

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

João Marcos Simões Ribeiro

**Uma Proposta Experimental de Reconstrução 3D a Partir de Imagens
Ortogonais**

Juiz de Fora

2020

João Marcos Simões Ribeiro

Uma Proposta Experimental de Reconstrução 3D a Partir de Imagens
Ortogonais

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, área de concentração: Sistemas de Energia Elétrica, da Faculdade de Engenharia, Universidade Federal de Juiz de Fora como requisito parcial para obtenção do grau de Mestre.

Orientador: Alexandre Bessa dos Santos, D. Sc.

Coorientador: Leonardo de Mello Honório, D. Sc.

Juiz de Fora

2020

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Ribeiro, João Marcos Simões.

Uma Proposta Experimental de Reconstrução 3D a Partir de Imagens Ortogonais / João Marcos Simões Ribeiro. – 2020.

101 f. : il.

Orientador: Alexandre Bessa dos Santos, D. Sc.

Coorientador: Leonardo de Mello Honório, D. Sc.

Dissertação de Mestrado – Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2020.

1. Reconstrução 3D a partir de imagens 2D. 2. *Structure From Motion*. 3. *Multi-View Stereo*. 4. UAV. I. Dos Santos, Alexandre Bessa, orient. Honório, II. Leonardo de Mello, co-orient. III. Reconstrução 3D a partir de um conjunto de imagens 2D de um cenário sobrevoado por um UAV

João Marcos Simões Ribeiro

Uma Proposta Experimental de Reconstrução 3D a Partir de Imagens
Ortogonais

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, área de concentração: Sistemas de Energia Elétrica, da Faculdade de Engenharia, Universidade Federal de Juiz de Fora como requisito parcial para obtenção do grau de Mestre.

Aprovada em: 10 de Março de 2020

BANCA EXAMINADORA



Prof. Dr. Alexandre Bessa dos Santos, D. Sc.
Universidade Federal de Juiz de Fora, UFJF
Orientador



Prof. Dr. Leonardo de Mello Honório, D. Sc.
Universidade Federal de Juiz de Fora, UFJF
Coorientador



Mário Dantas (Jun 4, 2020 11:58 ADT)

Prof. Dr. Mário Antonio Ribeiro Dantas, D. Sc.
Universidade Federal de Juiz de Fora, UFJF



Prof. Dr. André Luis Marques Marcato, D. Sc.
Universidade Federal de Juiz de Fora, UFJF



Prof. Dr. Carlos Henrique Valério de Moraes,
D. Sc.
Universidade Federal de Itajubá, UNIFEI

*Dedico este trabalho a minha Família,
estando comigo em todos
os momentos de dificuldade,
compartilhando alegrias e conquistas.*

AGRADECIMENTOS

Ao meu pai João Simões Neves e minha mãe Teresinha do Socorro Ribeiro Simões, pela educação, dedicação e base para poder desenvolver meus estudos da melhor forma possível.

A minha irmã Jordana por ajudar em todos os meus trabalhos.

A todos os meus familiares pelo incentivo e dedicação durante esse período.

Ao meu orientador Alexandre Bessa e coorientador Leonardo Honório, pela atenção e tempo dedicados no trabalho.

A todos os amigos do GRIn, pelos bons momentos e ajuda durante todo o processo.

Aos amigos, pelo apoio e incentivo nesse processo e pelos bons momentos que me proporcionaram.

RESUMO

Os avanços na última década na área da visão computacional trouxeram grandes benefícios para aplicações de realidade virtual, tendo a fotogrametria como um grande impulsionador. Essa técnica se popularizou após o surgimento dos UAVs, agregando melhorias a áreas como a engenharia civil, agronomia e geociência possibilitando realizar um mapeamento de uma região específica com missões aéreas. Neste contexto, este trabalho apresenta uma proposta de realizar a reconstrução 3D em escala real de cenários sobrevoados por um UAV. Para isso, foi desenvolvida uma abordagem em C++ capaz de realizar essa reconstrução para um conjunto de imagens não ordenadas. A metodologia *Struct From Motion* foi utilizada para realizar o cálculo da odometria das câmeras no momento da captura da imagem. Por fim, a técnica *Multi-View Stereo* é aplicada para obtenção de uma nuvem densa de pontos do cenário. Como metodologias inovadoras, tem-se a utilização das coordenadas de GPS de cada imagens para obtenção de uma reconstrução em escala real e um ajuste durante a propagação para melhorar o desempenho em regiões com pouca variação de cor. Os resultados obtidos foram satisfatórios, onde a reconstrução utilizando as imagens aéreas apresentou um resultado definido e com medidas similares as encontradas no Google Maps, obtendo um erro máximo de 1,59%.

Palavras-chave: Reconstrução 3D a partir de imagens 2D. *Structure From Motion*. *Multi-View Stereo*. UAV.

ABSTRACT

Advances in the last decade in the field of computer vision have brought great benefits to virtual reality applications, with photogrammetry as a major driver. This technique became popular after the emergence of UAVs, adding improvements to areas such as civil engineering, agronomy, and geoscience, making it possible to map a specific region with aerial missions. In this context, this work presents a proposal to perform 3D reconstruction real scale scenarios overflowed by a UAV. For this, an approach in C++ was developed capable of performing this reconstruction for a set of unordered images. The Struct From Motion methodology was used to perform the calculation of the odometer of the cameras at the time of image capture. Finally, the Multi-View Stereo technique is applied to obtain a dense point cloud of scenery. As innovative methodologies, the GPS coordinates of each image are used to obtain a real scale reconstruction and an adjustment during propagation to improve performance in regions with little color variation. The results obtained were satisfactory, where the reconstruction using aerial images presented a defined result and with measures similar to those found on Google Maps, obtaining a maximum error of 1,59%.

Key-words: 3D reconstruction from 2D Images. Structure From Motion. Multi-View Stereo. UAV.

LISTA DE ILUSTRAÇÕES

Figura 1 – Imagem digital vista pelo computador.	19
Figura 2 – Padrão RGB.	20
Figura 3 – Formação da imagem em uma câmera com projeção central. . .	20
Figura 4 – <i>Features</i> encontradas utilizando o método de Harris.	22
Figura 5 – Desempenho dos descritores LDB e M-LDB.	24
Figura 6 – Demonstração do eixo ótico e do ponto principal no plano da imagem.	26
Figura 7 – Calibração de uma câmera com auxílio de um objeto de dimensões conhecidas.	28
Figura 8 – Detecção dos pontos de interesse de um tabuleiro de xadrez para a calibração da câmera.	29
Figura 9 – Mapeamento de pontos utilizando uma transformada homográfica.	32
Figura 10 – Construção panorâmica a partir de duas imagens.	32
Figura 11 – RANSAC aplicado a um conjunto de dados bidimensional. . . .	34
Figura 12 – Plano e linha epipolar entre duas imagens.	35
Figura 13 – Comportamento das linhas epipolares.	37
Figura 14 – Posição do ponto para cada configuração da câmera.	40
Figura 15 – Erro de aferição na captura das Imagem.	41
Figura 16 – Representação de um <i>Template</i>	44
Figura 17 – Erro na seleção do <i>Template</i> gerado pela rotação e translação da câmera.	46
Figura 18 – <i>Templates</i> de um mesmo ponto para imagens diferentes.	46
Figura 19 – Seleção de <i>Template</i> com a transformação <i>Affine</i>	48
Figura 20 – <i>Templates</i> selecionados com a transformação <i>Affine</i> para diferentes normais.	48
Figura 21 – <i>Templates</i> selecionados utilizando melhor configuração da Normal.	49
Figura 22 – Configuração do <i>Template</i> para propagação de pontos.	50
Figura 23 – Identificação de vizinhos no espaço tridimensional.	51
Figura 24 – Formas Geométricas utilizadas para filtra <i>outliers</i>	54
Figura 25 – Reconstruções da três campanhas realizada por Meinen.	55
Figura 26 – Nuvem de Pontos obtida por Pagan.	55
Figura 27 – Reconstruções de toda a área mapeada por Warfield.	56
Figura 28 – Modelo digital do terreno levantado por Pijl.	57
Figura 29 – Drone realizando uma missão.	59
Figura 30 – Phantom 4 PRO.	59
Figura 31 – Tabuleiro utilizado para calibração.	60

Figura 32 – Exemplo da deformação de um objeto, ocasionada pela estimação errada da matrizes de Pose das câmeras. O objeto a esquerda corresponde forma real e o da direita o objeto deformado.	61
Figura 33 – Exemplo dos diferentes comportamento ao realizar a propagação.	65
Figura 34 – Limites para expansão de uma <i>feature</i>	67
Figura 35 – Imagens utilizadas para a calibração da câmera.	69
Figura 36 – Imagens obtidas pelo Phantom 4 em uma missão sobre a engenharia da UFJF.	70
Figura 37 – Representação 2D da área mapeada.	71
Figura 38 – Imagens utilizadas no primeiro cenário.	72
Figura 39 – Pares de imagens reconstruídos do primeiro cenário.	72
Figura 40 – Nuvem de pontos sobrepostas do primeiro cenário.	73
Figura 41 – Imagens retiradas do campo de engenharia da UFJF.	74
Figura 42 – Pares de imagens reconstruídos do segundo cenário.	74
Figura 43 – Nuvem de pontos sobrepostas do segundo cenário.	75
Figura 44 – Nuvem de pontos sobrepostas do segundo cenário de diferentes vistas.	76
Figura 45 – Nuvem de pontos sobrepostas do segundo cenário.	76
Figura 46 – Subconjunto das imagens obtidas na missão do Phantom 4.	76
Figura 47 – Nuvem de pontos obtida no terceiro cenário.	77
Figura 48 – Nuvem de pontos do terceiro cenário de diferentes vistas.	77
Figura 49 – Nuvem de pontos obtida no quarto cenário.	78
Figura 50 – Nuvem de pontos do quarto cenário de diferentes vistas.	79
Figura 51 – Nuvem de pontos após a aplicação do filtro.	80
Figura 52 – Nuvem de pontos filtrada de diferentes vistas.	80
Figura 53 – Nuvem de pontos do quinto cenário.	81
Figura 54 – Nuvem de pontos do quinto cenário de diferentes vistas.	82
Figura 55 – Comparação entre o desempenho dos algoritmos observados da mesma posição.	83
Figura 56 – Uma segunda vista comparando o desempenho entre os algoritmos.	83
Figura 57 – Reconstrução utilizando todas imagens obtida pela missão do Phantom 4.	84
Figura 58 – Diferentes vistas da nuvem obtida na reconstrução com todas imagens.	84
Figura 59 – Reconstrução após a aplicação da técnica de <i>Poisson</i>	85
Figura 60 – Imagens obtidas no Google Maps com a medida do comprimento entre pontos da cena em amarelo.	86
Figura 61 – Medidas realizadas sobre a nuvem de pontos obtidas.	87

LISTA DE FLUXOGRAMAS

Fluxograma 1 – Estratégias adotada no trabalho.	58
Fluxograma 2 – Estimação de odometria da câmera.	64
Fluxograma 3 – Obtenção da nuvem densa.	68

LISTA DE TABELAS

Tabela 1	– Exemplos de pontos correspondente em um conjunto de três imagens.	62
Tabela 2	– Exemplos de pontos correspondente mapeados em um conjunto de três imagens.	62
Tabela 3	– Parâmetros intrínsecos da câmera utilizada.	70
Tabela 4	– Número de pontos obtidos na reconstrução para o primeiro cenário.	73
Tabela 5	– Número de pontos obtidos na reconstrução para o segundo cenário.	75
Tabela 6	– Número de pontos detectados e uso de memória no terceiro cenário.	78
Tabela 7	– Número de pontos detectados e uso de memória no quarto cenário.	79
Tabela 8	– Número de pontos detectados e uso de memória com a aplicação do filtro.	80
Tabela 9	– Número de pontos detectados e uso de memória no quinto cenário.	82
Tabela 10	– Número de pontos detectados e uso de memória no sexto cenário.	85
Tabela 11	– Comparação entre as mediadas obtidas no Google Maps e os valores estimados na reconstrução.	87
Tabela 12	– Parâmetros extrínsecos de cada câmera.	100

SUMÁRIO

	LISTA DE FLUXOGRAMAS	10
1	INTRODUÇÃO	14
1.1	MOTIVAÇÃO	16
1.2	OBJETIVOS GERAIS	17
1.2.1	Objetivos Específicos	17
1.3	ESTRUTURA DO TRABALHO	17
2	REVISÃO BIBLIOGRÁFICA	19
2.1	IMAGENS	19
2.1.1	Formação das imagens	20
2.1.2	Pontos de Interesse - <i>Features</i>	21
2.1.3	Descritores e <i>Match</i> de Pontos	23
2.2	Modelagem da Câmera	25
2.2.1	Parâmetros Intrínsecos	25
2.2.2	Parâmetros Extrínsecos	27
2.2.3	Calibração da Câmera	28
2.3	<i>Structure From Motion</i> (SFM)	30
2.3.1	Homografia	31
2.3.2	<i>Random Sample Consensus</i> (RANSAC)	34
2.3.3	Geometria Epipolar	35
2.3.3.1	<i>Matriz Fundamental</i>	36
2.3.3.2	<i>Matriz Essencial</i>	38
2.3.4	Cálculo da Matriz de Pose da Câmera	39
2.3.5	Triangulação	40
2.3.6	<i>Bundle adjustment</i> (BA)	42
2.4	<i>Multi-View Stereo</i> (MVS)	43
2.4.1	<i>Zero Mean Normalized Cross-Correlation</i> (ZNCC)	43
2.4.2	Transformação <i>affine</i>	45
2.4.3	Propagação dos Pontos	49
2.4.4	Ajuste da Nuvem de Pontos	51
2.4.5	Filtragem da Nuvem de Pontos	53
2.5	APLICAÇÕES ENVOLVENDO AS TÉCNICAS SE SFM E MVS EM CONJUNTO COM UM UAV	54
3	METODOLOGIA	58
3.1	CAPTURA DAS IMAGENS E CALIBRAÇÃO DA CÂMERA	58
3.2	CÁLCULO DA ODOMETRIA DA CÂMERA	60
3.3	ESTIMAÇÃO DA NUVEM DENSA	64
4	RESULTADOS E DISCUSSÃO	69

4.1	CALIBRAÇÃO DA CÂMERA	69
4.2	CÁLCULO DA ODOMETRIA	70
4.3	NUVEM DE PONTOS	71
4.3.1	Cenário 1	71
4.3.2	Cenário 2	74
4.3.3	Cenário 3	76
4.3.4	Cenário 4	78
4.3.5	Cenário 5	81
4.3.6	Cenário 6	84
4.4	Comparação entre Medidas do Google Maps e a Nuvem Calculada . . .	86
5	CONCLUSÕES	89
5.1	TRABALHOS FUTUROS	89
	REFÊRENCIA	91
	APÊNDICE A – Parâmetros Extrínsecos das Câmeras	100

1 INTRODUÇÃO

A humanidade possui grandes habilidades, que permite adaptar-se a diversos meios. Dentre essas funcionalidades, a visão tem um papel fundamental de auxiliar na interação do indivíduo com o ambiente, ajudando na mobilidade, locação, orientação, comunicação e controle de ações (FAVILLA et al., 2014). Por esses motivos, a visão é considerada o sentido mais forte e valioso do ser humano. Seus olhos são capazes de detectar variações de cores, além de uma visão estereoscópica, que permite uma observação nítida de corpos tridimensionais (CAMPOS, 2008). Vale ressaltar que a visão estereoscópica é responsável por obter informações do espaço tridimensional analisando duas imagens obtidas de pontos distintos.

Com o intuito de automatizar tarefas que dependem da percepção humana, surgiu a visão computacional, área que compreende um conjunto de processos aplicados a imagens digitais, tais como: aquisição, processamento, análise e compreensão (MUNDY et al., 1992). Essa área, muitas vezes, exige um poder de processamento elevado, devido a grande quantidade de informações obtidas a partir de uma imagem (COUTO, 2012).

Neste contexto, Laussedat, considerado o pai da fotogrametria, em 1850 desenvolveu um método para construir um mapa da cidade de Paris, extraindo informações geométricas de fotografias retiradas dos telhados da cidade. Laussedat foi capaz de fazer essa reconstrução utilizando uma técnica de interseção para realizar análise de pares estereoscópicos ponto a ponto, além do conhecimento da distância entre as câmeras (ATKINSON, 1996).

Essa técnica foi aprimorada com o surgimento da câmera digital, onde os computadores passaram a ser capazes de receber informações de um ambiente pelas imagens digitais, promovendo o surgimento de algumas metodologias de reconstrução 3D (DE FARIAS, 2012). Dentre elas, destacam-se: *Structure From Motion* (SFM), utilizada para obter informações sobre as características e localizações das câmeras e *Multi-View Stereo* (MVS), responsável pela obtenção do pontos tridimensionais. Essas técnicas são utilizadas em cenários estáticos com a finalidade de obter um conjunto de pontos tridimensionais, denominado nuvem de pontos. Esses pontos podem ser utilizados tanto para visualização, como para extração de informações de um determinado cenário.

A metodologia SFM possui os mesmos princípios básicos da fotogrametria estereoscópica, ou seja, é possível obter a estrutura 3D a partir de uma série de imagens sobrepostas. Entretanto, diferente da fotogrametria convencional, a geometria da cena, as posições e a orientação da câmera são calculadas automaticamente, sem a necessidade de especificar *a priori*, um conjunto de pontos que possuem informações do espaço tridimensional. Esses cálculos são realizados simultaneamente, utilizando um procedimento iterativo de ajuste de limites e são baseados em um banco de dados de características extraídas automaticamente de um conjunto de imagens sobrepostas (HARTLEY; ZISSERMAN,

2003).

O SFM retorna uma nuvem de pontos espaçada, normalmente de pontos que possuem um grande contraste com seus vizinhos. Sendo assim, faz-se necessário a utilização da técnica do MVS, para a aquisição de uma nuvem mais densa. O MVS é capaz de construir uma geometria plausível em 3D, altamente detalhada, a partir de conjunto de imagens que, codifica a figura sob algumas exigências razoáveis, dentre as quais se destaca a rigidez da cena (FURUKAWA; HERNÁNDEZ, 2015).

