



UNIVERSIDADE FEDERAL DO CEARÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

MATEUS ARRUDA DE MEDEIROS

**RECONSTRUÇÃO TRIDIMENSIONAL EM NUVEM DE PONTOS UTILIZANDO O
MÉTODO SIFT E TÉCNICAS DE ESTRUTURAÇÃO POR MOVIMENTO**

SOBRAL

2018

MATEUS ARRUDA DE MEDEIROS

RECONSTRUÇÃO TRIDIMENSIONAL EM NUVEM DE PONTOS UTILIZANDO O
MÉTODO SIFT E TÉCNICAS DE ESTRUTURAÇÃO POR MOVIMENTO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia da Computação.

Orientador: Prof. Dr. Iális Cavalcante de Paula Júnior

SOBRAL

2018

MATEUS ARRUDA DE MEDEIROS

RECONSTRUÇÃO TRIDIMENSIONAL EM NUVEM DE PONTOS UTILIZANDO O
MÉTODO SIFT E TÉCNICAS DE ESTRUTURAÇÃO POR MOVIMENTO

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia da
Computação da Universidade Federal do Ceará,
como requisito parcial à obtenção do grau de
bacharel em Engenharia da Computação.

Aprovado em: 08/06/2018.

BANCA EXAMINADORA

Prof. Dr. Iális Cavalcante de Paula
Júnior (Orientador)
Universidade Federal do Ceará (UFC)

Prof^a. M.a Jermana Lopes de Moraes
Universidade Federal do Ceará (UFC)

Prof. Dr. Marcelo Marques Simões de Souza
Universidade Federal do Ceará (UFC)

À toda minha família pelo esforço que colocaram em mim e pelos sermões durante meu amadurecimento. Mãe, sua preocupação por mim me fez perceber o quanto sou importante em nossa família. Pai, obrigado pelos conselhos e espero ter sido um bom filho durante esses anos.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela dádiva da vida e oportunidade de aprender.

Aos meus pais Juarez Marques Junior e Antônia Lendengue e irmão Juarez Neto que sempre estiveram ao meu lado e se dedicaram a me orientar.

Aos meus avós Juarez Marques e Maria Luiza que me auxiliaram e aconselharam durante todo esse tempo.

Aos meus quatro amigos Carolina Magalhães, Lucas Mota, Samara Feitosa e Jordão Alcântara, que estiveram juntos comigo desde o início dessa jornada.

Ao meu amigo José Guilherme por ter me ajudado nas disciplinas mais difíceis e pela amizade verdadeira.

Ao meu amigo Hugo Damasceno pela grande amizade antes, durante e depois do curso.

Ao meu orientador Prof. Dr. Iális Cavalcante por ser um dos melhores professores da Universidade Federal do Ceará.

À minha namorada, Tatiane Matias, por ter me ajudado nos momentos difíceis em que estava sozinho.

Aos agentes da limpeza que mantém universidade limpa e aos servidores do Restaurante Universitário pelo trabalho duro e simpatia nos momentos de lanche.

“A medida de amar é amar sem medida.”

(Humberto Gessinger)

RESUMO

Com o avanço da tecnologia, o ramo da pesquisa, indústria e entretenimento têm focado no desenvolvimento de software e sua integração com o público em geral. Algumas dessas pesquisas tornaram-se muito promissoras como Processamento de Imagens, Visão Computacional e Modelagem Tridimensional. Iniciada já na década de 70, esta última teve um avanço enorme graças ao melhoramento do *hardware* com o decorrer do tempo. Devido a necessidade do público em geral em estar cada vez mais imerso nessa tecnologia, este trabalho tem o objetivo de trazer conceitos de modelagem e técnicas sobre reconstrução tridimensional. Serão vistos alguns conceitos sobre câmeras e como funciona o processo de captura de imagens, alguns conceitos de processamento de imagens como o método SIFT para extração de características, e um conjunto de técnicas em reconstrução 3D a partir de fotografias chamada de *Structure from Motion* (Estruturação por Movimento). Ao final, será apresentada uma aplicação para *smartphone* que utiliza esses conceitos para realizar uma reconstrução 3D.

Palavras-chave: Reconstrução 3D. Processamento de Imagens. SIFT. *Structure from Motion*.

ABSTRACT

With the advance of technology, the fields of research, industry and entertainment have been focusing on software development and its integration with the general public. Some of this researches became promising like Image Processing, Computer Vision and Three Dimensional Modeling. This last one being started on the 70's, had an enormous advance thanks to the hardware improvement along the years. Due the need of the public to be immersed in this technology, this work brings some 3D modeling concepts and techniques for 3D reconstruction. It will be seen some concepts about camera and how the image capturing works, some image processing concepts like the SIFT algorithm for feature extraction, and a set of techniques about 3D reconstruction based on photos, called Structure from Motion. Finally, will be presented an smartphone application that uses all theses concepts to do a 3D reconstruction.

Keywords: *3D Reconstruction. Image Processing. SIFT. Structure from Motion*

LISTA DE FIGURAS

Figura 1 – Modelo de captura de imagens por uma câmera digital.	16
Figura 2 – Representação de uma imagem com distorção radial e o resultado após a correção.	17
Figura 3 – Aplicação do algoritmo SIFT e representação dos seus <i>keypoints</i>	19
Figura 4 – Representação do mesmo objeto em coordenadas Euclidianas (esquerda) e Homogêneas (direita)	21
Figura 5 – Projeção de coordenadas 3D em um plano 2D.	22
Figura 6 – Exemplificação do mapeamento de um ponto usando geometria epipolar. . .	24
Figura 7 – Calibragem da câmera usando a biblioteca OpenCV com padrões de xadrez de 9x6 quadrados.	27
Figura 8 – Aplicativo Android para captura e envio das imagens a serem enviadas. . . .	28
Figura 9 – Aplicativo Android para captura e envio das imagens a serem enviadas. . . .	29
Figura 10 – Fluxograma com as etapas de funcionamento do aplicativo.	29
Figura 11 – Resultado da reconstrução utilizando apenas a biblioteca OpenCV.	30
Figura 12 – Resultado da reconstrução utilizando a biblioteca OpenMVG	31

LISTA DE TABELAS

Tabela 1 – Especificações dos materiais.	26
--	----

LISTA DE SÍMBOLOS

(x_u, y_u)	Coordenadas corrigidas referentes aos eixos (x, y)
(x_d, y_d)	Coordenadas distorcidas referentes aos eixos (x, y)
(x_c, y_c)	Coordenadas centrais da distorção (x, y)
k_n	Coefficientes de distorção da câmera
r	Raio entre o ponto central da imagem e o ponto extremo do círculo de distorção
f_x, f_y	Comprimento focal da câmera
p_x, p_y	Ponto principal da imagem
F	Matriz Fundamental
K	Matriz dos parâmetros intrínsecos das câmeras
I	Imagem projetada
G	Função Gaussiana
σ	Parâmetro de escala
e	Constante de Euler
DoG	Diferença de Gaussianas
L	Imagem projetada com filtro desfocada
d	Distância entre o plano bidimensional e o ponto focal da câmera

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Motivação	13
1.2	Objetivo Geral	14
1.3	Objetivos Específicos	14
1.4	Organização	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Captura de Imagens	15
2.2	Distorção e Calibragem	16
2.3	Extração e Combinação de Características	18
2.4	Geometria de Visão Múltipla	21
2.4.1	<i>Parâmetros da Câmera</i>	23
2.4.2	<i>Geometria Epipolar</i>	23
3	METODOLOGIA	25
3.1	Materiais	25
3.1.1	<i>OpenCV e OpenMVG</i>	25
4	RESULTADOS	30
4.1	Pesquisa de Mercado	33
5	CONCLUSÕES E TRABALHOS FUTUROS	35
5.1	Trabalhos Futuros	35
	REFERÊNCIAS	36

1 INTRODUÇÃO

A necessidade do homem moderno de utilizar tecnologias visuais levou o ramo da pesquisa e indústria a estudar novas tecnologias que utilizam processamento de imagens, especialmente na utilização de modelos 3D (do inglês, *Three Dimensional*). Obras cinematográficas com efeitos especiais, videogames, mapeamento por satélites, dentre outras aplicações, fazem o uso de técnicas de processamento e manipulação de imagens muitas vezes para transformar as informações do mundo real em modelos no ambiente digital.

