAN, Michael; HOPPE, Hugues. Screened poisson surface reconstruction. **ACM Transactions on Graphics (ToG)**, v. 32, n. 3, 2013, p. 1-13.
- KHATAMIAN, Alireza; ARABNIA, Hamid R. Survey on 3D surface reconstruction. **Journal of Information Processing Systems**, v. 12, n. 3, 2016, p. 338-357.
- KLEMT, A.; INFANTOSI, Antonio F. C. Método de superfície na visualização 3D da dissecação do crânio humano. **Research on Biomedical Engineering**, v. 16, n. 1, p. 21-37, 2011
- KUTULAKOS, Kiriakos N.; SEITZ, Steven M. A Theory of Shape by Space Carving. **International Journal Of Computer Vision**, [S.L.], v. 38, n. 3, p. 199-218, jan. 2000.
- LORENSEN, William E.; CLINE, Harvey E. Marching cubes: a high resolution 3d surface construction algorithm. **Acm Siggraph Computer Graphics**, [S.L.], v. 21, n. 4, p. 163-169, ago. 1987.
- MESCHEDER, Lars et al. Occupancy Networks: Learning 3D Reconstruction in Function Space. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10., 2019, California. **Proceedings...** California: Ieee, 2019. p. 4460-4470.
- QUAN, Long *et al.* Surface reconstruction by integrating 3D and 2D data of multiple views. In: ICCV 2003: 9TH INTERNATIONAL CONFERENCE ON COMPUTER VISION, 9., 2003, Nice, France. **Proceedings Ninth IEEE International Conference on Computer Vision**. [S.L.]: Ieee, 2003. p. 1313-1320.
- SILVA, Isabel C. S. **Avaliação da Qualidade de Imagens Médicas Geradas por Ray Casting**. 2003. 124. Dissertação para obtenção de grau de Mestre em Ciência da Computação –Universidade Federal do Rio Grande do Sul, Rio Grande do Sul.
- SONG, Hyun Oh et al. Deep Metric Learning via Lifted Structured Feature Embedding. In: 2016 Ieee Conference On Computer Vision And Pattern Recognition (Cvpr), 6., 2016, Las Vegas. **Proceedings...** Las Vegas: Ieee, 2016. p. 4004-4012.
- SZELISKI, Richard. **Computer Vision: Algorithms and Applications** (Texts in Computer Science). Washington, D.C: Springer, 2019. 832 p.
- WANG, Ding kang; WATKINS, Connor; XIE, Huikai. MEMS Mirrors for LiDAR: a review. **Micromachines**, [S.L.], v. 11, n. 5, p. 456-480, 27 abr. 2020.
- XIE, Wuyuan et al. Surface-from-Gradients: an approach based on discrete geometry processing. **2014 Ieee Conference On Computer Vision And Pattern Recognition**, [S.L.], p. 2203-2210, jun. 2014.
- ZENG, Gang et al. Surface Reconstruction by Propagating 3D Stereo Data in Multiple 2D Images. **Lecture Notes In Computer Science**, [S.L.], v. 3021, p. 163-174, maio 2004.