O SFM-MVS é uma abordagem barata e eficaz para captura de topografias complexas (WESTOBY et al., 2012). Sua contribuição para a geociência é ratificada por (SMITH; CARRIVICK; QUINCEY, 2016), que questiona a dificuldade de encontrar algoritmos com código aberto e incentiva o desenvolvimento de novas estratégias para este segmento.

São inúmeras as aplicações de reconstrução 3D a partir de imagens 2D na atualidade. (BADURA et al., 2019) utiliza a técnica de reconstrução para mapear o crescimento de plantas e, a partir destes dados, definir melhores técnicas de cultivo. (KURZ et al., 2019) apresenta o mapeamento de rodovias utilizando a técnica de MVS, para auxiliar na auto-localização de veículos autônomos.

Um estudo de mudança de topografias, abordado por (RIQUELME et al., 2019), georreferência as deformações ocorridas com o auxílio de pontos de controle no solo, comparando fotos aéreas atuais de alta resolução, como imagens históricas. Ainda neste segmento, (ZIMMER et al., 2018) e (SWIRAD; ROSSER; BRAIN, 2019) utilizam a reconstrução para quantificar a erosão ocorrida no solo.

Esses estudos se intensificaram após a popularização dos veículos aéreos não tripulados, *Unmanned Aerial Vehicles* (UAV), havendo um crescimento na pesquisa com sua utilização em conjunto com a técnica de reconstrução SFM-MVS (LUCIEER; ROBINSON; TURNER, 2010; MANCINI et al., 2013; LUCIEER; TURNER et al., 2014; GÓMEZ-CANDÓN; DE CASTRO; LÓPEZ-GRANADOS, 2014; GRAHAM et al., 2019). Essa metodologia é atraente por produzir um conjunto de dados de alta resolução além de cobrir grandes áreas, dificilmente realizadas a partir de técnicas terrestres (NESBIT; HUGENHOLTZ, 2019). Outro benefício importante desta tecnologia é a capacidade de mapear áreas inacessíveis ou perigosas (THOENI et al., 2016).

Essas pesquisas possibilitaram avanços em diversas áreas, como por exemplo, estratégia para o manejo de pastagem (FORSMO et al., 2018), monitoramento de fluvial e aquático (CARRIVICK; SMITH, 2019), mapeamento e monitoramento de mudanças detalhadas em um ambiente proglacial (EWERTOWSKI et al., 2019), avaliações de precisão de mapas temáticos em ambientes complexos (FRASER; CONGALTON, 2019), recriação de um complexo arquitetônico (CARVAJAL-RAMIREZ et al., 2019), reconstrução de construção (ALIDOOST; AREFI, 2015; MALIHI et al., 2016), entre outros.

Apesar do grande número de aplicações, essa metodologia possui alguns pontos a serem aperfeiçoados. Ao analisar o objeto de uma imagem, sem o conhecimento de suas dimensões, não é possível determinar se o objeto estava próximo ou longe da câmera quando a imagem foi capturada. Para contornar esse problema, ao calcular a posição das câmeras pelo processo do SFM, são utilizados valores normalizados (HARTLEY; ZISSERMAN, 2003).

Esses valores normalizados permitem obter configurações para as câmeras que preservem os formatos dos objetos, entretanto, suas escalas não são mentidas, ou seja, suas dimensões estimadas são diferentes do mundo real (HARTLEY; ZISSERMAN, 2003).

Trabalhos que necessitam de precisão das medidas dos cenários reconstruídos, como apresentados por (RIQUELME et al., 2019), (ZIMMER et al., 2018), (SWIRAD; ROSSER; BRAIN, 2019), e (FRASER; CONGALTON, 2019), adicionam pontos de controle no solo, que são utilizados no pós-processamento para a obtenção da escala real. No presente trabalho, foi apresentado uma nova abordagem, onde esse cálculo é realizado utilizando as coordenadas do GPS obtidas junto as imagens.

Outro problema frequente na reconstrução 3D a partir de imagens é percebido em regiões que não é possível encontrar características únicas, distinguíveis e constantes entre as imagens, ou seja, com baixo valor de variância. A identificação de correspondência entre as imagens nessas regiões é prejudicada, fazendo com que seja obtidos ruídos nessas regiões, podendo em alguns casos gerar planos deslocados (FÖRSTNER; WROBEL, 2016).

Essas regiões são encontradas em objetos sem textura, como por exemplo parede em branco, tigela branca. Um problema similar é encontrado em objetos que repetem o mesmo padrão em diferentes partes, gerando correspondências erradas entre as imagens (SOLEM, 2012). A fim de minimizar o ruído nessas regiões, foi proposto o cálculo dos limites de propagação durante o processo MVS.

1.1 MOTIVAÇÃO

A reconstrução 3D possui aplicações em diversas áreas, como inspeção de áreas de risco, engenharia civil, entre outros. O avanço da tecnologia, tais como, surgimento das câmeras digitais, dos drones e o aumento de processamento dos computadores, possibilitou que essa técnica ganhasse novas aplicações e desafios. A utilização da reconstrução 3D a partir de imagens possui algumas limitações, como a dificuldade para reconstruir espaços com baixo valor de variação, dificultando distinguir um ponto dos demais, além de não preservar a escala real do cenário.

Diante disso, faz-se necessário a aprofundamento dessas técnicas de reconstrução, com o intuito de adquirir cenários mais robustos e melhor representatividade do real. Isso motiva o desenvolvimento do presente trabalho, de forma a apresentar metodologias

capazes de realizar a reconstrução 3D a partir de imagens 2D em escala real, mesmos em cenários que apresentam dificuldades para distinguir cada ponto.

1.2 OBJETIVOS GERAIS

O objetivo desse trabalho é desenvolver uma abordagem capaz de gerar uma nuvem de pontos, densa e robusta, reconstruindo cenas de locais abertos a partir de múltiplas fotos de uma câmera *Red-Green-Blue* (RGB). Deve ser realizada uma reconstrução de uma parte da Universidade Federal de Juiz de Fora, referente ao campus da engenharia. Um Phantom 4 PRO (DJI, 2020) foi utilizado para as capturas das imagens e posteriormente a reconstrução é computada de modo *offline*.

1.2.1 Objetivos Específicos

- a) Desenvolver uma abordagem capaz de estimar a odometria das câmeras, utilizando um conjunto de imagens não ordenados;
- b) Desenvolver uma abordagem capaz de obter medidas similares ao valor real do cenário, utilizando coordenadas do *Global Position System* (GPS);
- c) Desenvolver uma abordagem capaz de manter seu desempenho em locais abertos, desde que possua a luminosidade necessária para identificação dos objetos da imagem;
- d) Desenvolver uma abordagem capaz de executar um número de imagens relativamente alto durante sua execução.

1.3 ESTRUTURA DO TRABALHO

No Capítulo 2 são apresentados conceitos importantes e necessários para o desenvolvimento deste trabalho, tais como as propriedades de uma imagem, uma breve descrição da modelagem de uma câmera digital, além de conceitos geométricos epipolares.

O Capítulo 3 apresenta as estratégias utilizadas para a reconstrução tridimensional, destacando a calibração da câmera, cálculo da odometria e propagação de pontos para obtenção da nuvem densa.

Reconstruções de diferentes imagens e configurações são apresentados no Capítulo 4, juntamente com número de pontos identificados em cada metodologia além de seu consumo computacional. Uma comparação entre os resultados obtidos é realizada.

O Capítulo 5 apresenta as conclusões gerais deste trabalho e propostas para trabalhos futuros.

O Apêndice A apresenta as odometrias da câmera no momento da captura de cada imagem obtida em uma missão do UAV.

2 REVISÃO BIBLIOGRÁFICA

Nesse capítulo serão abordados fundamentos teóricos das imagens digitais, bem como princípios geométricos para sua manipulação. A Seção 2.1 faz uma breve revisão da formação das imagens. Na Seção 2.2 é apresentado a modelagem de uma câmera de projeção central. Uma revisão sobre a metodologia *Structure From Motion* (SFM) é realizada na seção 2.3. Na Seção 2.4 é mostrado alguns conceitos sobre a técnica de *Multi-View Stereo* (MVS). Por fim, na Seção 2.5 é feito uma revisão de trabalhos que foram desenvolvidos no último ano utilizando essas metodologias.

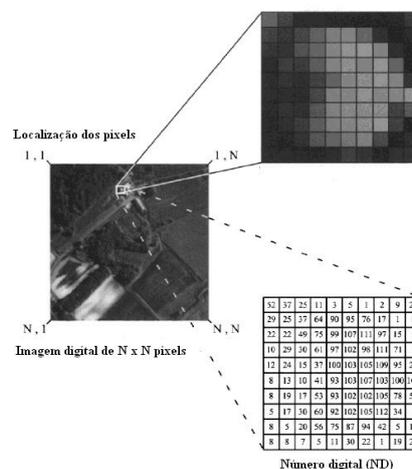
2.1 IMAGENS

As imagens são peças fundamentais para reconstrução 3D. Nelas estão contidas todas as informações necessárias para a obtenção da nuvem de pontos, onde uma imagem é interpretada pelo computador como uma matriz de números, que representam a coloração da imagem.

Ao capturar uma imagem digital, a luz proveniente do objeto é apanhada por um conjunto de lentes esféricas e projetada no fundo da câmera. A luz é transformada em sinais elétricos através do sensor semicondutor. Esses sinais são transportados até um *chip* que transforma as informações eletrônicas recebidas em sinais digitais, ou seja, a matriz de números interpretada pelo computador, recebe valores discretos de acordo com a intensidade da luminosidade.

A Figura 1 ilustra uma imagem em escala de cinza, com seus valores variando de 0 a 255. Um pequeno pedaço dessa imagem é ampliado, mostrando a relação da variação da coloração com a matriz de números.

Figura 1 – Imagem digital vista pelo computador.

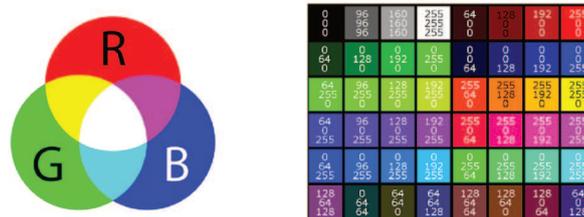


Fonte: (IPMA, 2020)

Uma imagem pode ser considerada como a transformação de uma cena tridimensional e um plano bidimensional. Sendo assim, a imagem pode ser interpretada como uma matriz de brilho bidimensional (MA et al., 2012). Cada elemento desta matriz é denominado de *pixel*. A quantidade de *pixels* de cada imagem depende da resolução da câmera. Imagens que capturam mais detalhes, ou seja, maior resolução, possuem um número maior de *pixels*.

Para fotos RGB, cada *pixel* possui um vetor com três valores discretos, correspondentes a cor vermelha (*Red*), verde (*Green*) e azul (*Blue*). Um imenso conjunto de cores pode ser gerado pela combinação dessas 3 posições, como demonstrado na Figura 2.

Figura 2 – Padrão RGB.

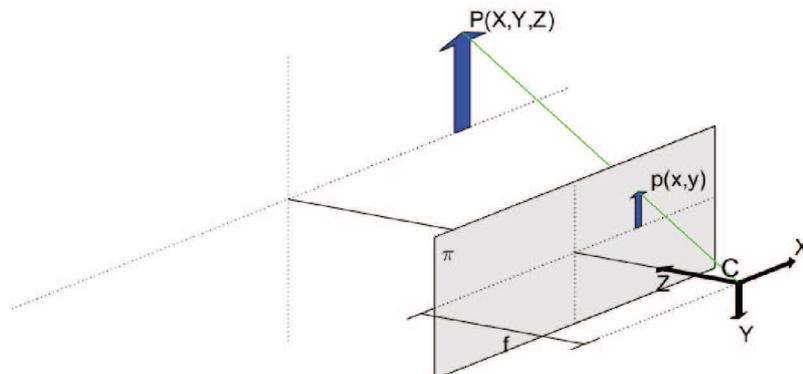


Fonte: (CLUB, 2020)

2.1.1 Formação das imagens

A câmera *pinhole* é um exemplo simples de como a imagem é gerada. Essa câmera pode ser representada por um recipiente fechado e com um orifício em uma de suas extremidades. Um feixe de luz passa por esse orifício e é projetado na parede da outra extremidade (STROBL; HIRZINGER, 2011). O modelo de câmera *pinhole* não considera a utilização de uma lente na câmera. Já a câmera de projeção central assume que a lente direciona todos os raios de luz em um único ponto, denominado Centro ótico (*C*) (CORKE, 2017). A Figura 3 demonstra o comportamento desse modelo.

Figura 3 – Formação da imagem em uma câmera com projeção central.



Fonte: Elaborado pelo autor.

A lente permite captar objetos com menores luminosidades para a formação da imagem, no entanto, necessita de ajuste de foco para distâncias específicas. Ao realizar a captura, é gerado um plano de imagem (π) em frente a \mathbf{C} . Adotando \mathbf{C} como a origem, esse plano fica a uma distância no eixo de profundidade igual a distância focal da câmera (f). A Equação 2.1 relaciona geometricamente um ponto de coordenadas $\mathbf{p} = [x, y]^T$, utilizando um ponto real $\mathbf{P} = [X, Y, Z]^T$ e \mathbf{C} .

$$\mathbf{p} = f \frac{\mathbf{P}}{Z} \quad (2.1)$$

Essa Equação é uma transformada projetista que possui as seguintes propriedades (CORKE, 2017):

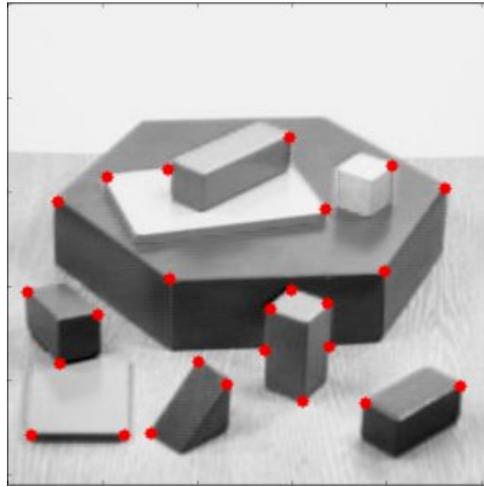
- a) Linhas retas na cena são projetadas como linhas retas no plano da imagem;
- b) Linhas paralelas na cena se interceptam no horizonte no plano da imagem, com exceção de linhas paralelas ao eixo x do plano da imagem;
- c) Geometrias cônicas são projetadas para cônicas no plano da imagem, mas não necessariamente se mantém iguais, por exemplo, um círculo pode ser projetado como uma elipse;
- d) O tamanho e o formato não são preservados;
- e) O mapeamento não é único, um ponto bidimensional não consegue definir sozinho um ponto tridimensional.

2.1.2 Pontos de Interesse - *Features*

Em imagens digitais que possuem um grande número de *pixels*, o rastreamento de cada um deles se torna redundante e implica em um alto esforço computacional (PAREEK; ARORA, 2019). Uma estratégia adotada nessa situação é a extração de pontos de interesse, definido como padrões, que possuem características diferentes de seus vizinhos, como textura, intensidade, cor, etc. Essas *features* podem ser determinadas por pontos, cantos ou *blobs* (segmento de imagens) e possui como uma das principais características a repetibilidade, encontradas em diferentes pontos de vista e condições (BAY et al., 2008).

Em 1988, (HARRIS; STEPHENS, 1988) apresentou um trabalho que se consagrou na literatura. Intitulado como detectores de cantos, tem como base, a localização de pontos de interesse com gradientes relevantes em todas as direções para uma determinada escala de análise (SOLEM, 2012). De forma empírica, (SCHMID; ZISSERMAN, s.d.) demonstrou que o erro de correspondência dessa técnica normalmente é menor que um *pixel*. A Figura 4 ilustra um exemplo do comportamento do método de Harris.

Figura 4 – *Features* encontradas utilizando o método de Harris.



Fonte: (OPENCV, 2020)

(SOLEM, 2012) demonstra como esse método é empregado de forma computacional. Para um determinado ponto no domínio da imagem é obtida uma matriz positiva, semi-definida e simétrica \mathbf{M}_I , descrita pela Equação 2.2.

$$\mathbf{M}_I = \nabla I \nabla I^T = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.2)$$

onde, ∇I é um gradiente de imagem que contém as derivadas na direção x (I_x) e na direção y (I_y). A matriz \mathbf{M}_I possui posto 1 com autovalor $\lambda_1 = |\nabla I^2|$ e $\lambda_2 = 0$.

Para o cálculo da matriz Harris, faz-se uma busca por variações de ponto utilizando um filtro tipicamente gaussiano \mathbf{G}_σ ($\bar{\mathbf{M}}_I$) como mostra a Equação 2.3. Os auto-valores da matriz Harris são capazes de indicar a variação entre o ponto utilizado e seus vizinhos.

$$\bar{\mathbf{M}}_I = \mathbf{G}_\sigma \mathbf{M}_I \quad (2.3)$$

A principal ideia desta metodologia é que, ao utilizar o filtro gaussiano, os auto-valores da matriz de Harris se diferenciam dos obtidos na matriz \mathbf{M}_I , de acordo com as propriedades locais da imagem. Esses auto-valores são capazes de descrever três situações para o ponto utilizado:

- a) Ponto de interesse, caso λ_1 e λ_2 possuam um valor positivo alto;
- b) Ponto em uma região de aresta ou traço da imagem, caso λ_1 seja alto e λ_2 próximo a zero;
- c) Ponto sem representatividade, caso λ_1 e λ_2 sejam baixos.