Após a invenção do primeiro computador com uma tela de osciloscópio conhecido como *Whirlwind I*, o homem vem desenvolvendo novas tecnologias capazes de imergir o usuário em novas experiências. Nesse aspecto, as tecnologias 3D vêm tomando muito espaço. Os jogos eletrônicos, que começaram no ambiente 2D, evoluíram e atualmente existem jogos com um grau de realidade quase idêntico ao do mundo real. Em filmes, especialmente nos de ficção, empresas gastam milhões de dólares para aplicar efeitos especiais em 3D. A tecnologia mais recente e bastante utilizada em todas as áreas de produção é a impressão 3D.

Atualmente, existem várias técnicas de criação de material 3D, mais conhecidas como modelagem 3D, que são mais eficientes dependendo da sua aplicação. Neste trabalho será abordada a técnica de modelagem conhecida como *IBM* (do inglês, *Image Based Modeling*) voltada para uma aplicação de smartphone. Essa técnica consiste na criação de um modelo tridimensional no ambiente virtual a partir de um conjunto de fotografias que compartilham do mesmo objeto. O modelo em si é representado por um conjunto de pontos distribuídos nos eixos X, Y e Z no ambiente 3D, chamados de nuvem de pontos.

1.1 Motivação

Equipamentos de boa qualidade e com um preço acessível para maioria da população é uma realidade cada vez mais concreta. Uma pesquisa da Google revelada pela Folha de São Paulo (FOLHA, 2017) mostrou que no ano de 2016, 62% dos brasileiros possuíam *smartphones*. É uma boa oportunidade para que as pessoas tenham mais familiaridade com a modelagem 3D, já que outras tecnologias como impressão 3D e realidade aumentada estão ganhando mais usuários no meio tecnológico. Por ser um tema relativamente novo, ainda é requerido um certo conhecimento tanto nos conceitos de modelagem 3D como de informática avançada para preparar os requisitos e realizar uma modelagem baseada em imagens.

Portanto, é interessante que esse tema se torne mais viável para os usuários que não tem necessariamente muito conhecimento em computação e que possa ser agregado a outras áreas de estudo.

1.2 Objetivo Geral

Criar um aplicativo para *smartphone* que facilite a experiência do usuário em relação a modelagem 3D, retornando-o um arquivo tridimensional em nuvem de pontos que pode ser executado por outro aplicativo de visualização 3D.

1.3 Objetivos Específicos

- Desenvolver de um aplicativo Android que utilize a câmera de celular para tirar fotos e enviá-las a um servidor.
- Fazer a calibragem da câmera e realizar o processo correção de distorção da imagem.
- Usar o conjunto de imagens para construir um modelo 3D.
- Retornar o modelo 3D do servidor de volta ao *smartphone*;

1.4 Organização

Este trabalho está organizado em cinco capítulos. A partir do capítulo 2, serão discutidos os conceitos que constroem a fundamentação teórica sobre modelagem 3D. No capítulo 3, serão vistos os materiais e qual metodologia será abordada no desenvolvimento do projeto. No capítulo 4, serão apresentados os dados obtidos e no capítulo 5, quais foram as conclusões tiradas ao fim do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Modelagem 3D baseada em imagens fotográficas é também chamada de reconstrução 3D. Neste capítulo, serão discutidas as ideias que constroem a base do estudo de reconstrução 3D, tais como a representação do ambiente 3D em um plano 2D, como obter os parâmetros da câmera, extração das informações mais importantes da imagem, entre outros.

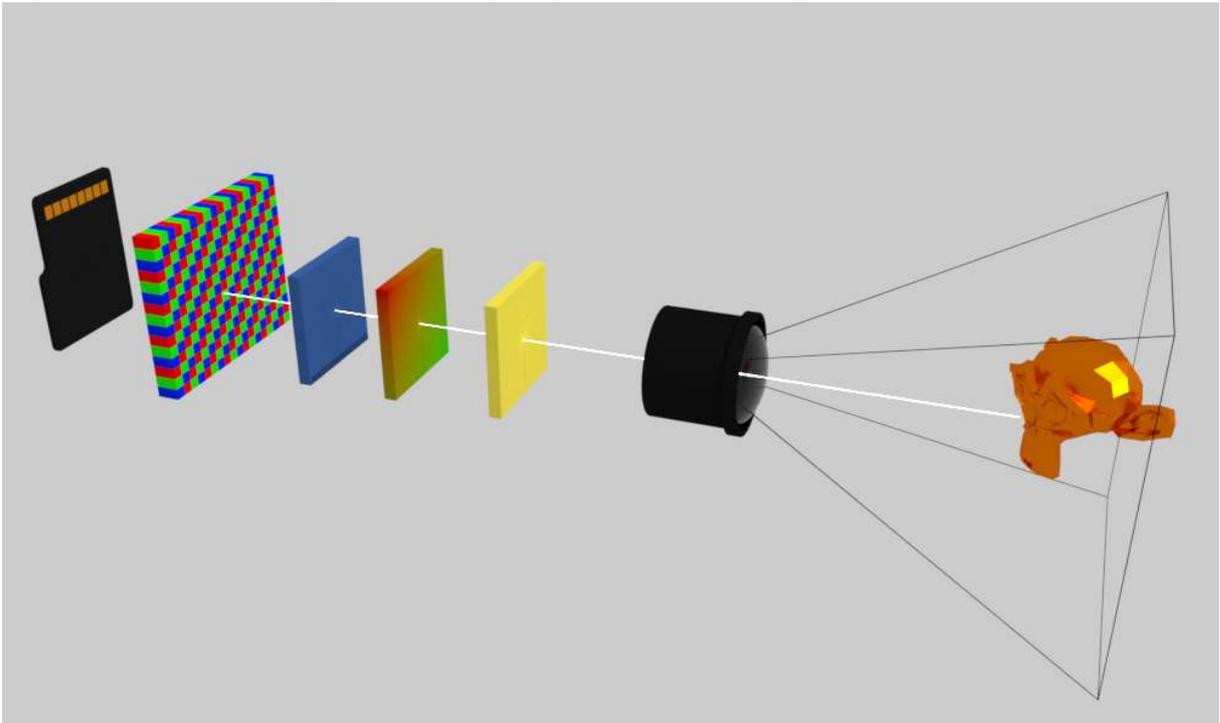
2.1 Captura de Imagens

As câmeras digitais encontradas em dispositivos móveis possuem uma capacidade similar a do olho humano em captar luz. Na obra Villegas (2009), é trazida uma abordagem de como ocorre a captura de imagens a partir de câmeras digitais. O processo se dá quando a luz atravessa a lente da câmera e incide no sensor de luz constituído por fotodiodos, onde quanto mais luz incide pelo sensor, mais corrente elétrica é gerada. Entretanto, nessa técnica são captadas imagens apenas em tons de cinza. Para resolver o problema, são utilizados filtros coloridos (vermelho, verde e azul) para captar cada conjunto de cores da imagem. Com isso, utiliza-se o padrão Bayer, mostrado na obra de Bayer (1976), para sobrepor a matriz de cores e formar uma imagem colorida. Por fim, o sinal digital é distribuído em uma matriz de elementos de exibição chamados *pixels*. Cada *pixel* é constituído por uma combinação das cores vermelha, verde e azul (sistema de cores RGB) e maioria dos dispositivos podem exibir cerca de 16 milhões de *pixels* em sua gama de cores. Caso seja necessário, a matriz de cores pode ser armazenada em um dispositivo de armazenamento, como um cartão de memória.

O processo de captura de luz por câmeras digitais, representado na Figura 1, mostra um objeto do mundo real onde a luz do próprio objeto e do cenário ao redor são focalizados pela lente da câmera e passam por vários sensores e por fim, os dados captados são distribuídos na matriz de *pixels*.

Cada câmera possui suas especificações como parâmetros focais e ponto central em relação aos eixos x e y . Estas especificações são relacionadas aos parâmetros intrínsecos da câmera enquanto parâmetros extrínsecos são relacionados a posição e rotação da câmera. Existem casos em que uma determinada câmera não capta fielmente as informações do ambiente, causando distorção da imagem. Um exemplo disso são câmeras olho-de-peixe, na qual a imagem é distorcida de forma que seja possível captar informações em ângulos de até 180°.

Figura 1 – Modelo de captura de imagens por uma câmera digital.



Fonte: Elaborado pelo autor.