A Equação 2.4 pode ser utilizada para distinguir a primeira situação sem necessidade de obter os auto-valores da matriz de Haris. Caso o valor obtido seja maior (th) que um limite estabelecido, trata-se de um ponto de interesse.

$$th = \frac{|\bar{\mathbf{M}}_I|}{Tr(\bar{\mathbf{M}}_I)^2} \quad (2.4)$$

2.1.3 Descritores e *Match* de Pontos

Um dos princípios mais importantes da reconstrução 3D a partir de imagens 2D é a identificação de pontos correspondentes em diferentes imagens. Para isso, trabalhos foram desenvolvidos com o intuito de descrever os pontos de interesse (JIANG; MERICKEL, 1989; LOWE, D. G, 1999; GRZYCUK; SCHERER; GABRYEL, 2015), criando os descritores que trabalham em conjunto com os detectores de *features*. A interpretação dos pontos de interesse não é importante, entretanto, seus descritores devem ser robustos o suficiente para encontrar um mesmo ponto em imagens diferentes.

Como as imagens analisadas no processo de reconstrução 3D podem apresentar transformações como rotação, translação e mudança de escalas, deve-se optar por um detector de *features* do tipo *blobs*, que apesar de ser menos precisos em localização se comparados a um detector de cantos é mais robusto a mudanças de formatos e escalas (FÖRSTNER; WROBEL, 2016).

Em 1999 foi desenvolvido uma das metodologias mais abordadas para a obtenção de *features* e seus descritores. (LOWE, D. G, 1999) apresentou um algoritmo invariante a iluminação, escala e rotação, denominado de SIFT (*Scale Invariant Feature Transform*). Esse algoritmo utiliza-se de uma técnica denominada DoG (*Difference of Gaussian*) para determinar as *features* da imagem.

O desfoque gaussiano não respeita os limites naturais dos objetos, suaviza os detalhes e o ruído sem um tratamento específico, reduzindo a precisão e a distinção da localização (BEDRUZ et al., 2018). O algoritmo denominado KAZE, apresenta uma estratégia que permite obter um melhor desempenho nessas circunstâncias. Esse algoritmo detecta e descreve recursos 2D em um espaço de escala não linear por meio de filtragem de difusão não linear. Com isso, o desfoque é realizado localmente, mantendo os limites do objeto e reduzindo o ruído (ALCANTARILLA; BARTOLI; DAVISON, s.d.).

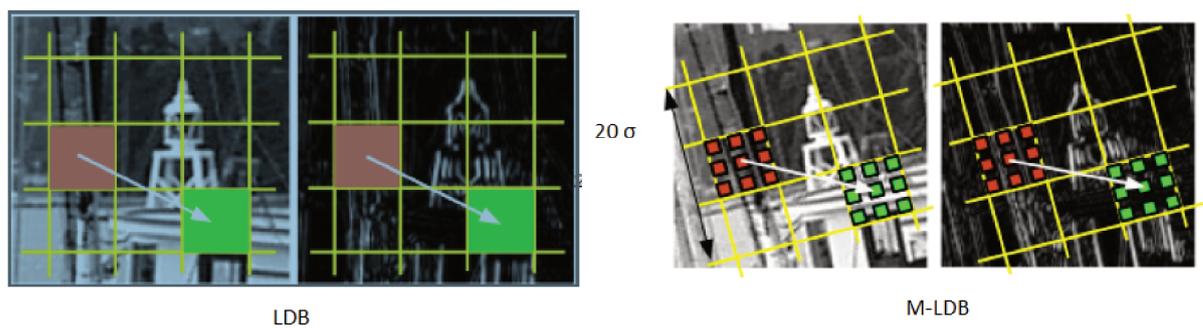
O KAZE utiliza o método *Additive Operator Splitting* (AOS), que apesar de ser estável para qualquer incremento de tempo, é um algoritmo mais lento, que requer o cálculo de um grande sistema de equações lineares para obter uma solução (BEDRUZ et al., 2018). Um melhor desempenho pode ser observado no desenvolvimento do *Accelerated Kaze* (ALCANTARILLA; SOLUTIONS, 2013), conhecido como AKAZE. Nesse algoritmo foi introduzido o método *Fast Explicit Diffusion* (FED) que possui resultados mais precisos,

além de menor custo computacional que o AOS.

O FED proposto utiliza uma estrutura piramidal fina e grossa, permitindo uma construção rápida de espaços em escala não linear, adequados para detecção e descida de recursos robustos. Um espaço em escala não linear é construído discretizando o espaço de escala em séries de oitavos e subníveis. Quando o último subnível de cada oitava é alcançado, a imagem é subamostrada por um fator de 2, utilizando uma máscara de suavização. Esse passo se repete, até que o número máximo de níveis de pirâmide seja alcançado.

Para cada ponto da imagem é calculado uma matriz Hessiana pertencente ao seu espaço em escala não linear. Caso a determinante dessa matriz seja maior que um valor estipulado e corresponda a máxima entre seus vizinhos e subníveis, esse ponto é identificado como um ponto de interesse. O descritor binário *Modified-Local Difference Binar* (M-LDB) é utilizado para obter as características do ponto, utilizando informações do gradiente e da intensidade do espaço de escala não linear. O LDB é um descritor que divide a imagem em *grids* de tamanho de dimensão $\mathfrak{R}^{N \times N}$ e, para cada célula desse *grid* é obtido a média de intensidade, o gradiente de primeira ordem nas coordenadas x e y. A invariância da rotação é obtida encontrando a orientação do ponto de interesse e rotacionando o *grid* do LDB. Já o M-LDB, subamostra o *grid* em partes, de acordo com a escala da *feature*, ao invés de calcular a média de intensidade de cada célula. Essa amostragem, que varia com a escala, faz o descritor resistente às mudanças de escala. A Figura 5 demonstra uma comparação entre os descritores.

Figura 5 – Desempenho dos descritores LDB e M-LDB.



Fonte: (ALCANTARILLA; SOLUTIONS, 2013)

Para a correspondência de um mesmo ponto em diferentes imagens, é utilizada a métrica de distância de Hamming entre os vetores dos descritores. Dado um ponto de uma imagem, é feita uma varredura dos pontos de interesse na imagem seguinte, a fim de identificar o que mais se parece com o ponto selecionado. Se a distância encontrada entre os dois pontos for menor que um valor estabelecido, esses pontos são considerados

um *match*. É valido ressaltar, que esse descritor é capaz de informar a variação de escala e rotação de um ponto para o outro.

2.2 Modelagem da Câmera

Essa seção apresenta como são realizados os cálculos dos parâmetros da câmera com projeção central e descreve sua modelagem matemática, ou seja, como obter seus parâmetros intrínsecos (Seção 2.2.1) e extrínsecos (Seção 2.2.2). Por fim a seção 2.2.3 demonstra como estimar os parâmetros intrínsecos, utilizando um processo de calibração através de imagens.

2.2.1 Parâmetros Intrínsecos

Os parâmetros intrínsecos de uma câmera correspondem às características da câmera, como por exemplo, a resolução de suas imagens, o quanto a lente distorce a imagem, sua distância focal. Ou seja, esse parâmetro demonstra como a câmera enxerga o mundo. A demonstração da modelagem deste comportamento é baseada em (HARTLEY; ZISSERMAN, 2003).

A Equação 2.1 demonstra como um ponto no espaço tridimensional é transformado para um ponto bidimensional no plano da imagem. Para uma modelagem mais precisa, essa equação pode ser descrita de forma matricial. Segundo (HARTLEY; ZISSERMAN, 2003), a relação de uma coordenada homogênea do plano de imagens $(u, v, w)^T$, sendo w o fator de escala, e um ponto do mundo real $(X, Y, Z)^T$, pode ser expressa pela Equação 2.5, onde f corresponde a distância focal da imagem.

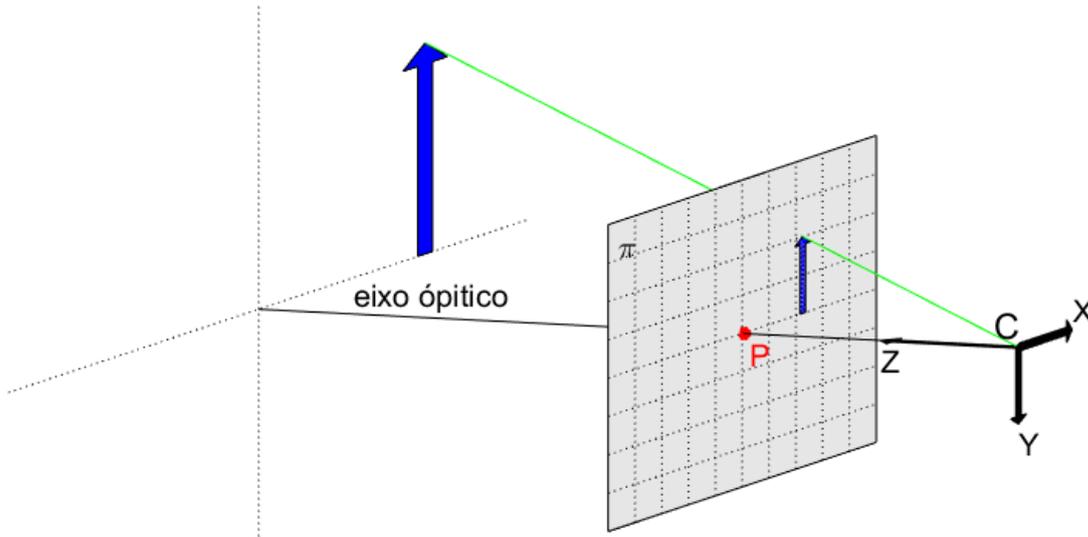
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.5)$$

Essa equação é capaz de expressar uma transformação para câmeras que possuem seus semicondutores quadrados, com isso a distância focal no eixo x e no eixo y possuem o mesmo valor. Entretanto, muitas vezes esses sensores possuem uma configuração retangular, apresentando uma variação na percepção de um eixo em relação ao outro. A fim de representar esse comportamento, a distância focal é subdividida em duas: distância focal no eixo X (f_x) e distância focal no eixo Y (f_y). Com isso a Equação 2.5 tem uma pequena variação, resultando na Equação 2.6.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.6)$$

Ao realizar um ajuste na configuração da lente ocorre uma variação nos valores da distância focal. Outro parâmetro que sofre alteração nesse procedimento é denominado ponto principal. Esse ponto sofre uma variação de acordo com a distorção gerada pela lente. A Figura 6 ilustra o ponto principal (P) para um modelo de câmera central.

Figura 6 – Demonstração do eixo óptico e do ponto principal no plano da imagem.



Fonte: Elaborado pelo autor.

O eixo perpendicular ao plano de imagem (π) e que contém o centro óptico é denominado eixo óptico. Já o ponto principal é representado pelo ponto de interseção do eixo óptico com o plano de imagem. Quando o eixo óptico é colinear com o eixo Z , como no caso da Figura 6, o centro óptico encontra-se no centro do plano de imagem.

A Equação 2.6 considera o ponto principal na origem $(0,0)$. No entanto, a imagem digital bidimensional é composta por uma matriz discreta positiva, onde o *pixel* posicionado no canto esquerdo inferior corresponde a origem $(0,0)$. Sendo assim, o ponto principal é deslocado para o centro da imagem e acrescido do deslocamento gerado pela distorção da lente. A Equação 2.7 demonstra o modelamento da transformação discreta do espaço 3D para o 2D.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & P_x \\ 0 & f_y & P_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.7)$$

A matriz obtida corresponde a matriz intrínseca (\mathbf{K}) da câmera, onde P_x e P_y correspondem às coordenadas do ponto principal. A inversa dessa matriz pode ser utilizada

para reprojeter um ponto da imagem para o espaço tridimensional, desde que o *frame* da câmera seja de referência.

2.2.2 Parâmetros Extrínsecos

O processo de captura de imagens sucessivas em cenários estáticos consiste em deslocar a câmera para a obtenção de diferentes perspectivas da cena. Essa variação permite o cálculo da odometria das câmeras. Dado um *frame* de referência, as câmeras devem possuir a relação de sua orientação em relação a ele. Essa relação é dada por uma matriz de rotação (\mathbf{R}) e um vetor de translação (\mathbf{t}). Para câmeras que possuem *frame* igual ao *frame* de referência, a rotação e a translação são nulas, como visto na Equação 2.8a.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & P_x \\ 0 & f_y & P_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & | & 0 \\ 0 & 1 & 0 & | & 0 \\ 0 & 0 & 1 & | & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.8a)$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underbrace{\mathbf{K} [\mathbf{R} | \mathbf{t}]}_{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.8b)$$

A matriz que representa as configurações da câmera, denominada matriz de pose (\mathbf{P}), é demonstrada na Equação 2.8b. Com essa matriz, é possível realizar a projeção do ponto 3D para qualquer plano de imagem. Vale ressaltar que o cálculo da reprojeção não é trivial, visto que a matriz \mathbf{P} é de dimensão $\mathbb{R}^{3 \times 4}$. Para isso, normalmente são utilizadas técnicas de triangulação.

Já a matriz extrínseca (ζ) pode ser obtida pela Equação 2.9. Essa matriz representa a orientação da câmera dado um *frame* de referência e pode ser utilizada para projetar informações de um *frame* da câmera para outro *frame* desejado utilizando sua inversa.

$$\zeta = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (2.9)$$

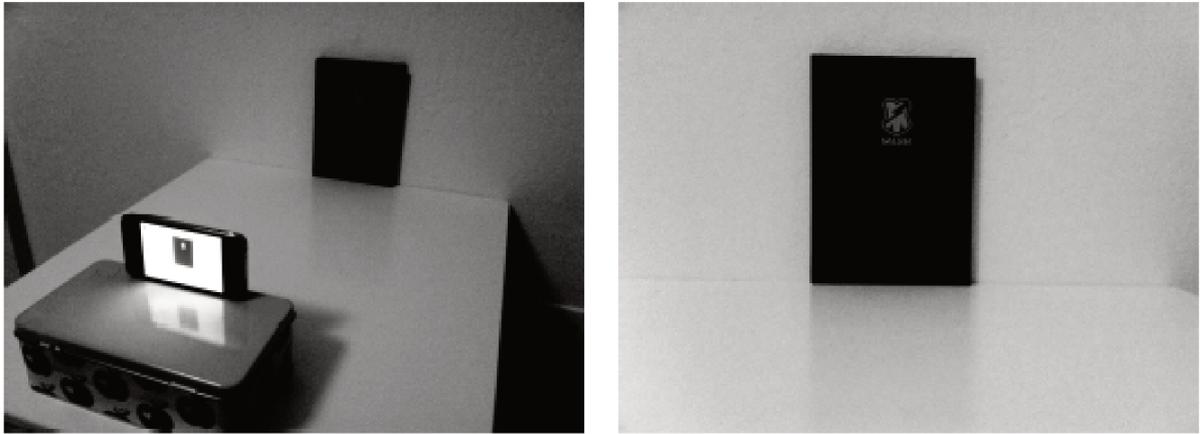
Por fim, o centro ótico (\mathbf{C}) da câmera é dado pela Equação 2.10. Com ele é possível obter o ponto exato do sensor da câmera no momento da captura.

$$\mathbf{C} = -\mathbf{R}^T \mathbf{t} \quad (2.10)$$

2.2.3 Calibração da Câmera

Esta subseção apresenta como determinar os valores dos parâmetros da matriz intrínseca. Durante um mapeamento de uma cena para a reconstrução 3D, normalmente é feito um ajuste na câmera, que é utilizado em todo o processo de captura das imagens. Nesse caso, todas as imagens irão apresentar o mesmo valor para suas matrizes. (SOLEM, 2012) apresenta um algoritmo de fácil aferição dos parâmetros intrínsecos. Um objeto retangular com dimensões conhecidas é posicionado a frente da câmera a ser calibrada em uma superfície plana. O objeto deve ficar no centro do campo de visão da câmera, como demonstra a Figura 7.

Figura 7 – Calibração de uma câmera com auxílio de um objeto de dimensões conhecidas.



Fonte: (SOLEM, 2012)

As dimensões em *pixels* do objeto podem ser obtidas utilizando os pontos das extremidades do objeto na imagem, sendo dx o comprimento horizontal e dy o comprimento vertical. Com a distância dZ entre a câmera e o objeto, é possível obter as distâncias focais da câmera utilizando as Equações 2.11a e 2.11b.

$$f_x = \frac{dx}{dX}dZ \quad (2.11a)$$

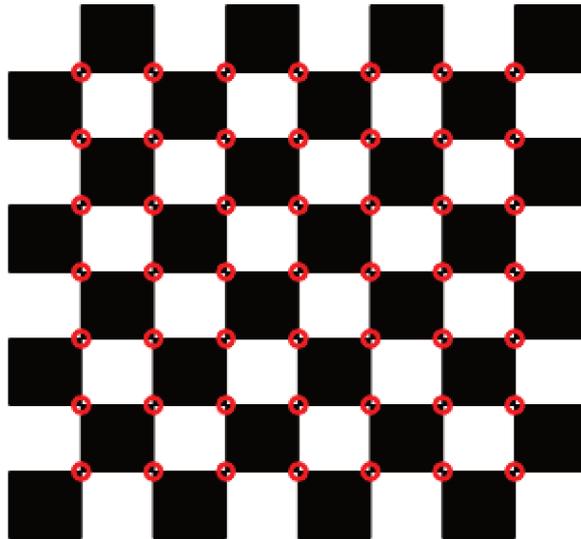
$$f_y = \frac{dy}{dY}dZ \quad (2.11b)$$

onde, dX e dY são as dimensões do objeto. Nessa calibração é assumido que o ponto principal encontra-se no centro da imagem.

Esse processo de calibração é realizado manualmente, podendo apresentar erros de medições. Muitos trabalhos foram desenvolvidos com o intuito de automatizar esse processo (BEVILACQUA; GHERARDI; CAROZZA, 2008; XINGFANG; YUMEI; FENG, 2010; ESCALERA; ARMINGOL, 2010; DEGLINT et al., 2016). Inicialmente, é necessário

um padrão conhecido e facilmente identificado na imagem. Um dos padrões mais utilizados é o tabuleiro de xadrez. Esse tabuleiro é um objeto plano e as extremidades de seus quadrados podem ser identificados facilmente. Como os quadrados do tabuleiro possuem tamanhos fixos, seu valor pode ser definido *a priori*, e uma comparação semelhante a demonstrada na Equação 2.11 pode ser realizada em diversas partes da imagem. A Figura 8 demonstra como esses padrões são obtidos em um tabuleiro.