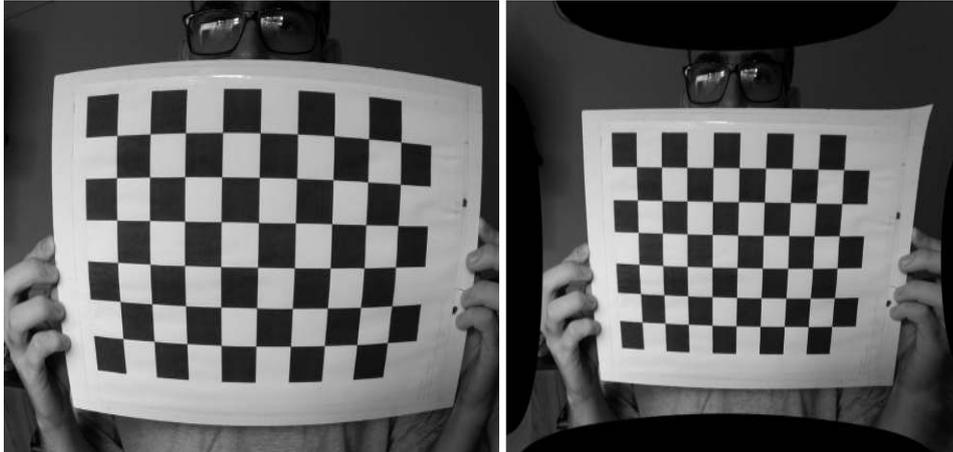
2.2 Distorção e Calibragem

Para realizar uma reconstrução 3D com uma taxa mínima de erros, faz-se necessário o processo de calibragem de câmera (BRADSKI; KAEHLER, 2008). Nesse processo, são tiradas fotografias de um padrão de textura (xadrez ou pontos circulares), onde um algoritmo de calibragem da biblioteca OpenCV (MORDVINTSEV; RAHMAN, 2013) detecta os padrões e determina a taxa de erro que uma câmera específica possui. A biblioteca ainda possui ferramentas para realizar o processo inverso de distorção baseado na taxa de erro encontrada na calibragem.

A calibragem da câmera está relacionada especificamente em determinar quais são os parâmetros intrínsecos da câmera digital. Após determinar os parâmetros da câmera e o quanto essa câmera distorce a imagem, aplica-se a taxa de erro encontrada para corrigir a imagem, alterando-a para que o padrão seja o mais próximo do objeto real possível. A distorção mais comum dentre as câmeras são do tipo radial, onde a imagem é esticada simetricamente para todas as direções partindo de um ponto central. Com isso, é possível visualizar mais informações por conta do ângulo de visão ser maior. Entretanto, isso causa a perda de fidelidade entre a projeção e o objeto real.

A representação da distorção radial e a diferença para uma imagem sem distorção (ou muito pequena) é mostrada na Figura 2.

Figura 2 – Representação de uma imagem com distorção radial e o resultado após a correção.



Fonte: Elaborado pelo autor.

Em Villiers *et al.* (2008), é tratada uma abordagem de distorção e correção de imagens. Matematicamente a correção da distorção radial nas câmeras é dada pelas seguintes equações:

$$x_u = x_d + (x_d - x_c)(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2.1)$$

$$y_u = y_d + (y_d - y_c)(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2.2)$$

Onde x_d e y_d são os pontos distorcidos na imagem, k_1, k_2 e k_3 são os coeficientes de distorção da câmera, e r é o raio onde o centro é ponto central do círculo de distorção, ou seja

$$r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}. \quad (2.3)$$

Onde x_c e y_c são os pontos centrais da distorção. Na equação, esses valores são atribuídos a x_u e y_u , que são as coordenadas corrigidas da imagem com distorção.

Para determinar os parâmetros da câmera utiliza-se um padrão texturizado para em seguida executar o reconhecimento dos padrões nessa textura. Para esse projeto, foi usado o padrão xadrez de 9 por 6 cantos impresso em uma folha papel A4. Nesse padrão, os cantos de cada quadrado são identificados e marcados. Nove marcações horizontais e seis marcações verticais são mapeadas na imagem e então é verificado se há uma curva no decorrer dessas marcações. E também são necessárias várias imagens para obter melhores resultados. Na seção 3 será mostrado o resultado desse processo.

Câmeras sem usos específicos como as de *smartphones*, possuem distorções, entretanto são muito inferiores e quase imperceptíveis, mas ainda se faz necessário determinar seus parâmetros de distorção para obter melhores resultados.

2.3 Extração e Combinação de Características

Na reconstrução 3D em nuvem de pontos é preciso fotografar um mesmo objeto em diferentes ângulos para que os elementos similares entre as fotografias possam ser extraídos e transformados em pontos no ambiente 3D. Para determinar quais pontos de uma imagem são mais característicos, usam-se métodos de extração de características locais de uma imagem. Os métodos mais eficientes na área de processamento de imagem e que são implementados na biblioteca OpenCV são os métodos SIFT (do inglês, *Scale Invariant Feature Transform*) (LOWE, 1999), SURF (do inglês, *Speeded Up Robust Feature*) (BAY *et al.*, 2006) e ORB (do inglês, *Oriented FAST and Rotated BRIEF*) (RUBLEE *et al.*, 2011). Para este projeto será tratado apenas o método SIFT, por possuir maior eficiência e ser mais utilizado em extração de características.

No trabalho de Gonzáles (2010), Aplicação da Técnica SIFT para Determinação de Campos de Deformações de Materiais são descritas as etapas para aplicação do método SIFT. Primeiramente, determina-se quais são os *pixels* mais característicos em uma imagem. Para isso, dada uma imagem $I(x, y)$, definimos a mesma imagem desfocada como $L(x, y, \sigma)$. O desfoque é realizado aplicando uma convolução entre a imagem $I(x, y)$ e uma função Gaussiana $G(x, y, \sigma)$, tal que

$$L(x, y, \sigma) = I(x, y) * G(x, y, \sigma), \quad (2.4)$$

onde $G(x, y, \sigma)$ é definido por

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}. \quad (2.5)$$

Para otimizar os resultados desse método, é aplicada também uma Diferença de Gaussianas (do inglês, *Difference of Gaussians*) ou *DoG*, dada pela subtração entre duas funções gaussianas em que uma delas possui uma constante de escalonamento k . Assim, tem-se uma *DoG* da seguinte forma

$$DoG(x, y, \sigma) = G(x, y, k\sigma) - G(x, y, \sigma). \quad (2.6)$$

Fazendo-se isso, é aplicado na imagem um filtro de suavização com parâmetros σ e $k\sigma$. O resultado da convolução entre a imagem e a *DoG* é dado por

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (2.7)$$

Logo,

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \quad (2.8)$$

Em seguida são detectados os pontos extremos. Para isso, um *pixel* específico na imagem é comparado com os *pixels* vizinhos e aquele que possuir maior *DoG* será um ponto extremo. É com esse método que se pode determinar quais pontos extremos na imagem são candidatos a pontos-chave (do inglês, *keypoints*). Os *keypoints* serão muito importantes no processo de reconstrução 3D, pois são deles as informações de posição no espaço.

O algoritmo SIFT já é implementado em uma das bibliotecas de processamento de imagens utilizada nesse projeto. É possível visualizar o resultado do algoritmo SIFT na Figura 3.

Figura 3 – Aplicação do algoritmo SIFT e representação dos seus *keypoints*



Fonte: Elaborado pelo autor.

Como a reconstrução 3D é obtida a partir de múltiplas imagens, é necessário extrair quais características são comuns entre duas imagens diferentes. Após extrair os últimos *keypoints* de uma imagem, faz-se necessário combiná-los para determinar quais características são referentes a um único ponto no espaço.

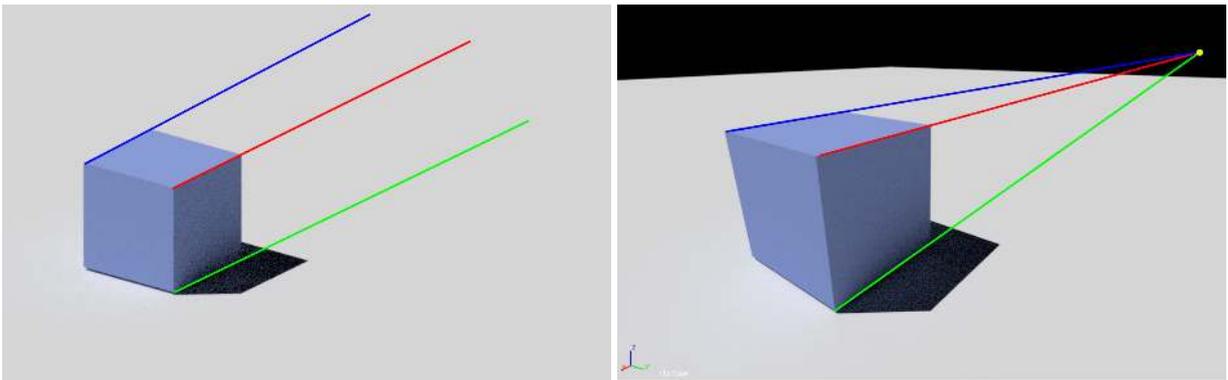
Será utilizado o processo de combinação por Força Bruta (do inglês, *Brute Force*), ou seja, as informações sobre um determinado *keypoint* são comparadas a outros *keypoints* de outra imagem analisando sua vizinhança. Aquele que obtiver resultados mais próximos do *keypoint* original será dado como seu equivalente.

2.4 Geometria de Visão Múltipla

A base para entender como funciona a reconstrução 3D tem início na Geometria de Múltipla Visão (*Multiple View Geometry*) tendo como a obra mais relevante nesse ramo atualmente o livro *Multiple View Geometry in Computer Vision* (HARTLEY; ZISSERMAN, 2003). Na obra são estudados os conceitos de projeção de imagens a partir de um ou mais pontos de vista, extração de informações em imagens, geometria da câmera e reconstrução 3D em si e que esses pontos serão discutidos a seguir baseando-se no livro citado.

Como já é conhecido pela álgebra linear, um plano de duas dimensões pode ser representado a partir das coordenadas (x,y) e o mesmo para um espaço de três dimensões representados por (x,y,z) . Entretanto, a forma que o olho humano e também as câmeras visualizam determinados pontos pode ser completamente diferente de como esses pontos estão realmente distribuídos no espaço. Isso se dá por conta da perspectiva. Dois exemplos disso podem ser representados pela geometria Euclidiana e Homogênea, representando em um plano 2D duas linhas paralelas no espaço 3D na geometria Euclidiana, as linhas serão paralelas em todos os pontos da imagem. Já na representação Homogênea, duas linhas infinitas tendem a se encontrar em um ponto específico da imagem, como mostra a Figura 4.

Figura 4 – Representação do mesmo objeto em coordenadas Euclidianas (esquerda) e Homogêneas (direita)



Fonte: Elaborado pelo autor.

Tanto o olho humano como a representação de uma imagem através de uma câmera são dados pelo sistema de coordenadas homogêneas. Um exemplo disso é a forma como dois trilhos ferroviários paralelos parecem se encontrar em um determinado ponto no horizonte. Portanto, é de extrema importância saber a diferença entre estes dois tipos de coordenadas, pois precisa-se saber como extrair a informação de profundidade em um determinado ponto da imagem.

Pelo fato de as informações relacionadas ao eixo Z serem representadas em um plano de projeção (PP), a representação de profundidade das coordenadas no mundo real (x, y, z) se dá $(x', y', -d)$, sendo d a distância entre o plano bidimensional e o ponto focal da câmera (POC). A representação matemática do ponto (x, y, z) em relação ao ponto $(x', y', -d)$, também abordada na Figura 5, é dada calculando seus triângulos similares, de tal forma que

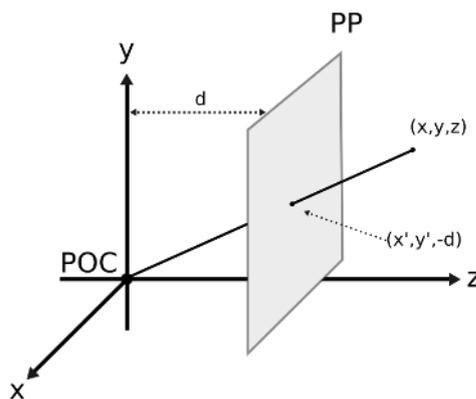
$$(x, y, z) = \left(-d\frac{x}{z}, -d\frac{y}{z}, -d\right). \quad (2.9)$$

De forma similar, o plano bidimensional da imagem digital é calculada da seguinte maneira:

$$(x', y') = \left(-d\frac{x}{z}, -d\frac{y}{z}\right). \quad (2.10)$$

É por este motivo que quando um determinado ponto se encontra a uma distância muito grande em relação a PP, ou seja, quando z tende a ser maior, menor será a projeção desse ponto na imagem.

Figura 5 – Projeção de coordenadas 3D em um plano 2D.



Fonte: Elaborado pelo autor.

É possível extrair informações de profundidade de um objeto em relação a duas imagens utilizando Geometria Epipolar. Essa geometria possibilita que duas imagens tiradas de um mesmo objeto possam compartilhar informações que determinam a localização desse objeto no espaço tridimensional. No livro de Hartley e Zisserman (2003) há um desenvolvimento teórico bastante completo sobre essa geometria, e também será comentada na subseção 2.4.2.

2.4.1 Parâmetros da Câmera

Uma câmera possui parâmetros intrínsecos como distância focal e ponto central de projeção. Já os parâmetros extrínsecos relacionam-se com parâmetros de localização da câmera e sua rotação.

Durante todo esse projeto e também em trabalhos de outros autores, o modelo de câmera utilizado foi o *pinhole camera* ou câmera estenopeica. Apesar de a forma como as câmeras digitais não extraírem as imagens da mesma maneira que câmeras estenopeicas a fazem, o modelo matemático utilizado é o mesmo para objetivos de extração de informações da câmera.

O que vale ressaltar nessa subseção é a forma como os parâmetros intrínsecos e extrínsecos da câmera são determinados. Toda câmera possui parâmetros específicos, que podem variar dependendo do seu modelo. As características contidas nesses parâmetros estão relacionados ao comprimento focal (ou *focal length*) nos eixos horizontal e vertical, sendo eles f_x e f_y , e também o ponto principal da imagem c_x e c_y . Segundo Hartley e Zisserman (2003), a matriz da câmera é definida por

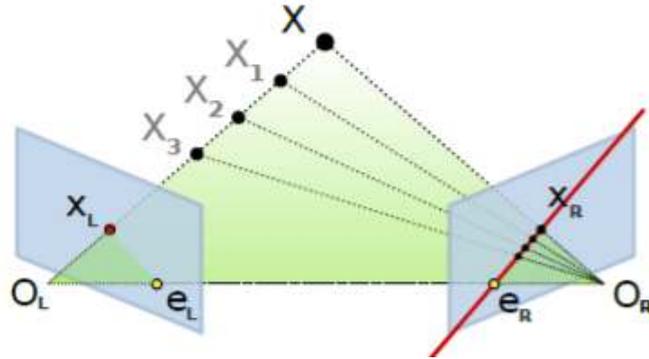
$$\mathbf{K} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

O comprimento focal f é a distância entre o sensor de ponto focal da câmera e a imagem projetada e o plano da imagem. É por esse motivo que quanto mais distante um objeto está, mais distante a projeção da imagem também deve estar do ponto focal da câmera para que esse objeto seja aumentado na imagem.

2.4.2 Geometria Epipolar

Com a utilização da Geometria Epipolar é possível determinar onde um determinado ponto se encontra em duas imagens diferentes, com ângulos e/ou distâncias variadas em relação ao ponto do cenário. Na Figura 6, onde os pontos centrais de projeção das câmeras são O_L e O_R , é possível visualizar o ponto X no espaço tridimensional em ambas as projeções. Na projeção da esquerda e direita esse ponto é representado por x_L e x_R respectivamente. Entretanto, os pontos x_1 , x_2 e x_3 correspondem ao mesmo ponto em relação a imagem da esquerda, enquanto estes mesmos pontos estão distribuídos em uma reta e_R na imagem a direita.

Figura 6 – Exemplificação do mapeamento de um ponto usando geometria epipolar.