Figura 8 – Detecção dos pontos de interesse de um tabuleiro de xadrez para a calibração da câmera.



Fonte: Elaborado pelo autor.

Um método de calibração é apresentado por (MA et al., 2012). Seu desenvolvimento matemático envolve o conceito de homografia (abordado na Seção 2.3.1) plana, a qual projeta um plano sobre o outro. Nessa estratégia, o *frame* de referência é alinhado ao tabuleiro para que seus pontos possuam coordenadas homogêneas descritas por $\mathbf{X} = [X, Y, 0, 1]^T$, ou seja, o eixo de profundidade (Z) é a normal desse *frame*. Dado um ponto $\mathbf{x} = [u, v, w]^T$, onde w corresponde ao fator de escala, pode-se relacioná-lo com um ponto do espaço tridimensional pela Equação 2.12.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \underbrace{\mathbf{K} \langle \mathbf{r}_1, \mathbf{r}_2, \mathbf{t} \rangle}_{\mathbf{H}} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2.12)$$

onde \mathbf{r}_1 e $\mathbf{r}_2 \in \mathfrak{R}^3$ são a primeira e segunda coluna da matriz de rotação e \mathbf{t} o vetor de translação. $\mathbf{H} \in \mathfrak{R}^{3 \times 3}$ é a matriz de homografia entre o tabuleiro de xadrez e a imagem. Uma matriz simétrica inclinada ($\hat{\mathbf{x}}$) (*skew-symmetric matrix*) é multiplicada pela equação com o intuito de eliminar a primeira parte de igualdade da Equação 2.12, resultando na

Equação 2.13.

$$0 = \mathbf{H} \begin{bmatrix} X & Y & 1 \end{bmatrix}^T \quad (2.13)$$

As duas primeiras colunas da matriz \mathbf{H} podem ser interpretadas como $[\mathbf{h}_1, \mathbf{h}_2] \approx \mathbf{K}[\mathbf{r}_1, \mathbf{r}_2]$, ou ainda, $\mathbf{K}^{-1}[\mathbf{h}_1, \mathbf{h}_2] \approx [\mathbf{r}_1, \mathbf{r}_2]$. Como r_1 e r_2 são ortogonais, a Equação 2.13 pode ser reescrita pelas equações 2.14a e 2.14b.

$$\mathbf{h}_1 \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 = 0 \quad (2.14a)$$

$$\mathbf{h}_1 \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_1 = \mathbf{h}_2 \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 \quad (2.14b)$$

Assumindo $\mathbf{S} = \mathbf{K}^{-T} \mathbf{K}^{-1}$, sendo $\mathbf{S} \in \mathfrak{R}^{3 \times 3}$, pode-se obter os parâmetros intrínsecos da câmera utilizando o método de *Singular Value Decomposition* (SVD). Como a matriz \mathbf{K} proposta possui 4 parâmetros a serem identificados, são necessárias 5 equações para o cálculo dessa matriz. Cada imagem obtém duas equações lineares, além disso a condição $\mathbf{e}_1^T \mathbf{S} \mathbf{e}_2 = 0$ deve ser satisfeita, sendo \mathbf{e}_1 e \mathbf{e}_2 (Seção 2.3.3) os pontos epipolares para cada imagem. Com isso pode-se usar apenas 2 imagens para a calibração, entretanto, quanto mais fotos utilizadas, maior será a eficiência do método.

2.3 Structure From Motion (SFM)

O SFM busca através de uma sequência de imagens obter informações da câmera utilizada e estimar uma nuvem de pontos esparsa (YI et al., 2014). (BROWN; LOWE, 2005) apresenta um dos primeiros algoritmos SFM capaz de operar com um conjunto de imagens não ordenadas de forma não supervisionada. Para cada imagem é associada uma matriz de câmera, parametrizada por rotação, translação e distância focal. A correspondência (*match*) entre as imagens são obtidas utilizando os seus pontos de interesse (*features*), além da utilização do algoritmo *Random Sample Consensus* (RANSAC), responsável por filtrar os pontos consistentes com a matriz fundamental. As imagens são separadas em pares e suas correspondências são utilizadas para a criação da nuvem esparsa. Por fim, para o ajuste da nuvem obtida, bem como das características da câmera, é utilizado a técnica *Bundle adjustment* (BA).

Em um aperfeiçoamento dessa técnica, um sistema para reconstrução em larga escala foi desenvolvido por (SNAVELY; SEITZ; SZELISKI, 2006), utilizando imagens de diferentes câmeras, níveis de zoom, resoluções, iluminação, clima e quantidades de oclusão. É atribuído uma condição inicial estimada para novas câmeras, diminuindo a incidência de mínimos locais, além de apresentar uma heurística capaz de identificar o melhor par de imagens inicial.

Ainda para o cálculo da odometria, (SHAH; DESHPANDE; NARAYANAN, 2014) propõe a utilização de um filtro de alta escala, dos descritores identificados na imagem, diminuindo assim, o gasto computacional para a estimativa da odometria das câmeras. Já (WU, 2013) apresenta outra estratégia adotada, a obtenção de um modelo mais grosseiro que possui apenas um subconjunto das imagens, é denominado esqueleto de imagens. Em uma próxima etapa, as demais imagens podem ser utilizadas de forma independentes, possibilitando a paralelização deste processo.

Para um melhor desempenho do BA, (WU, 2013) sugere realizar um ajuste global de um esqueleto de imagens, os seja, um subconjunto com as informações necessária para o BA. Para as demais imagens deve ser realizado um ajuste local. Partindo do mesmo princípio, (SCHONBERGER; FRAHM, 2016) defende que essa metodologia é viável quando trabalha com uma larga escala de imagens.

2.3.1 Homografia

A Homografia, também conhecida como transformação projetiva planar, mapeia o ponto de um plano para outro. Essa transformação bidimensional é realizada por uma matriz, denominada de matriz homogênea (\mathbf{H}), pertencente ao espaço $\mathfrak{R}^{3 \times 3}$ e que possui oito graus de liberdade. Dado um ponto em um plano π_1 de coordenada $[u_1^i, v_1^i, w_1^i]^T$ e o seu correspondente do plano π_2 de coordenada $[u_2^i, v_2^i, w_2^i]^T$, a relação entre eles pode ser dada pela Equação 2.15, onde i corresponde ao ponto da imagem utilizado.

$$\begin{bmatrix} u_2^i \\ v_2^i \\ w_2^i \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} u_1^i \\ v_1^i \\ w_1^i \end{bmatrix} \quad (2.15)$$

Trabalhando com coordenadas homogêneas, essa matriz tem como característica a obtenção de linhas retas das imagens, entretanto, não preserva a orientação, ângulos, comprimento e paralelismos (DUBROFSKY, 2009). A Figura 9 demonstra o comportamento da transformação homogênea. Os pontos em vermelho representam os pontos da imagem, enquanto os pontos em verde correspondem aos pontos projetados. Quando o plano calculado não corresponde a um local da imagem, ocorrem erros de projeção, demonstrado pelos pontos azuis.

Figura 9 – Mapeamento de pontos utilizando uma transformada homográfica.



Fonte: (SOLEM, 2012)

A homografia pode ser utilizada para a correção de perspectiva de fotos, adição de efeitos visuais (como a ordem de chegada de atletas da natação), construção de panoramas e a estimação da posição da câmera. A Figura 10, demonstra o exemplo da construção de uma vista panorâmica, utilizando as correspondências encontradas na Figura 9.

Figura 10 – Construção panorâmica a partir de duas imagens.



Fonte: Elaborado pelo autor.

Como a matriz possui oito graus de liberdade, são necessárias oito equações para obtenção de seus parâmetros, dessa forma, a matriz homogênea pode ser estimada utilizando

quatro pontos de uma imagem e seus correspondentes em outra imagem (HARTLEY; ZISSERMAN, 2003). É válido ressaltar que cada correspondência possui dois graus de liberdade. A matriz homogênea, utilizando a resolução do sistema, pode ser demonstrada pela Equação 2.16, considerando h_{rc} diferente de 0. O SVD pode ser utilizado para a sua resolução.

$$\begin{bmatrix} 0 & 0 & 0 & -w_2^1 u_1^1 & -w_2^1 v_1^1 & -w_2^1 w_1^1 & v_2^1 u_1^1 & v_2^1 v_1^1 & v_2^1 w_1^1 \\ w_2^1 u_1^1 & w_2^1 v_1^1 & w_2^1 w_1^1 & 0 & 0 & 0 & -u_2^1 u_1^1 & -u_2^1 v_1^1 & -u_2^1 w_1^1 \\ 0 & 0 & 0 & -w_2^2 u_1^2 & -w_2^2 v_1^2 & -w_2^2 w_1^2 & v_2^2 u_1^2 & v_2^2 v_1^2 & v_2^2 w_1^2 \\ w_2^2 u_1^2 & w_2^2 v_1^2 & w_2^2 w_1^2 & 0 & 0 & 0 & -u_2^2 u_1^2 & -u_2^2 v_1^2 & -u_2^2 w_1^2 \\ \vdots & \vdots \\ 0 & 0 & 0 & -w_2^n u_1^n & -w_2^n v_1^n & -w_2^n w_1^n & v_2^n u_1^n & v_2^n v_1^n & v_2^n w_1^n \\ w_2^n u_1^n & w_2^n v_1^n & w_2^n w_1^n & 0 & 0 & 0 & -u_2^n u_1^n & -u_2^n v_1^n & -u_2^n w_1^n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0 \quad (2.16)$$

Para um conjunto de pontos de uma imagem, pertencentes a planos diferentes, a solução encontrada varia de acordo com o subconjunto utilizado. Para a obtenção de uma matriz homogênea, que melhor representa um plano específico, deve-se utilizar-se de técnicas para selecionar o maior número de *inliers* possível. Um algoritmo bem utilizado é o RANSAC que será descrito na Seção (2.3.2). Para que os eixos possuam a mesma representatividade, ou seja, o mesmo peso, é aplicada uma técnica de normalização, aumentando a precisão do cálculo da matriz. Essa transformação realiza uma translação nos pontos, de forma que a origem seja sua centroide e a distância média de um ponto para a origem seja $\sqrt{2}$. A transformação deve ser realizada para cada conjunto separadamente.

A matriz de homografia também pode ser obtida utilizando os parâmetros intrínsecos e extrínsecos das câmeras (HARTLEY; ZISSERMAN, 2003), como demonstrado na Equação 2.17. Essa equação é válida quando o plano da primeira imagem seja na origem. Para um conjunto de câmeras deslocadas da origem, basta referenciá-las no plano da primeira imagem, utilizando a inversa de sua matriz extrínseca (ζ) apresentada na Equação 2.9.

$$\mathbf{H}_{1,2} = \mathbf{K}_2 (\mathbf{R}_2 - \mathbf{t}_2 \frac{\mathbf{N}^T}{d}) \mathbf{K}_1 \quad (2.17)$$

Dado uma coordenada tridimensional \mathbf{X} , d corresponde a constante do plano com \mathbf{N} e que passa pela coordenada \mathbf{X} , representada pela Equação 2.18.

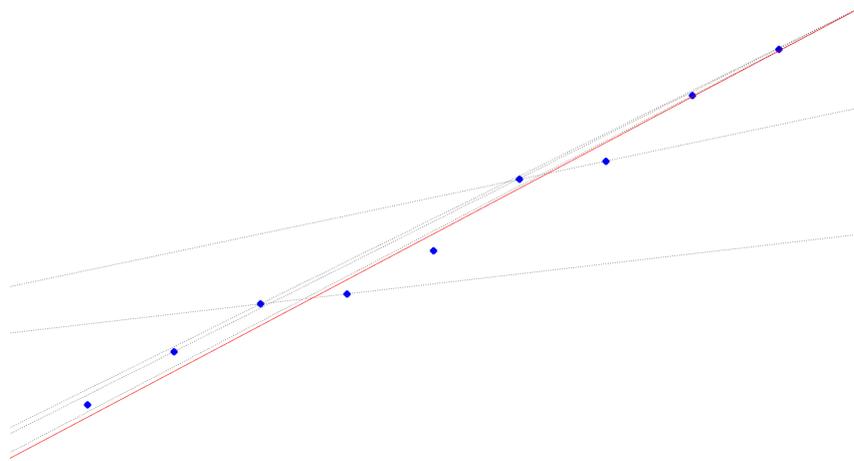
$$d = -\mathbf{N}^T \mathbf{X} \quad (2.18)$$

2.3.2 *Random Sample Consensus (RANSAC)*

RANSAC é um método iterativo para aproximar um modelo matemático a partir de dados experimentais (SOLEM, 2012). Esse método é capaz de suavizar dados com uma quantidade significativa de *outliers*, e é ideal para as aplicações que os dados são gerados a partir de detectores de *features*. Esses detectores apresentam erros de classificação, indicando pontos de interesse incorretos e erros de medições, tal como sua orientação (HARTLEY; ZISSERMAN, 2003). O RANSAC se destaca sobre algumas técnicas clássicas como os mínimos quadrados, por conseguir extrair o efeito dos *outliers* para sua solução (FISCHLER; BOLLES, 1981).

O algoritmo do RANSAC seleciona um subconjunto de um espaço de amostras, com valor mínimo de pontos para obter uma solução. A solução desse subconjunto é comparada com as soluções já encontradas e a que possui o maior número de *inliers* é selecionado como a resposta para o sistema. O sorteio desses subconjuntos é realizado de forma aleatória por um ciclo de n_{min} repetições. A Figura 11 demonstra possíveis configurações de uma reta para representar um conjunto bidimensional, sendo a melhor solução representada em vermelho.

Figura 11 – RANSAC aplicado a um conjunto de dados bidimensional.



Fonte: Elaborado pelo autor.

Trabalhando de forma probabilística, o RANSAC estima o número de interações (n_{min}) mínima, capaz de gerar uma boa solução. Sendo assim, é definido um valor probabilístico (p), normalmente igual a 99%, e uma probabilidade (α) de um dado selecionado ser *outlier*, sendo α atualizado de acordo com as interações. Para um conjunto de “s” elementos, o número de interações necessárias é dado pela Equação 2.19.

$$n_{min} = \frac{\log(1 - p)}{\log(1 - (1 - \alpha)^s)} \quad (2.19)$$

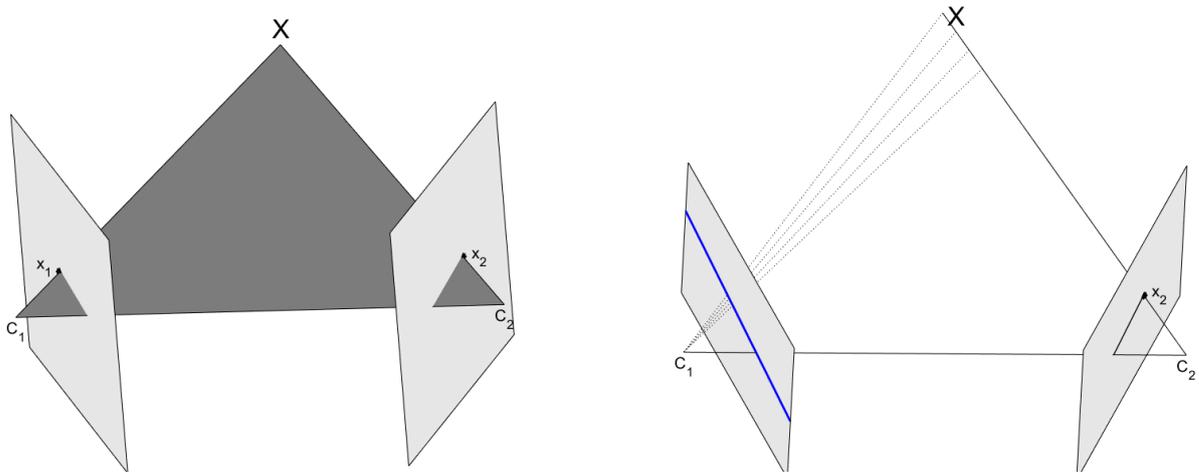
2.3.3 Geometria Epipolar

Geometria de múltiplas vistas (do inglês, *Multiple View Geometry*) é o campo que estuda a relação entre câmeras e os pontos de interesse para imagens retiradas de diferentes pontos e que possuem correspondências entre si, destacando-se a relação entre duas imagens (SOLEM, 2012).

A relação entre duas imagens, baseada na posição dos pontos 3D, possuem algumas restrições geométricas, dentre elas destacam-se os resultados da orientação relativa das câmeras e suas propriedades. Essa relação geométrica, descrita como geometria epipolar, independente da estrutura da cena e considera apenas os parâmetros intrínsecos da câmera (HARTLEY; ZISSERMAN, 2003).

A geometria epipolar é essencialmente a geometria de interseção de planos de imagens que possuem como *baseline* a reta gerada entre os centros das câmeras. Dado um ponto \mathbf{X} no espaço 3D, nomeado como \mathbf{x}_1 na primeira vista e \mathbf{x}_2 na segunda vista, é possível criar um plano entre o centro das câmeras e esses pontos. Esse plano é chamado de plano epipolar e pode ser observado na cor cinza escuro na Figura 12.

Figura 12 – Plano e linha epipolar entre duas imagens.



Fonte: Elaborado pelo autor.

Essa geometria é utilizada para a procura de pontos correspondentes entre as câmeras. Dado um ponto e as propriedades das câmeras, é possível definir o plano epipolar, onde o ponto \mathbf{x}_1 pertence ao plano epipolar e ao plano da imagem. A intersecção entre esses planos é denominada linha epipolar (linha azul), e contém o ponto \mathbf{x}_1 . Essa linha facilita a procura de pontos correspondentes entre as câmeras, pois não é mais necessário realizar a busca em toda a imagem. Entretanto, essa linha é capaz de indicar apenas possíveis pontos correspondentes, visto que um ponto de uma imagem pode estar oculto em outra imagem. Já os pontos que estão contidos no plano da imagem e no eixo gerado

entre as duas câmeras são denominados de epipolo.

2.3.3.1 *Matriz Fundamental*

A geometria epipolar pode ser representada algebricamente através da matriz fundamental F , que é uma matriz pertencente ao espaço $\mathfrak{R}^{3 \times 3}$ e de posto 2, ou seja, ela tem duas linhas que são linearmente independentes (LUONG; FAUGERAS, 1996). Essa matriz mapeia os pontos de uma imagem para suas correspondentes linhas epipolares na outra imagem (SZELISKI, 2010). A condição que os pontos devem satisfazer para serem considerados correspondentes é dada pela Equação 2.20.

$$\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1 = 0 \quad (2.20)$$

Cada par de pontos correspondentes, pode-se obter uma equação. Com pelo menos sete correspondências, é possível realizar o cálculo da matriz fundamental. Para um conjunto de pontos da primeira imagem $[u_1^i, v_1^i, 1]^T$ e seus correspondentes numa segunda imagem $[u_2^i, v_2^i, 1]^T$, é possível desenvolver o sistema demonstrado na Equação 2.21, sendo f_{rc} diferente de 0.

$$\begin{bmatrix} u_2^1 u_1^1 & u_2^1 v_1^1 & u_2^1 & v_2^1 u_1^1 & v_2^1 v_1^1 & v_2^1 & u_1^1 & v_1^1 & 1 \\ u_2^2 u_1^2 & u_2^2 v_1^2 & u_2^2 & v_2^2 u_1^2 & v_2^2 v_1^2 & v_2^2 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots \\ u_2^n u_1^n & u_2^n v_1^n & u_2^n & v_2^n u_1^n & v_2^n v_1^n & v_2^n & u_1^n & v_1^n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0 \quad (2.21)$$

Para solucionar esse problema, pode-se utilizar a SVD, onde a solução é o autovetor correspondente ao menor autovalor, localizado na última coluna da matriz $\mathbf{V}([U, \mathbf{S}, \mathbf{V}] = \text{SVD}(\mathbf{A}))$. Entretanto, como a matriz fundamental possui posto 2, deve-se calcular o SVD da matriz gerada e zerar o menor autovalor, respeitando essa condição. A fim de maximizar o resultado, é aconselhável realizar a normalização dos pontos, além de utilizar o algoritmo RANSAC.

Quando a matriz Pose (\mathbf{P}) das câmeras é conhecida, essa transformação pode ser calculada sem a utilização das imagens. O equacionamento da matriz fundamental a partir

das matrizes de Pose (\mathbf{P}) das câmeras pode ser observado na Equação 2.22.

$$\mathbf{F}(i, j) = (-1)^{i+j} \det \begin{vmatrix} \mathbf{P}_1^{k!=j} \\ \mathbf{P}_2^{k!=i} \end{vmatrix} \quad (2.22)$$

onde, $\mathbf{P}_n^{k!=r}$ corresponde às linhas da matriz de projeção (\mathbf{P}) de uma câmera n , com exceção da linha r .

Como dito anteriormente, a Matriz Fundamental é capaz de projetar um ponto de uma imagem, na sua linha epipolar da outra imagem. O ponto de correspondência se desloca na linha epipolar à medida que a profundidade do ponto tridimensional varia. Essa linha epipolar pode ser descrita pelas Equações 2.23a e 2.23b.

$$l_1 = x_2^T F \quad (2.23a)$$

$$l_2 = F x_1 \quad (2.23b)$$

A Figura 13 ilustra a projeção de um ponto, para sua respectiva linha epipolar.

Figura 13 – Comportamento das linhas epipolares.



Fonte: Elaborado pelo autor.

A linha epipolar é bem utilizada na filtragem *matches* de pontos de interesse. Dado dois pontos correspondentes \mathbf{x}_1 e \mathbf{x}_2 , a linha epipolar \mathbf{l}_1 pode ser calculada através do ponto \mathbf{x}_2 e da matriz \mathbf{F} . Logo em seguida, calcula-se a distância do ponto \mathbf{x}_1 para a reta \mathbf{l}_1 , caso seja maior que um *threshold*, o ponto é considerado *outlier*. Uma métrica utilizada para calcular essa distância é a *Sampson distance*, expressa pela Equação 2.24.

$$d_{\perp}^2 = \|\boldsymbol{\delta}\|^2 = \frac{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}}{\mathbf{J} \mathbf{J}^T} \quad (2.24a)$$

$$d_{\perp}^2 = \frac{(\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1)^2}{(\mathbf{F} \mathbf{x}_1)_1^2 + (\mathbf{F} \mathbf{x}_1)_2^2 + (\mathbf{F} \mathbf{x}_2)_1^2 + (\mathbf{F} \mathbf{x}_2)_2^2} \quad (2.24b)$$

onde, \mathbf{J} corresponde a matriz de derivada parcial; $\boldsymbol{\epsilon}$ corresponde ao erro encontrado na Equação 2.20, ou seja, $\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1$; e, $(\mathbf{F} \mathbf{x}_2)_j$ representa a posição j da linha epipolar gerada.

2.3.3.2 Matriz Essencial

Um aperfeiçoamento da matriz Fundamental é a matriz Essencial (\mathbf{E}). A matriz essencial é obtida para casos de imagens com as coordenadas normalizadas. Essa matriz só pode ser usada para câmeras calibradas, pois os parâmetros internos da câmera devem ser conhecidos para alcançar a normalização. Se, no entanto, as câmeras forem calibradas, a matriz essencial poderá ser útil para determinar a posição e orientação relativa entre as câmeras e a posição 3D dos pontos de imagem correspondentes (FÖRSTNER; WROBEL, 2016). Sendo assim, essa matriz pode ser representada pela Equação 2.25.

$$\mathbf{E} = \mathbf{K}_2^{-T} \mathbf{F} \mathbf{K}_1^{-1} \quad (2.25)$$

A matriz essencial possui cinco graus de liberdade, definidos pela matriz de rotação e o vetor de translação da câmera, número esse menor que o da matriz fundamental. A matriz essencial pode ser obtida pela relação cruzada dessas orientações como demonstrado na Equação 2.26. Apesar de tanto a rotação quanto a translação possuírem três graus de liberdade, na relação cruzada ocorre ambiguidades que reduzem um grau de liberdade (YANG; LI; JIA, s.d.).

$$\mathbf{E} = [\mathbf{t}]_x \mathbf{R} \quad (2.26a)$$

$$\mathbf{E} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \mathbf{R} \quad (2.26b)$$

A matriz essencial pertence ao espaço $\Re^{3 \times 3}$ e possui dois autovalores iguais, além de um autovalor nulo. Dessa forma, ela pode ser representada pela decomposição em valores singulares demonstrada na Equação 2.27 (MA et al., 2012).

$$\mathbf{E} = \mathbf{U} \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^T \quad (2.27)$$

2.3.4 Cálculo da Matriz de Pose da Câmera

Para o cálculo da matriz de pose da câmera através de imagens 2D, faz-se necessário pelo menos duas figuras que possuam sobreposição. Nesse cálculo assume-se o *frame* da primeira câmera como referência. Utilizando pontos correspondentes nas imagens é possível, com o auxílio da matriz fundamental, estimar uma transformada projetista, sem o conhecimento das características da câmera. Essa transformação consiste em relacionar os epipolos das imagens e a matriz fundamental, entretanto, não respeita ângulos e distâncias da cena (SOLEM, 2012).

De posse dos parâmetros intrínsecos da câmera, é possível utilizar a matriz essencial para a realização de uma transformada mais robusta, capaz de manter as características da cena. Isso não se aplica a escala do cenário, uma vez que não é possível identificar o tamanho do objeto (HARTLEY; ZISSERMAN, 2003).

Utilizando parâmetros obtidos na decomposição em valores singulares da matriz essencial (Equação 2.27), é possível obter a matriz de rotação e o vetor de translação da câmera utilizando a Equação 2.28.

$$\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^T \quad \text{ou} \quad \mathbf{R} = \mathbf{U}\mathbf{W}^T\mathbf{V}^T \quad (2.28a)$$

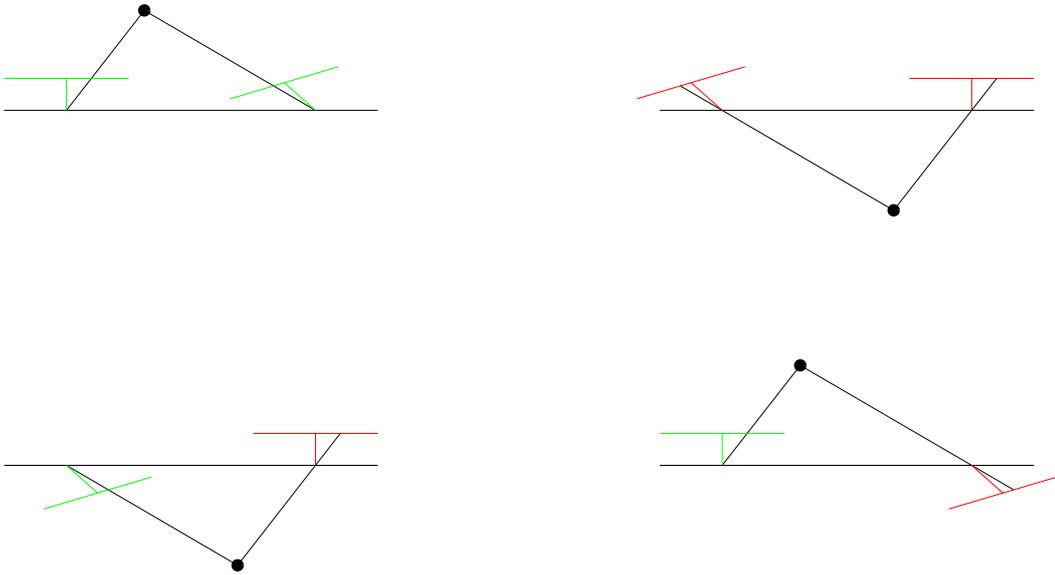
$$\mathbf{t} = \mathbf{U}_3 \quad \text{ou} \quad \mathbf{t} = -\mathbf{U}_3 \quad (2.28b)$$

onde \mathbf{W} corresponde a uma matriz ortogonal dada pela Equação 2.29.

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.29)$$

Como pode ser observado na Equação 2.28, é possível encontrar duas matrizes de rotação (Equação 2.28a) e dois vetores de translação (Equação 2.28b), com isso, quatro soluções podem ser geradas. A fim de identificar a solução que corresponde ao sistema, deve-se utilizar um ponto tridimensional e seus pontos correspondentes nas imagens, observando em qual par de soluções o ponto se encontra a frente das duas câmeras. A Figura 14 demonstra como seria essa avaliação.

Figura 14 – Posição do ponto para cada configuração da câmera.



Fonte: Elaborado pelo autor.

Como mostrado na Figura 14, as câmeras que estão com a orientação no sentido do ponto tridimensional estão na cor verde e as demais na cor vermelha. Apenas a primeira configuração possui as duas câmeras orientadas para o ponto, sendo assim, corresponde a solução para essa matriz essencial.

A matriz de pose da câmera pode ser representada pela matriz intrínseca (\mathbf{K}) e sua orientação como demonstrado na Equação 2.30.

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \quad (2.30)$$

2.3.5 Triangulação

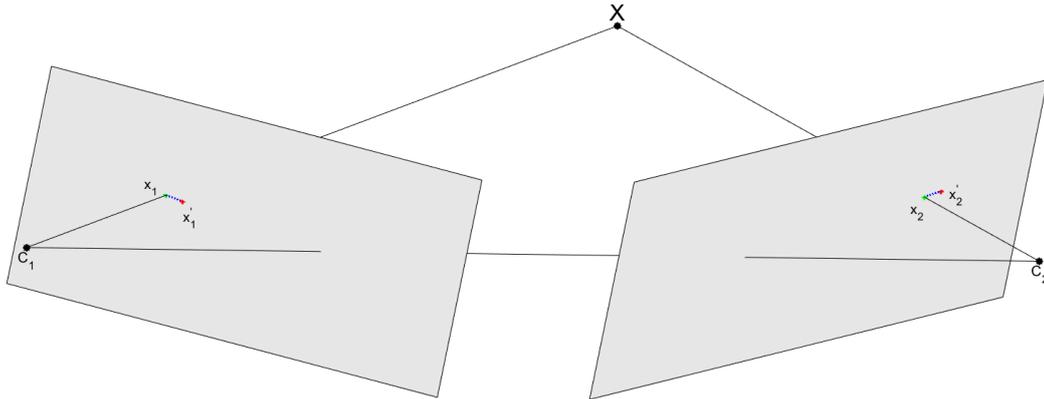
A triangulação é o processo responsável por transformar correspondências de pontos das imagens em pontos no espaço tridimensional. Baseado em modelos matemáticos simplificados da câmera, a triangulação é capaz de retornar valores aceitáveis, tendo um melhor desempenho quando utiliza-se matrizes de poses das câmeras mais precisas (FÖRSTNER; WROBEL, 2016).

Para um par de imagem com matrizes de poses definidas por \mathbf{P}_1 e \mathbf{P}_2 , e pontos correspondentes $\mathbf{x}_1 = [u_1, v_1, 1]^T$ e $\mathbf{x}_2 = [u_2, v_2, 1]^T$ do espaço da imagem, a triangulação τ é capaz de identificar seu ponto 3D equivalente $\mathbf{X} = [X, Y, Z, 1]^T$, como demonstrado na Equação 2.31.

$$\mathbf{X} = \tau(\mathbf{x}_1, \mathbf{x}_2, \mathbf{P}_1, \mathbf{P}_2) \quad (2.31)$$

Normalmente existem erros de aferições na captura das imagens, distorcendo os raios retroprojetivos dos pontos, ou seja, a relação geométrica entre um ponto 2D e um ponto 3D, dado pela Equação 2.8b, além da geometria epipolar descrita pela Equação 2.20 não são satisfeitas (HARTLEY; ZISSERMAN, 2003). A Figura 15 demonstra quais eram os pontos ideais em verde, e os valores aferidos em vermelho.

Figura 15 – Erro de aferição na captura das Imagem.



Fonte: Elaborado pelo autor.

A fim de contornar esse efeito, pode-se assumir um ponto no espaço tridimensional $\hat{\mathbf{X}}$, com suas coordenadas próximas ao mundo real \mathbf{X} , apresentando um pequeno erro quando reprojeto na imagem. Sendo assim, a triangulação deve encontrar um $\hat{\mathbf{X}}$ com mínimo erro de reprojeção, além ser invariante a transformações. Para isso, deve-se estimar uma solução linear e posteriormente utilizar uma otimização (HARTLEY; ZISSERMAN, 2003).

Na triangulação linear, aplica-se um produto cruzado entre o ponto de cada imagem \mathbf{x}_i e o vetor definido por $\mathbf{P}_n \hat{\mathbf{X}}$, eliminando com isso, o fator de escala linear. Para cada imagem, é gerado duas equações lineares independentes, como demonstrado Equação 2.32, onde \mathbf{P}_n^{jT} , corresponde a Matriz de pose da imagem n, coluna j.

$$\mathbf{A} = \begin{bmatrix} \mathbf{xP}_1^{3T} - \mathbf{xP}_1^{1T} \\ \mathbf{yP}_1^{3T} - \mathbf{xP}_1^{2T} \\ \vdots \\ \mathbf{xP}_n^{3T} - \mathbf{xP}_n^{1T} \\ \mathbf{xP}_n^{3T} - \mathbf{xP}_n^{2T} \end{bmatrix} = 0 \quad (2.32)$$

Utilizando o SVD, é possível obter a coordenada tridimensional $\hat{\mathbf{X}}$, que é determinado pelo autovetor correspondente ao menor autovalor. A fim de um refinamento no ponto obtido, é aplicada uma técnica de otimização com o objetivo de minimizar a distância de *Sampson* da reprojeção dos pontos nas imagens. A definição do ponto

tridimensional ajustado é demonstrada na Equação 2.33.

$$\mathbf{X} = \hat{\mathbf{X}} - \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}\boldsymbol{\epsilon} \quad (2.33)$$

onde a matriz jacobiana \mathbf{J} e o erro de *Sampson* $\boldsymbol{\epsilon}$ são descritos pelas Equação 2.34a e 2.34b.

$$\boldsymbol{\epsilon} = \mathbf{x}_2^T \mathbf{F} \mathbf{x} \quad (2.34a)$$

$$\mathbf{J} = \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{x}} = \left[(\mathbf{F}^T \mathbf{x}_2)_1 \quad (\mathbf{F}^T \mathbf{x}_2)_2 \quad (\mathbf{F} \mathbf{x}_1)_1 \quad (\mathbf{F} \mathbf{x}_1)_2 \right] \quad (2.34b)$$

2.3.6 *Bundle adjustment* (BA)

Como citado na Seção 2.3.5, podem existir erros na aferição da imagem, fazendo com que algumas relações epipolares não sejam respeitadas. Visto isso, quando utiliza-se dessas relações para a estimativa dos parâmetros das câmeras, seus valores poderão apresentar inconsistência devido à propagação de erro (FÖRSTNER; WROBEL, 2016).

Bundle adjustment (BA) é um método unificado para estimar os parâmetros intrínsecos (distância focal, ponto principal, etc.) e extrínsecos (rotação, translação) das câmeras, além da coordenada 3D do ponto de uma cena (FÖRSTNER; WROBEL, 2016). Utilizando um método de otimização, esses parâmetros são estimados através da minimização de uma função de custo que quantifica o erro de reprojeção (SCHONBERGER; FRAHM, 2016).

O erro de reprojeção é dado pela distância euclidiana entre o ponto da imagem \mathbf{x}_1 , correspondente ao ponto 3D \mathbf{X} , e o ponto \mathbf{x}'_1 correspondente a reprojeção \mathbf{X} no plano da imagem. Como em alguns casos as coordenadas de \mathbf{X} não são conhecidas, não é possível calcular este erro. Uma alternativa muito utilizada é estimar o ponto $\hat{\mathbf{X}}$ relacionando os pontos correspondentes entre as imagens e quantificar a diferença entre os pontos das imagens com as respectivas reprojeções do ponto estimado (HARTLEY; ZISSERMAN, 2003). A função de minimização pode ser observada na Equação 2.35.

$$\min_{\mathbf{P}_i, \hat{\mathbf{X}}^j} \sum_i \sum_j d(\mathbf{P}_i \hat{\mathbf{X}}^j, \mathbf{x}_i^j)^2 \quad (2.35)$$

onde \mathbf{P}_i representa a matriz de pose da câmera i ; $\hat{\mathbf{X}}^j$ representa o j -ésimo ponto tridimensional estimado; \mathbf{x}_i^j representa o ponto da imagem i , correspondente ao j -ésimo ponto tridimensional estimado.