Fonte: (MOULON *et al.*, 2013)

Pode-se relacionar os pontos x_L e x_R utilizando uma matriz 3×3 denominada Matriz Fundamental \mathbf{F} , de tal forma que

$$x_R^T \mathbf{F} x_L = 0. \quad (2.12)$$

Com a matriz fundamental é possível estimar a posição e o ângulo da câmera a partir de outra matriz 3×3 , chamada Matriz Essencial. A relação que se dá entre a matriz essencial \mathbf{E} e a matriz fundamental \mathbf{F} se dá da seguinte forma:

$$\mathbf{E} = \mathbf{K}_R^T \mathbf{F} \mathbf{K}_L. \quad (2.13)$$

Onde \mathbf{K}_R e \mathbf{K}_L são os parâmetros intrínsecos das câmeras, como distância focal e ponto central da câmera e podem ser determinadas no processo de calibragem.

Em SANTOS (2012), o autor traz vários conceitos apresentados aqui, assim como a calibragem de câmera, geometria epipolar e um processo detalhado de como são obtidas as características de uma câmera estenopeica.

Expandindo esses conceitos para utilização de mais de duas imagens, tem-se a técnica de reconstrução 3D chamada de Estruturação por Movimento (do inglês, *Structure from Motion*) ou *SfM*. Com essa técnica, são tratados muitos problemas como estimar a posição das câmeras no conjunto de fotografias, extração de características das imagens e triangulação dos pontos.

3 METODOLOGIA

Nesta seção serão discutidas os materiais utilizados no projeto, não só de *hardware* mas também de *software*. Em seguida, é relatado como se deu o desenvolvimento do projeto utilizando esses materiais.

3.1 Materiais

Serão utilizadas duas bibliotecas fundamentais para a execução da reconstrução 3D. Uma delas já mencionada é a OpenCV, e a segunda, direcionada especificamente para reconstrução tridimensional, a OpenMVG (MOULON *et al.*, 2013). Ambas bibliotecas são desenvolvidas na linguagem C++, sendo que a biblioteca OpenCV possui também suporte para as linguagens Java e Python.

3.1.1 OpenCV e OpenMVG

A biblioteca OpenCV foi desenvolvida pela companhia de tecnologia Intel por volta do ano 2000, não sendo apenas desenvolvida só para processamento de imagens mas também conta com ferramentas de Álgebra Linear e interface gráfica. Essa ferramenta. Além dos módulos de detecção e identificação de objetos, reconhecimento facial e detecção de movimentos (do inglês, *motion tracking*), a OpenCV também possui muitas funções que auxiliam na calibragem de câmera e correção de distorção. Existe um módulo de desenvolvimento na área de reconstrução 3D, entretanto não possui o ferramental suficientemente desenvolvido como a biblioteca OpenMVG. No artigo de Moulon *et al.* (2016), é dito que apenas implementações incompletas dessa área são desenvolvidas e não oferecem maior suporte para técnicas de estruturação por movimento.

A biblioteca OpenMVG é uma proposta similar a da biblioteca OpenCV. Porém, é uma biblioteca de visão computacional focada na solução de problemas que envolvem geometria de múltipla visão, extração de características na implementação de técnicas para estruturação por movimento e a reconstrução 3D. Essa ferramenta possui seu código aberto no GitHub¹, desenvolvida tanto pelos criadores do projeto como pela comunidade de programadores, assim como o suporte técnico prestado.

As duas ferramentas são bastante úteis para esse tipo de projeto. Todavia, dois pontos

¹ <https://github.com/openMVG/openMVG>

negativos contribuíram para que o desenvolvimento do *software* e o processamento das imagens não fossem feitos exclusivamente no dispositivo móvel:

- A biblioteca OpenMVG não é desenvolvida para a plataforma Android, apenas para Windows, Linux e MacOS.
- O tempo necessário para realizar a reconstrução 3D pode ser muito longo dependendo da quantidade de imagens e o poder de processamento dos dispositivos móveis ainda é inferior aos dos computadores desktop.

Apesar do melhoramento dos processadores da atual geração, ainda tem-se dificuldades em processar imagens de forma extremamente rápida pois trabalhar com imagens é uma tarefa que exige bastante poder de processamento das máquinas. Assim, foi desenvolvida uma metodologia de comunicação entre dispositivos onde o dispositivo móvel captura as imagens do objeto a ser reconstruído e uma máquina servidor executa o processamento dos dados.

As especificações de *hardware* se encontram na Tabela 1.

Tabela 1 – Especificações dos materiais.

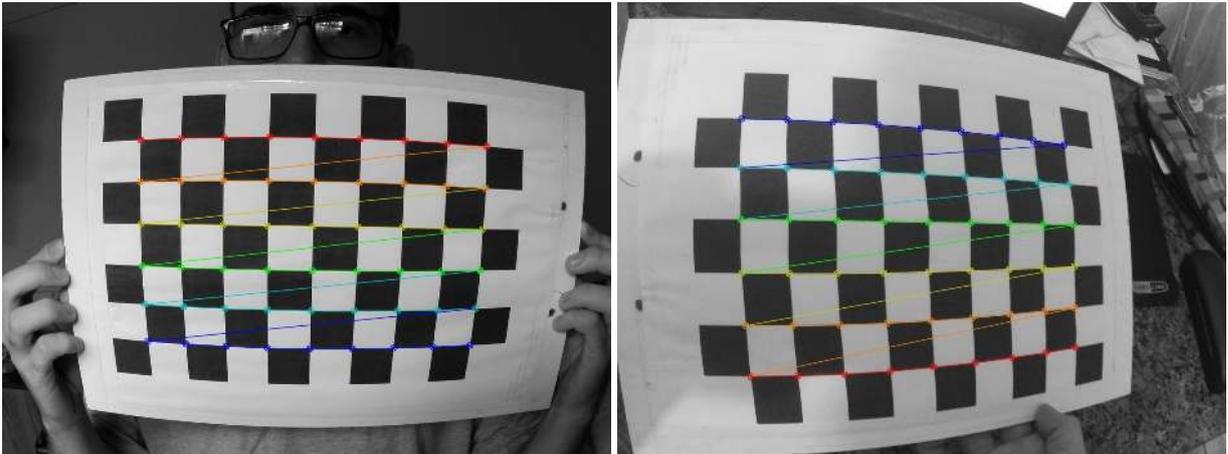
	Servidor	Dispositivo Móvel
Marca	Dell	Samsung
Modelo	Inspiron 14R	Galaxy J5
Sistema	Ubuntu 16.04	Android 6.0
Câmera	Não se Aplica	2048x1152 px
Processador	Intel Core i5 1.6GHz	Quad-Core 1.4GHz
Placa de Vídeo	Nvidia 740m 2GB	Não se Aplica
Memória Ram	4GB	2GB

Fonte: Elaborado pelo autor.

Nos testes realizados, foi utilizado uma rede local (LAN) (do inglês, *Local Area Network*), onde o aplicativo Android envia as imagens a serem recebidas por um roteador de modelo FiberHome AN5506-04-FG ONU.

Como foi citado na seção 2.2, na calibragem da câmera foi utilizado um padrão de textura xadrez de 9 por 6 cantos. Utilizando a biblioteca OpenCV, foi possível realizar a detecção dos pontos, determinar os parâmetros de distorção e por fim corrigir a imagem. A detecção dos cantos é representada na Figura 7, onde cada círculo colorido representa um canto.

Figura 7 – Calibragem da câmera usando a biblioteca OpenCV com padrões de xadrez de 9x6 quadrados.



Fonte: Elaborado pelo autor.

Ao serem encontrados os cantos de cada quadrado do padrão, uma linha horizontal é plotada passando por todos cantos selecionados. Com isso, é possível determinar a curvatura da lente e então saber o quão distorcida está a imagem.

Também é possível fazer a reconstrução 3D sem o uso da calibragem e correção das imagens. Apesar de não ser perceptível em muitos casos, todas as câmeras possuem um certo nível de distorção, e depois que é encontrado a magnitude dessa distorção e realizar a calibragem, a imagem distorcida pode ser corrigida. Quando não se sabe os parâmetros intrínsecos da câmera, não é possível realizar a correção da imagem, por menor que seja essa taxa de distorção. Contudo, ainda é possível realizar a reconstrução sem perda de informações significativas.

O aplicativo Android nomeado de *Reccoön 3D*, representado na Figura 8, possui uma interface com quatro botões, nos quais dois deles realizam capturas de imagens para calibragem para determinar os parâmetros da câmera e correção de imagem, e reconstrução para aplicar as técnicas de *SfM*. Já os outros dois realizam o processo de calibragem e reconstrução, enviando os dados para o servidor para serem processados.