Normalmente a otimização é realizada com o método *Levenberg-Marquardt* (LM), que é capaz de resolver problemas não lineares de mínimos quadrados. Esse método

possui a vantagem de ser tolerante a falta de dados, além de permitir atribuições de covariâncias individuais a cada medição e adicionais restrições para os parâmetros das câmeras. Em contrapartida, esse algoritmo necessita de uma boa inicialização e pode se tornar um problema de minimização extremamente grande devido ao número de parâmetros envolvidos (HARTLEY; ZISSERMAN, 2003).

2.4 *Multi-View Stereo (MVS)*

O MVS é capaz de construir modelos 3D altamente detalhados a partir de um conjunto de imagem, além das informações de suas câmeras. Ele provou ser muito bem sucedido devido à qualidade de sua produção e à flexibilidade da configuração experimental necessária (FURUKAWA; HERNÁNDEZ, 2015).

Esses algoritmos são classificados em quatro abordagens diferentes, sendo elas baseada em voxel (FERRARI; TUYTELAARS; VAN GOOL, s.d.; VOGIATZIS et al., 2007), malha poligonais deformáveis (TOLA; LEPETIT; FUA, 2008), mapas de profundidade múltiplos (GOESELE; CURLESS; SEITZ, 2006; ZACH; POCK; BISCHOF, 2007; VOGIATZIS et al., 2007; ZACH, 2008) e expansão de superfície (CECH; SARA, 2007; FURUKAWA; HERNÁNDEZ, 2015; STRECHA et al., 2008).

As abordagens baseadas em voxel necessitam do conhecimento dos limites do cubo capaz de conter toda a cena. A precisão de seus modelos depende da densidade de seu voxel, portanto, os custos computacionais e de memória são muito altos e não são adequados para modelar cenas em grande escala, como cidades. Algoritmos baseados em malhas poligonais deformáveis possuem uma aplicabilidade limitada devido sua necessidade de um bom ponto de partida para iniciar o processo de otimização correspondente (KAZHDAN; BOLITHO; HOPPE, 2006). Para obtenção de um modelo único 3D, os algoritmos baseados em mapas de profundidade múltiplos geram mapas de profundidade a partir de um conjunto de imagens e realiza a fusão deles. Os métodos baseados na expansão da superfície consistem em expandir uma nuvem de pontos esparsa de forma interativa.

Como visto, esses algoritmos têm como princípio, obterem pontos tridimensionais robustos, através de pontos correspondentes das imagens para gerar a estrutura inicial da reconstrução (SEITZ et al., 2006). Esses pontos normalmente são semelhantes a nuvem de pontos encontradas durante o SFM. É realizado um processo interativo sobre esses pontos, a fim de encontrar vizinhos correspondentes nas imagens e com isso gerar uma nuvem de pontos densa (HABBECKE; KOBELT, 2007).

2.4.1 *Zero Mean Normalized Cross-Correlation (ZNCC)*

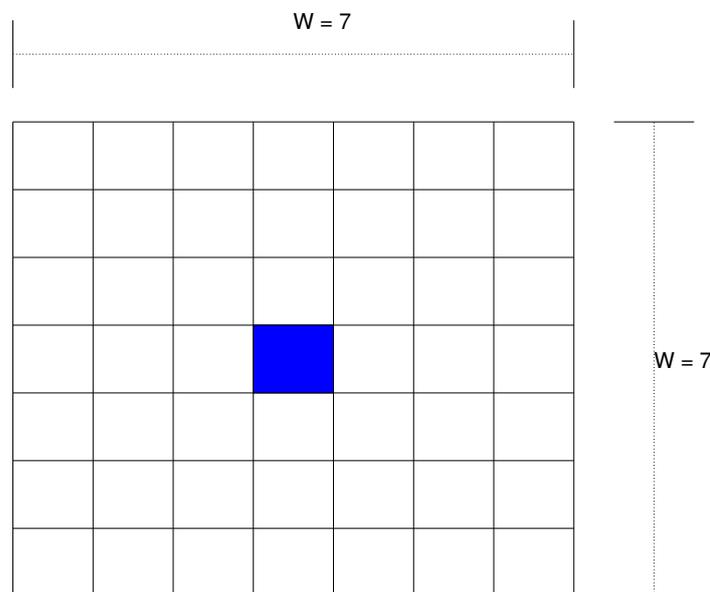
Um processo importante na reconstrução 3D a partir de imagens é identificar partes semelhantes entre as imagens. Os detectores de *features* são capazes de obter alguns pontos

correspondentes, porém, esses detectores concentram-se em ponto com grande variação. A fim de obter mais correspondências nas imagens, outras técnicas são aplicadas, destacando-se a correlação cruzada normalizada (do inglês, *Normalized Cross-Correlation* (NCC)) e a correlação cruzada normalizada com média zero (do inglês, *Zero Mean Normalized Cross-Correlation* (ZNCC)) (DI STEFANO; MATTOCCIA; TOMBARI, 2005).

Essas métricas são amplamente utilizadas devido sua robustez na correspondência de modelos, que permite a tolerância de variações de brilho linear e contrastes, sendo com isso invariante em relação às distorções fotométricas (MATTOCCIA; TOMBARI; DI STEFANO, 2008). Logo, essas métricas são utilizadas para identificar padrões em uma região de uma imagem fonte que mais se assemelha a uma imagem de referência em uma janela de dimensões iguais nas imagens, denominadas *Templates* (ALBUQUERQUE, 2017).

Esses *Templates* representam um pequeno pedaço da imagem, delimitado pelo tamanho da janela W . Para esse tamanho é adotado um valor ímpar, a fim da comparação ser simétrica em todas as direções. A Figura 16 demonstra a configuração de um *Template*, onde o quadrado azul é o ponto de referência e corresponde a seu centro.

Figura 16 – Representação de um *Template*.



Fonte: Elaborado pelo autor.

Graças a subtração da média local, o ZNCC permite uma maior tolerância à variações uniformes de brilho se comparado ao NCC. Essa métrica apresenta, baseada em valores normalizados que variam entre -1 e 1, o quanto um *Template* é semelhante a outro, através do grau de similaridade entre eles (GEORGOULAS; ANDREADIS, 2010). Vale ressaltar que quanto mais próximo de 1 for a resposta dessa métrica, maior é o grau de similaridade.

O cálculo dessa métrica pode ser obtido pela Equação 2.36, que se baseia na correlação cruzada e usa a diferença de intensidade de cada *pixel* em conjunto com a média do *Template* de imagem, de dimensão $\Re^{W \times W}$, onde $\mathbf{I}_n(\mathbf{x}_m)$, corresponde ao *pixel* da imagem n , na coordenada \mathbf{x} .

$$ZNCC = \sum_i^{WW} \frac{(\mathbf{I}_1(\mathbf{x}_1 + i) - \bar{\mathbf{I}}_1(\mathbf{x}_1))(\mathbf{I}_2(\mathbf{x}_2 + i) - \bar{\mathbf{I}}_2(\mathbf{x}_2))}{\sqrt{(\mathbf{I}_1(\mathbf{x}_1 + i) - \bar{\mathbf{I}}_1(\mathbf{x}_1))^2 (\mathbf{I}_2(\mathbf{x}_2 + i) - \bar{\mathbf{I}}_2(\mathbf{x}_2))^2}} \quad (2.36)$$

A fim de simplificar o cálculo dessa métrica, a Equação 2.36 pode ser expandida para a Equação 2.37.

$$T_{x_n}^i = \mathbf{I}_n(\mathbf{x}_n + i) \quad (2.37a)$$

$$S_{x_n} = \sum_i^{WW} T_{x_n}^i \quad (2.37b)$$

$$S_{x_n x_n} = \sum_i^{WW} T_{x_n}^i T_{x_n}^i \quad (2.37c)$$

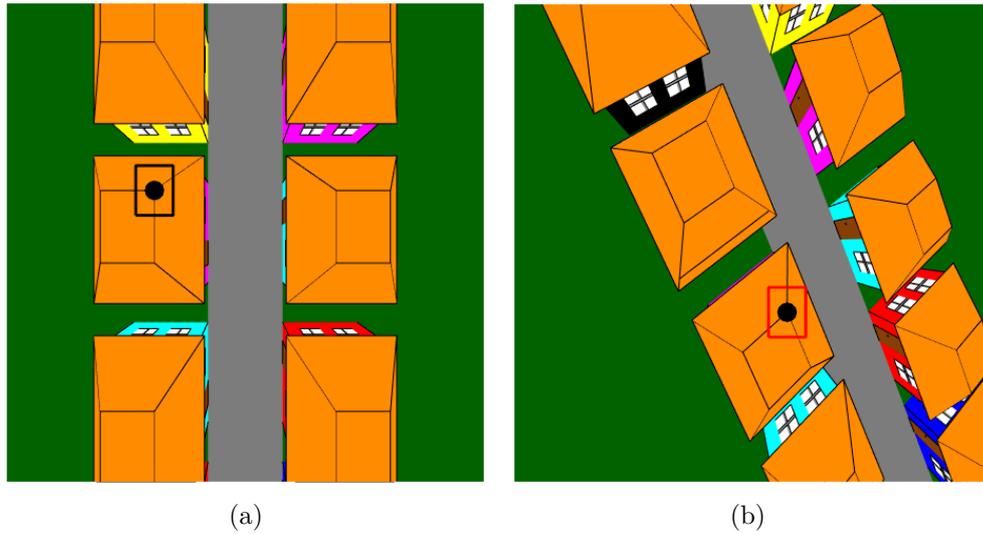
$$ZNCC = \frac{\sum_i^{WW} (T_{x_1}^i T_{x_2}^i) - \frac{S_{x_1} S_{x_2}}{WW}}{\sqrt{\left(S_{x_1 x_1} \frac{S_{x_1} S_{x_1}}{WW}\right) \left(S_{x_2 x_2} \frac{S_{x_2} S_{x_2}}{WW}\right)}} \quad (2.37d)$$

A equação simplificada apresentada na Equação 2.37 é bastante útil quando desejar-se comparar um mesmo *Template* mais vezes, com isso é necessário realizar todas os cálculos novamente, exigindo um esforço computacional menor.

2.4.2 Transformação *affine*

Durante a captura de imagens, as câmeras são rotacionadas e/ou transladadas. Essas variações, fazem com que as imagens mudem as características dos objetos, de acordo com o plano de captura (plano da imagem). Durante a seleção de *Templates* correspondentes para o cálculo de similaridade, essas deformações podem fazer com que a comparação seja prejudicada (LOWE, D. G., 2004), como demonstrado na Figura 17.

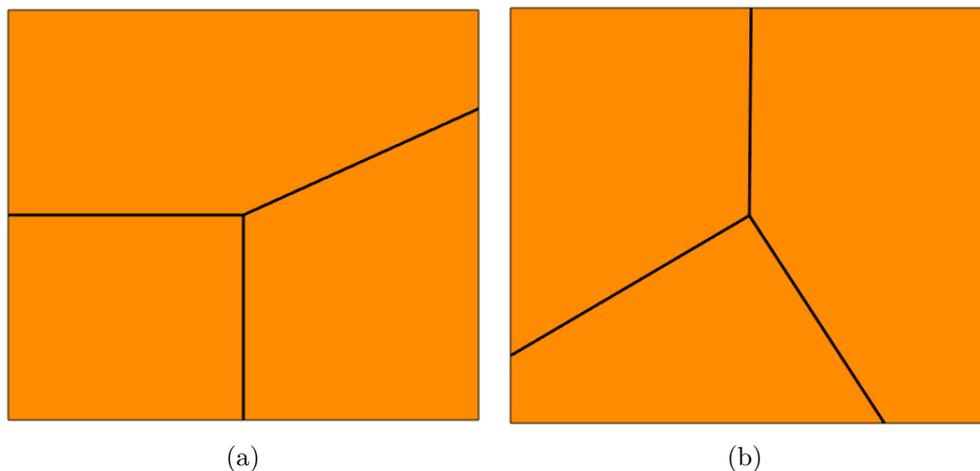
Figura 17 – Erro na seleção do *Template* gerado pela rotação e translação da câmera.



Fonte: Elaborado pelo autor. a) Imagem de referência. b) Imagem de comparação.

O quadrado em preto na Figura 18 (a) corresponde ao *Template* de referência, enquanto o quadrado vermelho na Figura 18 (b) corresponde ao *Template* para comparação. Nesse caso, os *Templates* possuem as mesmas dimensões e o mesmo ponto no centro, entretanto, os seus valores são muitos distintos, como pode ser observado na Figura 18.

Figura 18 – *Templates* de um mesmo ponto para imagens diferentes.



Fonte: Elaborado pelo autor. a) *Template* de referência. b) *Template* de comparação.

A similaridade desses *Templates* é baixa o suficiente para considerar essa correspondência com *outlier*. Para esses casos, é recomendado o uso de uma transformada denominada *Affine*. A transformada *Affine* é uma transformação linear adicionada de uma translação, sendo capaz de projetar um ponto entre planos e representada por uma matriz de dimensão $\mathfrak{R}^{2 \times 2}$ (HARTLEY; ZISSERMAN, 2003).

Uma relação semelhante pode ser observada pela matriz de homografia. A transformada *Affine* pode utilizar dos parâmetros da matriz de homografia, para determinar o plano de projeção (BARATH; HAJDER, 2016).

Uma matriz de homografia H é capaz de projetar um ponto $\mathbf{x}_1 = [u_1, v_1, 1]^T$ de um plano de imagem à seu correspondente $\mathbf{x}_2 = [u_2, v_2, 1]^T$ em outra imagem. A correspondência entre esses pontos pode ser dada pela Equação 2.38, onde \mathbf{h}_r corresponde a linha r da matriz de homografia.

$$u_2 = \frac{\mathbf{h}_1^T [u_1, v_1, 1]^T}{\mathbf{h}_3^T [u_1, v_1, 1]^T} \quad (2.38a)$$

$$v_2 = \frac{\mathbf{h}_2^T [u_1, v_1, 1]^T}{\mathbf{h}_3^T [u_1, v_1, 1]^T} \quad (2.38b)$$

Os parâmetros da transformada *Affine* correspondem à derivada parcial dessas Equações 2.38a e 2.38b (BARATH; HAJDER, 2016). Sendo assim, o primeiro parâmetro da matriz de transformação *Affine* a_{11} pode ser dada pela Equação 2.39.

$$a_{11} = \frac{\partial u_2}{\partial u_1} = \frac{h_{11}\mathbf{h}_3^T [u_1, v_1, 1]^T - h_{31}\mathbf{h}_1^T [u_1, v_1, 1]^T}{(\mathbf{h}_3^T [u_1, v_1, 1]^T)^2} = \frac{h_{11} - h_{31}u_2}{\mathbf{h}_3^T [u_1, v_1, 1]^T} \quad (2.39)$$

Seguindo o mesmo raciocínio, os demais parâmetros são obtidos através da Equação 2.40.

$$a_{12} = \frac{\partial u_2}{\partial v_1} = \frac{h_{12} - h_{32}u_2}{\mathbf{h}_3^T [u_1, v_1, 1]^T} \quad (2.40a)$$

$$a_{21} = \frac{\partial v_2}{\partial u_1} = \frac{h_{21} - h_{31}v_2}{\mathbf{h}_3^T [u_1, v_1, 1]^T} \quad (2.40b)$$

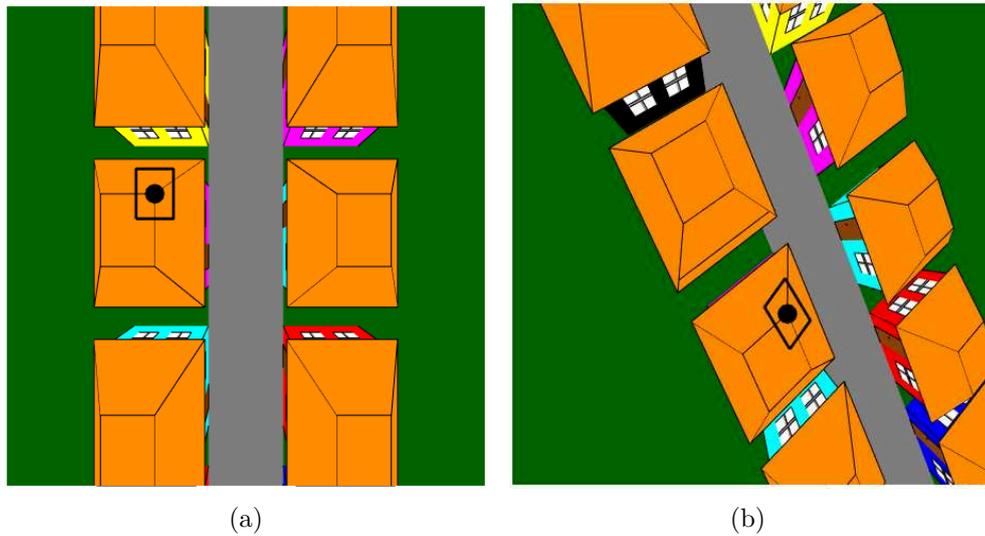
$$a_{22} = \frac{\partial v_2}{\partial v_1} = \frac{h_{22} - h_{32}v_2}{\mathbf{h}_3^T [u_1, v_1, 1]^T} \quad (2.40c)$$

A matriz de transformação *Affine* pode ser dada pela Equação 2.41.

$$aff = \frac{\begin{bmatrix} (h_{11}h_{32} - h_{12}h_{31})v_1 + h_{11}h_{33} - h_{13}h_{31}, & (h_{12}h_{31} - h_{11}h_{32})u_1 + h_{12}h_{33} - h_{13}h_{32} \\ (h_{21}h_{32} - h_{22}h_{31})v_1 + h_{21}h_{33} - h_{23}h_{31}, & (h_{22}h_{31} - h_{21}h_{32})u_1 + h_{22}h_{33} - h_{23}h_{32} \end{bmatrix}}{(h_{31}u_1 + h_{32}v_1 + h_{33})^2} \quad (2.41)$$

A matriz da Equação 2.41 pode ser utilizada para selecionar o *Template* correspondente entre as imagens com uma maior precisão. A Figura 20 (b) mostra um exemplo da transformação *Affine* modificando o *Template* da imagem 18 (b).