Caso não seja feita a calibragem, o *script* de correção das imagens não será executado, e o conjunto de imagens será enviado diretamente para a pasta de reconstrução. Mas caso o usuário tenha acesso ao padrão de textura e fizer a calibragem, os dados intrínsecos da câmera serão armazenados para dispositivo *smartphone*. Como consequência, as imagens futuras tiradas por esse mesmo dispositivo poderão ser corrigidas e então enviadas para a pasta de reconstrução com uma taxa mínima de distorção, o que resultará em melhores resultados.

Quando as fotos são tiradas pelo dispositivo, utiliza-se o método para facilitar o trabalho assíncrono enquanto o dispositivo se comunica com o servidor, chamado de *AsyncTask*.

Figura 8 – Aplicativo Android para captura e envio das imagens a serem enviadas.



Fonte: Elaborado pelo autor.

As imagens são enviadas para o servidor através da rede enquanto o dispositivo aguarda os dados de resposta. Ao final do processo, o servidor retorna um arquivo .PLY (do inglês, *Polygon File Format*) ao dispositivo Android para que o usuário possa visualizar o resultado da reconstrução 3D. Duas aplicações onde é possível ver o resultado são o Meshlab² e o Emb3D³. A Figura 9 representa o envio dos dados do *smartphone* para o servidor através da internet e o retorno do modelo para o dispositivo móvel após a reconstrução.

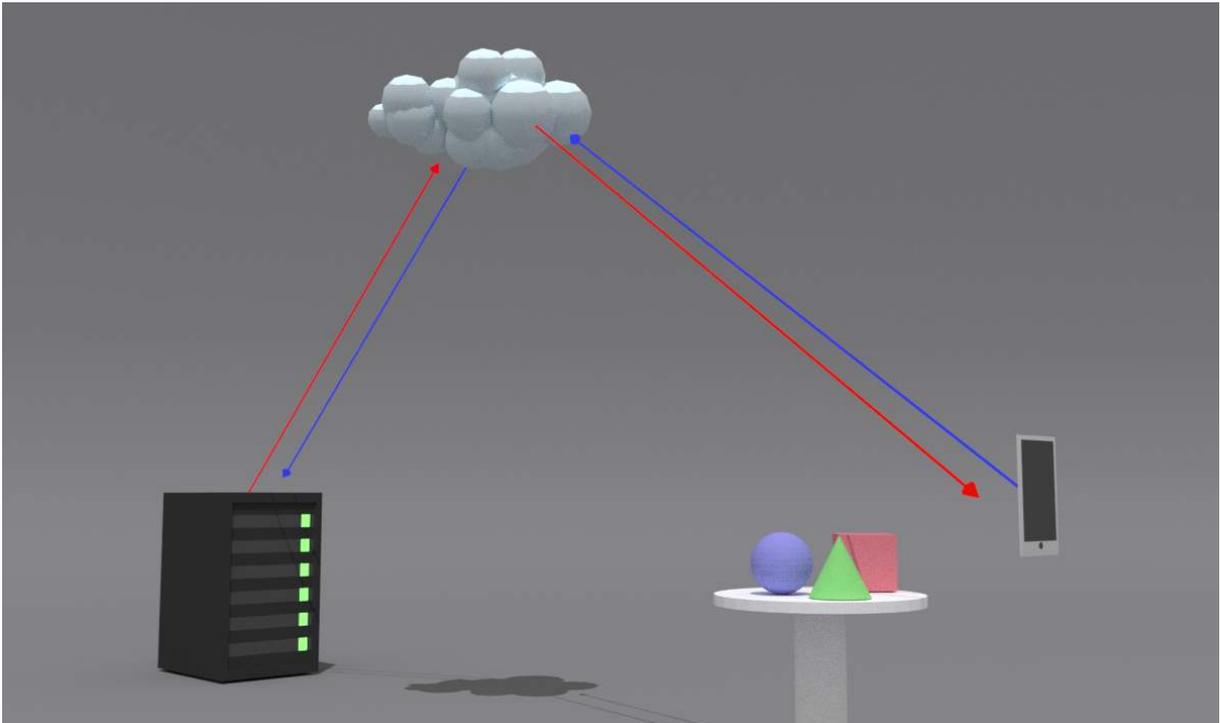
O funcionamento do aplicativo segue o fluxograma mostrado na Figura 10, onde o usuário tem as opções de calibragem ou reconstrução. Caso a câmera ainda não tenha sido calibrada, são tiradas as fotos para a execução desse processo, sendo necessária apenas uma vez para cada dispositivo *smartphone*. Após a calibragem, os parâmetros de uma determinada câmera são salvos e utilizados nas reconstruções seguintes. Caso a calibragem já tenha sido executada o usuário pode tirar as fotos do objeto que queira reconstruir para em seguida as imagens serem enviadas ao servidor, sendo processadas pelas bibliotecas de reconstrução, que ao final do processo retorna o arquivo 3D de volta ao *smartphone*.

Para os testes feitos nesse projeto, a câmera foi posicionada a uma distância de um metro em relação ao objeto. Entretanto, outros testes feitos com estruturas maiores mostram que o importante para uma boa reconstrução 3D é a qualidade das imagens e a maior quantidade de pontos equivalentes entre as imagens. Com isso, podem ser modelados objetos de pequeno

² www.meshlab.net

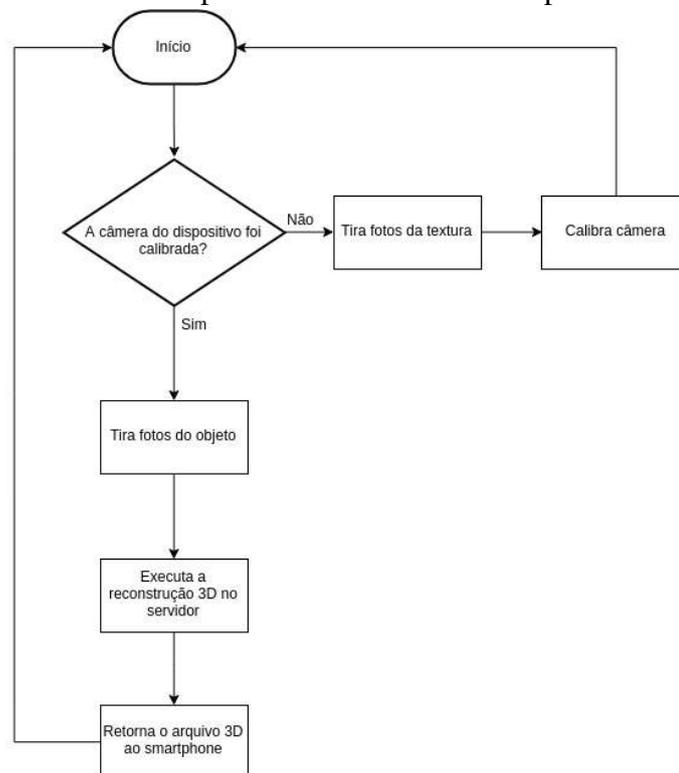
³ www.emb3d.com

Figura 9 – Aplicativo Android para captura e envio das imagens a serem enviadas.



Fonte: Elaborado pelo autor.

Figura 10 – Fluxograma com as etapas de funcionamento do aplicativo.



Fonte: Elaborado pelo autor.

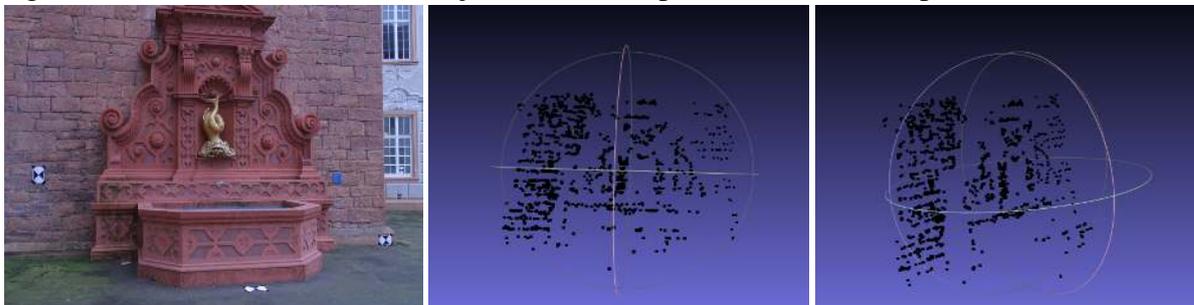
porte como os bonecos utilizados nesse projeto e também grandes estruturas como prédios e monumentos.

4 RESULTADOS

No início do desenvolvimento desse projeto, foram feitos alguns testes usando a biblioteca OpenCV, nas quais a imagem do modelo pode ser visto na Figura 11 a esquerda e os resultados nas figuras seguintes. Nesse exemplo, as imagens são de um conjunto de fotografias conhecido como fountain-P11, de resolução 3071×2048 px, referente a uma fonte suíça utilizada como referência para vários projetos de reconstrução 3D. Um ponto negativo é relacionado ao conjunto de imagens feito somente com duas imagens por vez, utilizando apenas os recursos de geometria epipolar.

Após todo um estudo das funcionalidades de cada biblioteca, percebe-se que a biblioteca OpenCV não é o mais recomendado para aplicações de reconstrução 3D devido a escassez de desenvolvimento nesse tema. Portanto essa biblioteca ficou responsável apenas pela calibragem e correção das câmeras, enquanto a biblioteca OpenMVG trabalhou diretamente com a reconstrução utilizando as técnicas de estruturação de movimento, permitindo um numero maior de imagens por vez.

Figura 11 – Resultado da reconstrução utilizando apenas a biblioteca OpenCV.

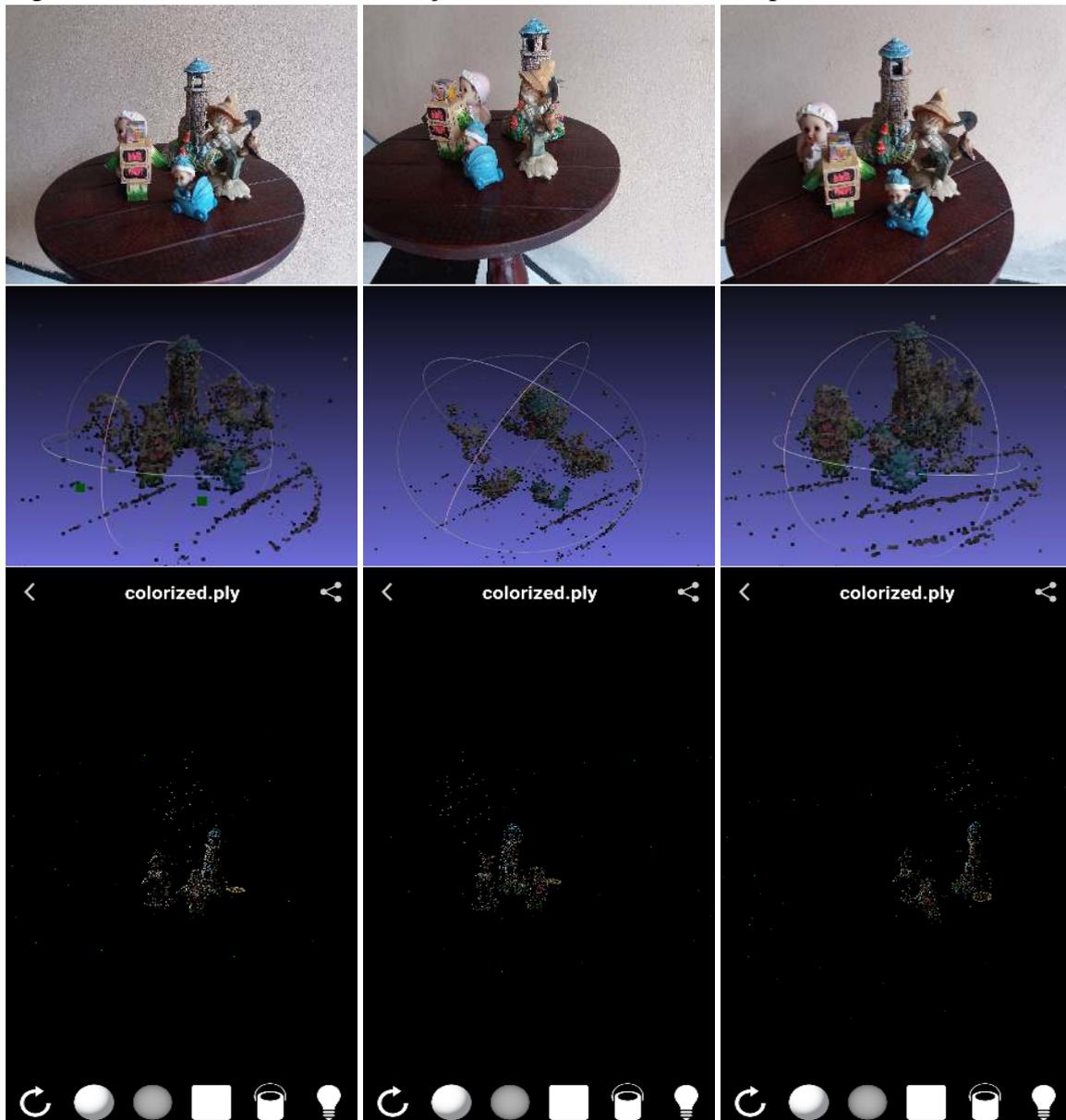


Fonte: Elaborado pelo autor.

Depois de obter esses resultados, foi preferido o uso da biblioteca OpenMVG devido ao melhor tratamento em reconstrução 3D. Para esse projeto, foram usados inicialmente 48 imagens de resolução 2048×1152 px. A comunicação entre o dispositivo e o servidor levou cerca de 3 segundos para o envio de cada imagem. O tempo total de processamento das imagens e aplicação dos algoritmos da biblioteca OpenMVG foram de 5 minutos e 49 segundos. Algumas imagens, juntamente com o resultado final são, apresentados na Figura 12. Nas três imagens superiores estão os bonecos de gesso utilizados como modelo para a reconstrução, e nas imagens logo abaixo estão os resultados da reconstrução visualizados de vistas diferentes no *software* Meshlab e no aplicativo Emb3D.

O modelo 3D é apresentado como um conjunto de pontos tridimensionais que

Figura 12 – Resultado da reconstrução utilizando a biblioteca OpenMVG



Fonte: Elaborado pelo autor.

ocupam o ambiente tridimensional que não possuem nenhum tipo de interligação. Esse modelo de vértices individuais é conhecido como Nuvem de Pontos (do inglês, *Point Cloud*), que diferente do modelo de mechas poligonais (do inglês, *polygon mesh*) onde seus vértices são interligados por arestas e formam uma face 3D.

É possível perceber uma melhora de qualidade entre os dois experimentos pois apesar de ter uma resolução um pouco menor, os resultados possuem uma nuvem de pontos relativamente mais preenchida em relação aos resultados do experimento da fonte suíça.

No arquivo apresentado na Figura 12 obteve-se um total de 12.597 vértices. O mesmo teste com as fotos da fonte suíça foi realizado utilizando a biblioteca OpenMVG, e enquanto a

OpenCV obteve um modelo 3D com 1.216 vértices, a OpenMVG retornou um resultado com 3.151 vértices. Como a forma de representação do modelo foi em *Point Cloud*, totalizou-se zero faces, uma vez que os vértices não são ligados por nenhuma aresta entre si.

Na obra de Batistela *et al.* (2017), os autores fazem a comparação de cinco ferramentas de reconstrução 3D, sendo elas o Photoscan, 123d Catch, ReCap 360, PPT-GUI e por fim, a OpenMVG juntamente com a OpenMVS. Nos testes, a OpenMVG possuiu o terceiro melhor resultado em relação a quantidade de vértices retornadas, inferior apenas ao Photoscan e o 123d Catch. Isso prova a eficiência da OpenMVG em comparação com outros *softwares* proprietários.

Outro requisito a ser ressaltado é o tempo de processamento. No atual projeto, o processamento levou pouco mais de cinco minutos. Na obra de Batistela *et al.* (2017) foram apresentadas comparações de tempo entre as ferramentas, mas não informando a quantidade exata de minutos. Entretanto, dentre as cinco ferramentas comparadas, a OpenMVG teve a melhor faixa de tempo, estando em um intervalo de 1 a 30 minutos e sendo considerada como rápida. O restante das ferramentas levou tempos medianos e longos, sendo divididos em 31 a 120 minutos e mais de 120 minutos, respectivamente.