Figura 19 – Seleção de *Template* com a transformação *Affine*.

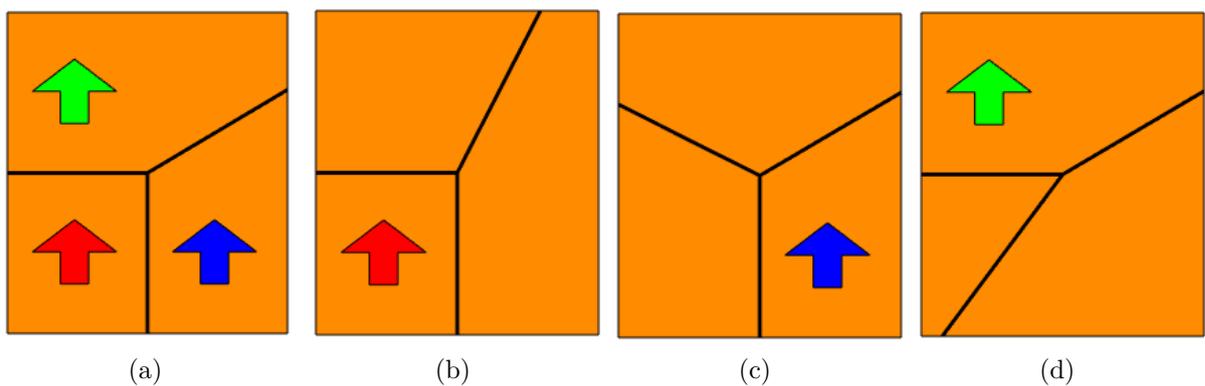


Fonte: Elaborado pelo autor. a) Imagem de referência. b) Imagem de comparação.

A transformação *Affine* utiliza um plano como referência. Esse plano é calculado através da Normal do ponto 3D, como observado na Equação 2.17. Entretanto, as quinas, por exemplo, representam mudança de planos, prejudicando a transformação.

A Figura 20 exemplifica a interferência de mudança de planos, onde o *Template*, representado na Figura 21 (a) corresponde a referência e as setas, nas Figuras 21 (b), 21 (c) e 21 (d), indicam qual plano está sendo utilizado.

Figura 20 – *Templates* selecionados com a transformação *Affine* para diferentes normais.

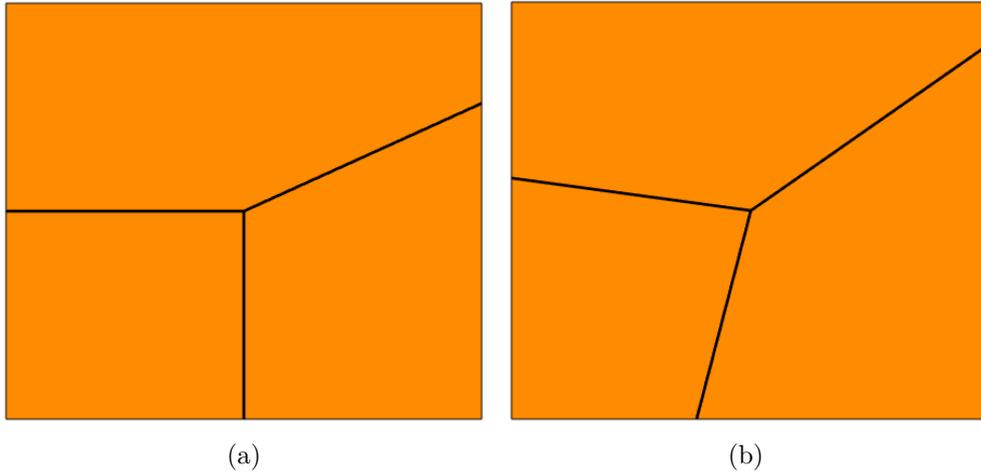


Fonte: Elaborado pelo autor. a) Referência. b) Normal 1. c) Normal 2. d) Normal 3.

Percebe-se que o *Template* tem um comportamento satisfatório no plano de referência, porém se perde nos demais. A fim de encontrar uma maior representatividade para casos como esses, deve-se identificar a Normal que gere o melhor plano para o conjunto de

Template. A Figura 21 demonstra um par de *Templates* com a melhor configuração para esse exemplo. É possível perceber divergências nos *Templates*, porém ambos possuem um comportamento similar.

Figura 21 – *Templates* selecionados utilizando melhor configuração da Normal.



Fonte: Elaborado pelo autor. a) *Template* de referência. b) *Template* de comparação.

2.4.3 Propagação dos Pontos

A utilização de pontos de interesses identificados durante a execução do algoritmo SFM é suficiente para calcular ou rastrear as posições da câmera, mas não para representar a cena ou os objetos na cena, pois apenas reconstrói pontos 3D esparsamente distribuídos. A fim de obter uma melhor representatividade da cena, esses pontos são propagados identificando outras correspondências nas imagens. Uma boa combinação de pontos de interesses iniciais é suficiente para provocar uma avalanche de correspondências nas imagens (LHULLIER; QUAN, 2005).

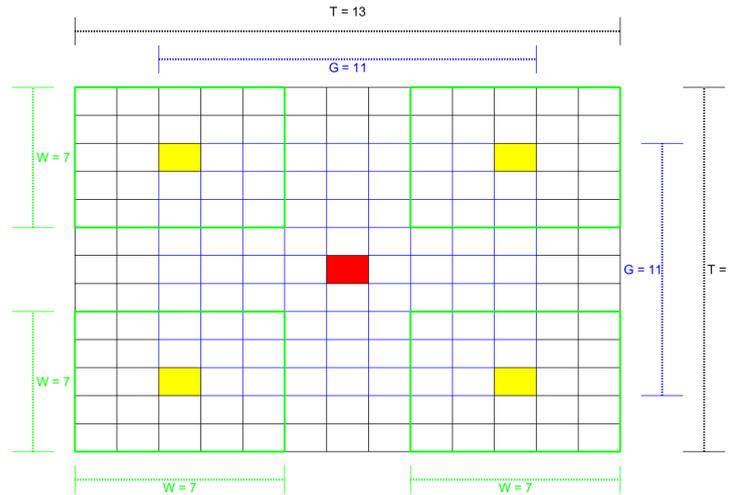
Uma metodologia muito utilizada é a realização da seleção de *Templates* em volta de cada ponto de interesse e procurar por novas correspondências dentro desses *Templates*. Nessa técnica é utilizado um par de imagens, onde, para cada ponto do *Template* da primeira imagem, é feito uma comparação com os pontos do *Template* da segunda imagem. É válido ressaltar que pontos cuja sua correspondência já tenha sido encontrada, não são utilizados nessa busca.

Utilizando-se dessa metodologia, (YLIMÄKI et al., 2015) utiliza da métrica ZNCC para identificar os novos pontos correspondentes. Ylimaki define um *Template* de busca de dimensão $\mathbb{R}^{G \times G}$, sendo G um valor ímpar que contém os possíveis pontos de propagação. O *Template* da segunda imagem é obtido utilizando a transformação *Affine*, com o intuito de aumentar o número de correspondências.

Para um melhor tempo computacional, esse template é expandido de tal forma

que seja capaz de conter todos os *pixels* utilizados na métrica ZNCC, como demonstrado na Figura 22. O quadrado em vermelho, representa o ponto de referência, ou seja, o ponto a ser expandido. Os pontos em amarelo são os limites do *Template* de vizinhos. Com essa abordagem, não é necessário criar um par de *Templates* para o cálculo da similaridade à cada comparação entre as imagens.

Figura 22 – Configuração do *Template* para propagação de pontos.



Fonte: Elaborado pelo autor.

Para cada novo *match* encontrado no par de imagens, é utilizado a reprojeção para verificar se existe uma similaridade daquele ponto nas outras imagem. Esses pontos são marcados como percorridos e não é realizada uma nova procura de *match* pelo mesmo. O desempenho dessa técnica se limita ao par de imagens selecionados para efetuar a propagação do ponto. Ao realizar a busca por similaridade, deformações, como mostrado na Figura 20, diminuem a performance do algoritmo.

Esse problema é contornado por (FURUKAWA; HERNÁNDEZ, 2015), realizando a expansão do ponto no espaço tridimensional, que garante, também, um aumento a eficiência computacional ao criar um *grid* na imagem. Na busca por vizinhos, é identificado apenas um ponto por posição desse *grid*, reduzindo o gasto de memória.

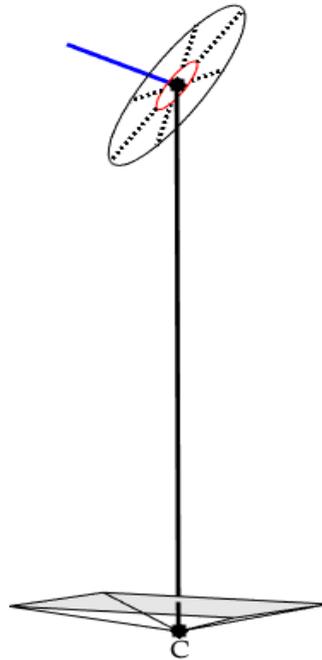
Para cada ponto 3D encontrado, é gerado um pequeno retângulo (*patch*) que representa um pequeno fragmento de uma superfície geométrica, e possui esse ponto em seu centro. É feito um cálculo para estimar qual deslocamento no espaço tridimensional corresponde ao deslocamento de uma posição do *grid* da imagem de referência. Esse cálculo pode ser obtido pela relação da distância entre o ponto de interesse (\mathbf{X}) e o centro ótico da câmera (\mathbf{C}), com a média das distâncias focais (f_x, f_y), multiplicado pelo tamanho

do grid (g), como demonstrado na Equação 2.42.

$$unit = g \frac{2 \|(\mathbf{X} - \mathbf{C})\|}{f_x + f_y} \quad (2.42)$$

A área ao redor do ponto de referência é subdividido em n pedaços, como apresentado na Figura 23, sendo \mathbf{C} o centro óptico da câmera e a reta em azul a normal do *patch*.

Figura 23 – Identificação de vizinhos no espaço tridimensional.



Fonte: Elaborado pelo autor.

O espaço entre cada linha pontilhada corresponde a área de propagação de um novo vizinho. Uma busca na nuvem de ponto é realizada, verificando se aquele espaço já está preenchido, caso contrário, cria-se um ponto naquele intervalo e reprojeta-o nas imagens que possuem o ponto de referência. Caso os pontos da imagem reprojetaos possuam uma similaridade desejada com o ponto da imagem de referência, o vizinho é salvo na nuvem de pontos.

2.4.4 Ajuste da Nuvem de Pontos

Tanto o processo de detecção *feature* quanto o de estimação de nuvem densa podem gerar pontos tridimensionais com erro de aferição. Técnicas de refinamento são empregadas tanto para o ajuste das coordenadas do ponto, quanto para o ajuste da normal do plano que esse pertence (FURUKAWA; HERNÁNDEZ, 2015). Com isso,

esses refinamentos melhoram a precisão das reconstruções e tornam a expansão menos dependente da qualidade dos pontos iniciais (YLIMÄKI et al., 2015).

(YLIMÄKI et al., 2015) apresenta um método de refinamento baseado no algoritmo de alinhamento de imagem proposto por Lucas-Kanade, onde os *Templates* das imagens são alinhados, minimizando a soma da diferença ao quadrado entre eles. Utilizando-se do gradiente da imagem, é calculado o passo para ajuste dos pontos, como demonstrado na Equação 2.43.

$$\mathbf{M}(\mathbf{x}) = \nabla \mathbf{I} \frac{\partial \mathbf{T}_2}{\partial \mathbf{X}} - \mathbf{T}_1 - 1 \quad (2.43a)$$

$$\Delta \mathbf{X} = \left[\sum_{\mathbf{x}} \mathbf{M}^T(\mathbf{x}) \mathbf{M}(\mathbf{x}) \right]^{-1} \times \sum_{\mathbf{x}} \mathbf{M}^T(\mathbf{x}) [\mathbf{T}_2 - \lambda_1 \mathbf{T}_1 - \lambda_0] \quad (2.43b)$$

onde $\nabla \mathbf{I}$ corresponde ao gradiente da imagem; $\frac{\partial \mathbf{T}_2}{\partial \mathbf{X}}$ corresponde a Jacobiana do segundo *Template*; \mathbf{T}_1 corresponde ao *Template* da imagem de referência; e \mathbf{T}_2 corresponde ao *Template* da segunda imagem.

Como a otimização é iterativa e requer o cálculo repetido de gradientes em todas as regiões da superfície, ela gera um gargalo que afeta a eficiência computacional do refinamento (LI et al., s.d.).

Utilizando da métrica de similaridade ZNCC em conjunto com a geometria epipolar, (FURUKAWA; HERNÁNDEZ, 2015) realiza o refinamento dos pontos de forma simples e eficiente. Dado um ponto $\hat{\mathbf{X}}$ estimado no tridimensional, que corresponde a um ponto \mathbf{x}_1 pertencente a imagem de referência, é calculado o raio (\mathbf{r}) entre o ponto $\hat{\mathbf{X}}$ até o centro da câmera de referência \mathbf{C} . Fixando esse raio como eixo de profundidade, é possível deslocar-se na linha epipolar de uma segunda imagem a fim de encontrar uma correspondência, como pode ser visto na Figura 12. Com essa metodologia, é possível realizar o ajuste das três coordenadas do ponto $\hat{\mathbf{X}}$ variando apenas uma variável de escala como mostra a Equação 2.44.

$$\mathbf{X} = \hat{\mathbf{X}} + \lambda \mathbf{r} \quad (2.44)$$

onde λ corresponde a variável de escala e \mathbf{X} ao ponto 3D refinado. Entretanto, como a linha epipolar da segunda imagem é fixa, a correspondência pode estar fora dessa linha. É válido lembrar da possibilidade de se obter a linha epipolar através da normal do plano utilizado. Sendo assim, uma forma de encontrar essa correspondência é variando a Normal, fazendo como que a linha epipolar se desloque.

A fim de reduzir o número de variáveis a ser otimizada, o vetor Normal é representado por dois ângulos α e θ , resultando em um problema com três graus de liberdade.

2.4.5 Filtragem da Nuvem de Pontos

Segundo (KUHN et al., 2019), um dos grandes desafios do MVS é a reconstrução de nuvens de pontos 3D da maneira mais completa possível, minimizando o número de *outliers* e mantendo uma alta precisão desses pontos. Para a remoção dos *outliers*, Kuhn realiza a clusterização da nuvem densa pela proximidade dos pontos e da semelhança entre as direções de suas normais.

(KUHN et al., 2019) determina que *clusters* que possuam um número pequeno de pontos, podem ser considerados *outlier*, pois não representam nenhuma superfície expressiva. Uma abordagem bem similar é realizada aplicando uma filtragem de superfícies quádricas conhecidas.

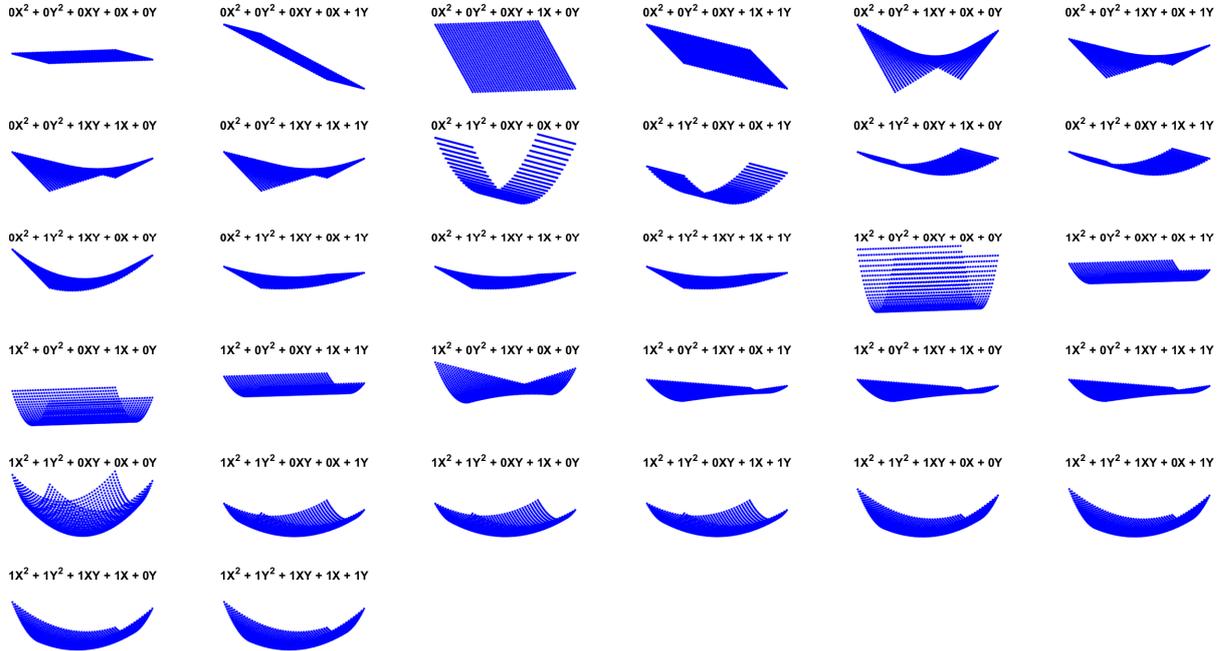
A filtragem por forma quádrica, consiste em identificar um padrão quádrico em um conjunto de vizinhos, ou seja, pontos próximos devem possuir algum formato geométrico. Para identificar se os pontos se enquadram em um formato geométrico, pode-se utilizar a Equação 2.45.

$$\begin{bmatrix} \Delta X_1^2 & \Delta Y_1^2 & \Delta X_1 \Delta Y_1 & \Delta X_1 & \Delta Y_1 \\ \Delta X_2^2 & \Delta Y_2^2 & \Delta X_2 \Delta Y_2 & \Delta X_2 & \Delta Y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta X_n^2 & \Delta Y_n^2 & \Delta X_n \Delta Y_n & \Delta X_n & \Delta Y_n \end{bmatrix} = \begin{bmatrix} \Delta Z_1 \\ \Delta Z_2 \\ \vdots \\ \Delta Z_n \end{bmatrix} \quad (2.45)$$

onde ΔX_n representa a diferença entre o ponto 3D analisado e o vizinho n , no eixo X; ΔY_n representa a diferença entre o ponto 3D analisado e o vizinho n , no eixo Y; e ΔZ_n representa a diferença entre o ponto 3D analisado e o vizinho n , no eixo Z.