Tanto o atual projeto como o trabalho de Batistela *et al.* (2017) utilizaram um conjunto de imagens similar, sendo que neste trabalho foram utilizados 48 imagens de um conjunto de bonecos de gesso sobre um pedestal, enquanto o de Batistela *et al.* (2017) foram 42 imagens de um crânio humano e ambos os projetos as imagens foram capturadas a partir de um *smartphone*.

Outra obra de comparação de resultados é o de Strupczewski e Czupryński (2014), onde os autores trazem uma abordagem que compara algumas ferramentas de reconstrução 3D, onde também são comentadas características de cada *software*. Nessa obra também são comentadas com mais profundidade a quantidade de testes feitos e imagens em diferentes resoluções.

Com base nesses trabalhos, é possível concluir que os resultados obtidos nesse projeto foram satisfatórios, já que a quantidade de imagens utilizadas e o tempo de processamento dessas imagens foram similares. E mais uma vez é possível reconhecer a eficiência da OpenMVG e a importância de projetos com código aberto que recebem implementações de diversas pessoas, de forma constante.

4.1 Pesquisa de Mercado

Acredita-se que com a chegada de computadores de melhor desempenho e mais acessíveis, o desenvolvimento de material 3D está se tornando cada vez mais acessível a todos. Com isso, há pretensão de tornar esse projeto modelo em um aplicativo comercializável, utilizando servidores pagos de empresas terceiras. Muitas das grandes empresas de tecnologia como Amazon, Google e Microsoft oferecem recursos de computação em nuvem (do inglês, *cloud computing*) que poderiam manter esse projeto para um grande número de pessoas em vários lugares do globo.

Foi realizada uma pesquisa nas lojas de aplicativos e empresas que alugam servidores para fazer um orçamento teórico de desse projeto. A loja virtual de aplicativos Google Play¹ informa em seu próprio site que aplicativos custam 25 dólares para serem publicados a todos os usuários da loja. Em relação aos servidores, foram procurados os serviços da empresa Amazon, onde o serviço chamado Amazon Elastic Cloud², os servidores são disponibilizados por hora e os preços variam dependendo do poder de processamento de cada máquina contratada.

Os servidores voltados para processamento de imagens estão classificados em instâncias de Copmutação Acelerada e os preços variam entre 0.45 até 13.464,00 dólares por hora. Os serviços da Google Cloud³, voltados para processamento de dados utilizando placas gráficas variam de 0.22 a 1.46 dólares. Por fim, os serviços da Microsoft Azure⁴ também possuem pacotes por hora que variam de 0.013 até 1.60 dólares por hora. Vale lembrar que a grande diferença de preços entre alguns serviços se dá pela configuração de *hardware* oferecida por cada empresa, onde cada máquina listada possui especificações diferentes, como quantidade de memória RAM, velocidade de processador e placas gráficas, armazenamento, entre outros aspectos. Apesar dos preços, todas as ferramentas citadas possuem versões gratuitas por um certo período de tempo.

Com isso, é possível concluir que a viabilidade desse projeto é dependente principalmente de um único fator que é a procura do público para a produção de material 3D. Calculando-se o valor desse projeto com os serviços citados anteriormente, tem-se uma fórmula básica de custos mensais, de tal forma que

$$\text{valor mensal} = (\text{valor por hora}) \times 24 \text{ horas} \times 30 \text{ dias.} \quad (4.1)$$

¹ www.developer.android.com/distribute/googleplay/

² www.aws.amazon.com

³ www.cloud.google.com

⁴ www.azure.microsoft.com

Dependendo da quantidade de pessoas ao redor no mundo interessadas na criação de material 3D de forma fácil e rápida, o projeto pode ser sustentado financeiramente.

5 CONCLUSÕES E TRABALHOS FUTUROS

Percebe-se que os resultados entre uma biblioteca de processamento de imagens específica para reconstrução 3D traz melhores resultados do que a implementação da biblioteca sem foco específico nessa área.

Contudo, foram obtidos bons resultados em relação a qualidade do modelo 3D. É possível perceber a fidelidade do modelo 3D em relação ao modelo real dos objetos da amostra. E o mais importante, que foi trazer essa aplicação a um dispositivo menor, modelando a ideia de que pessoas que não tem conhecimento na área de processamento de imagens ou reconstrução 3D podem gerar material tridimensional e manipulá-los em seus dispositivos móveis.

Em relação aos objetivos citados na seção 1.2, percebe-se que foi possível a criação de um aplicativo para dispositivo *mobile* que facilitou a experiência de reconstrução 3D utilizando apenas a câmera digital do dispositivo.

Apesar do poder de processamento dos computadores pessoais serem essenciais para suprir necessidades mais básicas como por exemplo o acesso a internet e consumo de vídeos ou músicas e dispositivos móveis ainda não serem os ideais para tarefas de processamento de imagens, a cada dia surgem novas ferramentas que facilitam a aquisição de material 3D nessa área. Computadores específicos para processamento de imagens que contam com processadores mais velozes e placas de vídeo dedicadas ao processamento dessas imagens com certeza são capazes de trazer resultados em um menor intervalo de tempo.

5.1 Trabalhos Futuros

Por ser um tópico em desenvolvimento apenas plataformas *desktop* tem uma maior preferência para desenvolvimento. Portanto, esse projeto pode servir como sugestão para trabalhos futuros no que se refere a desenvolvimento de bibliotecas nativas para sistemas de dispositivos móveis, como Android ou iOS.

REFERÊNCIAS

- BATISTELA, G. C.; MORAES, C. A. da C.; MIAMOTO, P. Comparação entre cinco sistemas de digitalização 3d por fotogrametria aplicados à antropologia forense e odontologia legal. **RBOL-Revista Brasileira de Odontologia Legal**, v. 4, n. 3, 2017.
- BAY, H.; TUYTELAARS, T.; GOOL, L. V. *Surf: Speeded up robust features*. v. 3951, p. 404–417, 2006.
- BAYER, B. E. *Color imaging array*. [S.l.]: Google Patents, 1976. US Patent 3,971,065.
- BRADSKI, G.; KAEHLER, A. *Learning OpenCV: Computer vision with the OpenCV library*. [S.l.]: "O'Reilly Media, Inc.", 2008.
- FOLHA, F. de S. P. *Smartphones estão nas mãos de 62% dos brasileiros*. 2017. Disponível em: <<http://www1.folha.uol.com.br/tec/2017/02/1862362-smartphones-estao-nas-maos-de-62-dos-brasileiros-diz-google.shtml>>. Acesso em: 8 fev. 2017.
- GONZÁLES, G. L. G. *Aplicação da Técnica SIFT para Determinação de Campos de Deformações de Materiais*. Tese (Doutorado) — PUC-Rio, 2010.
- HARTLEY, R.; ZISSERMAN, A. *Multiple view geometry in computer vision*. [S.l.]: Cambridge university press, 2003.
- LOWE, D. G. *Object recognition from local scale-invariant features*. v. 2, p. 1150–1157, 1999.
- MORDVINTSEV, A.; RAHMAN, A. *Camera Calibration*. 2013. <http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html>. Acesso em: 15 mar. 2017.
- MOULON, P.; MONASSE, P.; MARLET, R.; OTHERS. *OpenMVG. An Open Multiple View Geometry library*. 2013. <<https://github.com/openMVG/openMVG>>. Acesso em: 15 mar. 2018.
- MOULON, P.; MONASSE, P.; PERROT, R.; MARLET, R. *OpenMVG: Open multiple view geometry*. p. 60–74, 2016.
- RUBLEE, E.; RABAU, V.; KONOLIGE, K.; BRADSKI, G. *ORB: An efficient alternative to SIFT or SURF*. p. 2564–2571, 2011.
- SANTOS, M. *Revisão de Conceitos em Projeção, Homografia, Calibração de Câmera, Geometria Epipolar, Mapas de Profundidade e Varredura de Planos*. 2012.
- STRUPCZEWSKI, A.; CZUPRYŃSKI, B. *3D reconstruction software comparison for short sequences*. v. 9290, p. 929030, 2014.
- VILLEGAS, A. O controle da cor: Gerenciamento de cores para fotografos. **Santa Catarina**, 2009.
- VILLIERS, J. P. D.; LEUSCHNER, F. W.; GELDENHUYS, R. *Centi-pixel accurate real-time inverse distortion correction*. v. 7266, p. 726611, 2008.