Nesse caso, para obter a equação quadrática, é necessário, para cada ponto da nuvem, cinco pontos vizinhos selecionados. A Figura 24 demonstra algumas formas geométricas que podem ser obtidas com essas equações.

Figura 24 – Formas Geométricas utilizadas para filtra *outliers*.



Fonte: Elaborado pelo autor.

É esperado que os demais pontos próximos ao de referência pertença a essa superfície, ou seja, retorne um valor nulo quando inserido seus valores na equação obtida. A fim de verificar se o ponto é um *outlier*, a média dos erros desses pontos são mensurados e caso esse valor seja maior que um estipulada, o ponto de referência é removido.

2.5 APLICAÇÕES ENVOLVENDO AS TÉCNICAS SE SFM E MVS EM CONJUNTO COM UM UAV

Os algoritmos SFM e MVS com os UAVs tornaram-se uma ferramenta popular na geociência para modelar paisagens complexas sob demanda, permitindo estudos de alterações topográficas de alta resolução, com um custo mínimo (MEINEN; ROBINSON, 2020). Diante disso, trabalhos foram desenvolvidos no último ano utilizando-se dessa ferramenta.

(MEINEN; ROBINSON, 2020) apresentou um trabalho de mapeamento de erosão e deposição em uma paisagem agrícola. Foi realizado um estudo da melhor configuração da captura das imagens durante uma missão do UAV, sendo comparado o desempenho das técnicas SFM e MVS entre imagens nadir (capturadas com a câmera direcionada para o solo) e imagens oblíquas (com um ângulo de 15 graus de inclinação em relação ao nadir), além dessas duas configurações utilizadas em conjunto.

Foram realizadas três campanhas, ou seja, mapeamentos em três dias diferentes.

Diante dos resultados obtidos, Meinen concluiu que apesar de se obter respostas altamente precisas com a incorporação das imagens oblíquas, o uso dessas imagens provou ser um prejuízo significativo para a geração de certos modelos. A Figura 25 apresenta as reconstruções encontradas para cada campanha.

Figura 25 – Reconstruções da três campanhas realizada por Meinen.

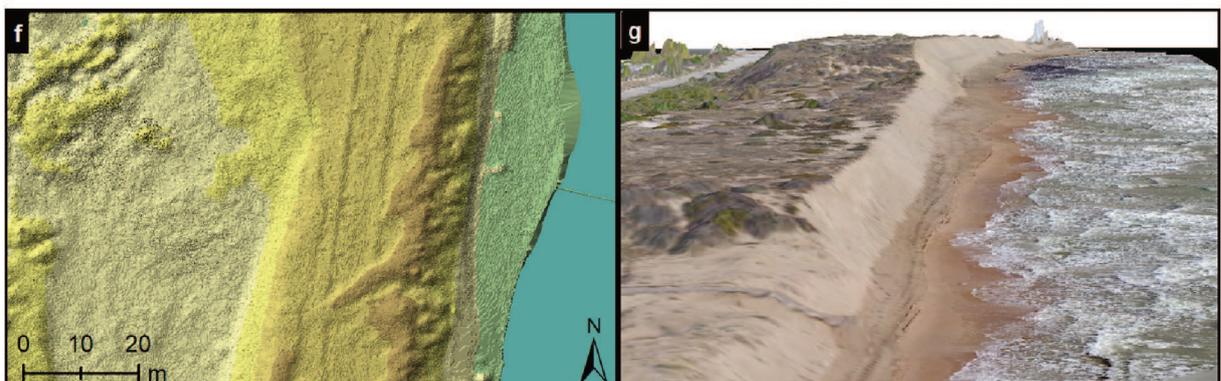


Fonte: (MEINEN; ROBINSON, 2020)

Em outra abordagem, (PAGÁN et al., 2019) apresenta um monitoramento do sistema de dunas da praia de Guardamar del Segura - Espanha. Nesse monitoramento foi verificado a evolução histórica do sistema de dunas e sua relação com a erosão costeira. Para isso, foram comparadas algumas reconstruções realizadas entre os anos de 2001 à 2016 e a nuvem de pontos gerada pelas técnica SFM e MVS de um conjunto de fotos retiradas por um quadricóptero Phantom 4 em junho de 2017.

O *software* comercial Agisoft PhotoScan foi utilizado para a geração da nuvem de pontos, onde Pagán distribuiu 154 pontos de controle no solo ao longo da zona de estudo, de forma que pelo menos dois pontos de controle fossem visíveis em todas as imagens capturadas. Esses pontos de controle foram necessários para realizar o georreferenciamento preciso dos dados coletados. A Figura 26 ilustra a nuvem de pontos obtida no experimento.

Figura 26 – Nuvem de Pontos obtida por Pagan.



Fonte: (PAGÁN et al., 2019)

Nesse experimento, Pagán encontrou um valor de raiz quadrado do erro médio (RMSE) de 0,173 metros. No eixo Z, foi encontrado um RMSE de 0,136 metros na área das dunas, 0,222 para a areia molhada, 0,083 metros para as estradas e 0,231 para as passarelas. Já o RMSE planimétricos (eixos x e y) apresentou um valor abaixo de 0,10 metros.

Também utilizando o quadricóptero Phantom 4 e o *software* comercial Agisoft PhotoScan, (WARFIELD; LEON, 2019) realizou a estimação do volume de uma floresta de mangue, localizada dentro do Parque de Conservação Weyba Creek, Austrália. Nesse experimento, Warfield compara as metodologias SFM e MVS em conjunto com um UAV, com a técnica de varredura a laser (do inglês, “*Terrestrial Laser Scanning*” - TLS) predominantemente usada para esse fim.

O terreno sobrevoado é delimitado por 3 estados de vegetação, sendo que cada estado encontra-se em uma área de 25x25 metros. Esses estados são: juvenis, constituída por mudas juvenis menores que 1,37 metros; misto, constituído de indivíduos com idade variada; maduros, com indivíduos maiores que 2,5 metros. Para este experimento foram utilizados 11 pontos de controle no solo perto as bordas de cada local. A Figura 27 mostra o resultado obtido por Warfield.

Figura 27 – Reconstruções de toda a área mapeada por Warfield.



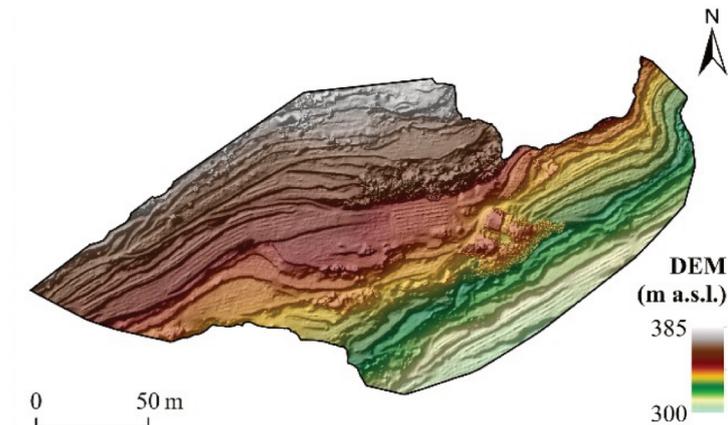
Fonte: (WARFIELD; LEON, 2019)

Segundo Warfield, o tempo de pesquisa foi menor para o SFM-MVS do que o TLS, visto que a área de pesquisa é coberta por lamaceiro dificultando o posicionamento dos equipamentos para TLS. Já os valores estimados por esses métodos tiveram uma diferença de 42,95% na área juvenil, 28,23% na área mista e 12,72% na área adulta. Diante dos resultados, Warfield concluiu que o método de TLS é mais indicado para captura da estrutura da floresta de mangue, enquanto que o SFM-MVS é mais indicado para captura da altura dossel de mangue e o volume subsequente.

Ainda com o *software* comercial Agisoft PhotoScan, (PIJL et al., 2019) apresenta

um projeto de três redes de drenagem de terraços para um vinhedo no vale Valcamonica, na região de Lombardia, Itália. Nesse experimento, os métodos SFM e MVS foram usados com um UAV para estimar uma nuvem densa que posteriormente seria convertida em um modelo digital do terreno (do inglês, “*Digital Terrain Model*” - DTM), ilustrado na Figura 28.

Figura 28 – Modelo digital do terreno levantado por Pijl.



Fonte: (PIJL et al., 2019)

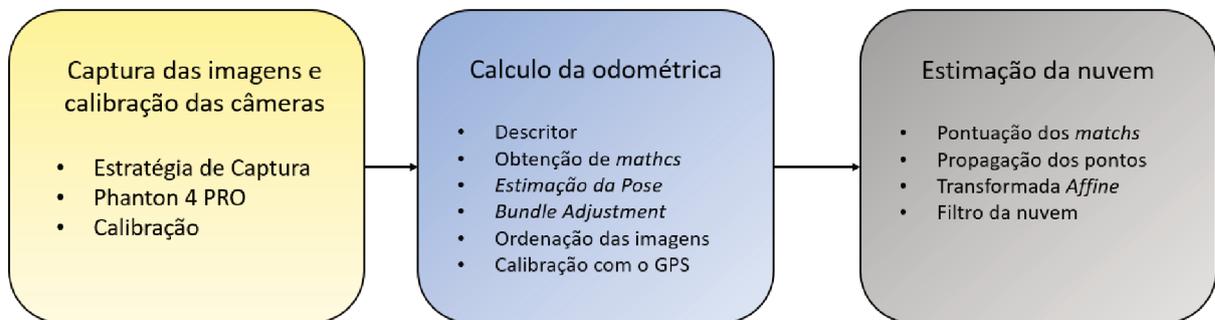
Para esse experimento, Pijl utilizou 16 pontos de controle no solo para auxiliar no cálculo da precisão do modelo encontrado, obtendo um valor de RMSE igual a 0,044 metros. Segundo Pijl, esse valor é satisfatório quando comparado com outros trabalhos nessa área.

3 METODOLOGIA

Nesse capítulo será abordado a evolução da pesquisa, explicando as etapas desenvolvidas de uma forma mais aplicada, assim como os algoritmos de apoio.

Inicia-se com uma seção que descreve o processo de calibração das câmeras e capturas das imagens. Posteriormente é descrito o processo de obtenção dos parâmetros da câmera. Por fim, têm-se uma seção que apresenta o processo de obtenção da nuvem reconstruída, junto com os erros efetuados e seu tratamento no desenvolver do trabalho. O Fluxograma 1 demonstra a sequência das etapas desenvolvidas no trabalho.

Fluxograma 1 – Estratégias adotada no trabalho.



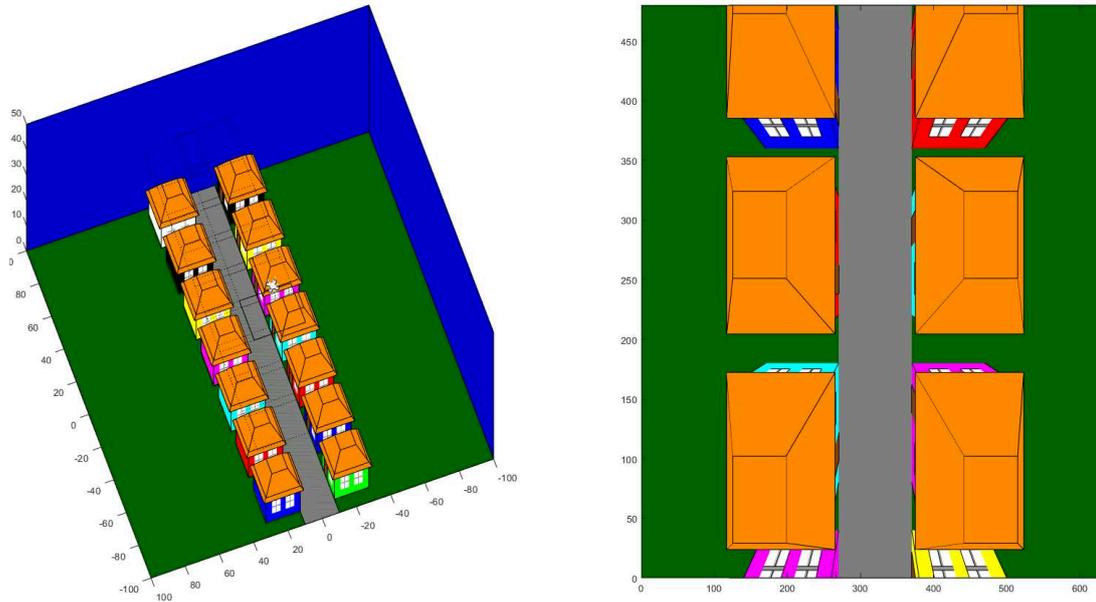
Fonte: Elaborado pelo autor.

3.1 CAPTURA DAS IMAGENS E CALIBRAÇÃO DA CÂMERA

O processo de captura de imagens é realizado com um UAV, Phantom 4 PRO, responsável por sobrevoar uma área estipulada e fotografá-la. O *software* da DJI, capaz de estipular a missão a ser realizada pelo Phantom que permite a realização de uma captura automática. Essa missão corresponde a trajetória e altura que deve ser realizada, assim como a porcentagem de sobreposição entre as imagens. A Figura 29 simula a captura de um drone, para uma missão estipulada.

É válido ressaltar que essas técnicas podem apresentar um rendimento abaixo do esperado na presença de ruídos obtidos na captura de imagens. Como trata-se de um processo que interpreta imagens, faz-se necessário, durante a captura, um controle da iluminação do cenário, evitando mudanças de iluminação entre as imagens. Deve-se ficar atento, também, a sobreposição relativamente alta das imagens, a fim de obter maior correspondência entre elas (HARTLEY; ZISSERMAN, 2003).

Figura 29 – Drone realizando uma missão.



Fonte: Elaborado pelo autor.

O Phantom 4 PRO, demonstrado na Figura 30, é um drone de última geração, capaz de gravar vídeos em 4K a 60fps. Esse drone possui uma conexão estável com o usuário juntamente com um sensor de obstáculos em cinco direções que garante uma boa condução. O Phantom fornece fotos e vídeos com riqueza de detalhes, vibrantes e nítidos, ao mesmo tempo em que mantém a precisão de cores. Isso ocorre devido a câmera com um sensor avançado, uma lente grande-angular otimizada e um bom processador. Suas imagens possuem dimensão 4864x3648 *pixels*, além de algumas informações sobre o voo, como por exemplo as coordenadas do GPS. Essas coordenadas serão utilizadas para determinar a precisão da câmera com escala real.

Figura 30 – Phantom 4 PRO.



Fonte: Elaborado pelo autor.

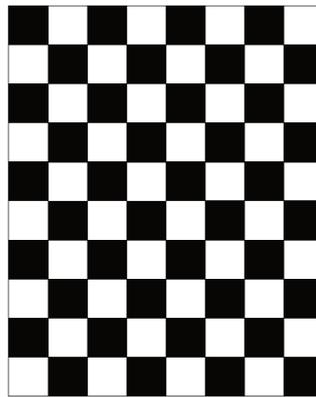
Vale ressaltar que a câmera do Phantom 4 PRO é do tipo *Global Shutter*, com

isso consegue obter imagens com precisão em movimento, por capturar todos o *pixels* no mesmo instante. Para outros tipos de câmera, pode ser necessário realizar a captura das imagens parado.

Visto que todo o processo de captura é realizado com as mesmas configurações da câmera, seus parâmetros intrínsecos para cada imagem terão o mesmo valor e podem ser estimados *a priori*.

Para padronizar as imagens, é realizada antes do voo um processo de calibração. Para a calibração da câmera é utilizado um tabuleiro impresso, com quadrados de arestas de 20 milímetros, como apresentado na Figura 31. Para obtenção de uma calibração precisa, deve-se realizar a captura do tabuleiro de diferentes ângulos e seguir o processo descrito na Seção 2.2.3.

Figura 31 – Tabuleiro utilizado para calibração.



Fonte: Elaborado pelo autor.

3.2 CÁLCULO DA ODOMETRIA DA CÂMERA

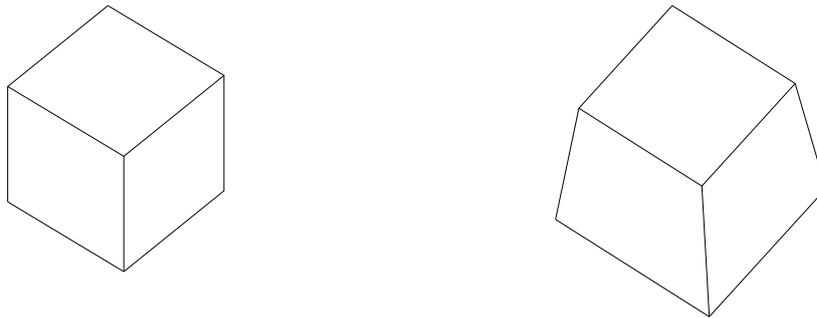
Os parâmetros extrínsecos das câmeras são determinados utilizando a teoria abordada na Seção 2.3. Em primeiro momento, deve-se obter os *matches* entre as imagens. Para isso, as *features* são extraídas das imagens e interpretadas pelo descritor AKAZE. No presente trabalho, foi utilizado algoritmo do AKAZE implementado pelo OpenCV para a extração dessas informações.

Para cada par de imagens, faz-se uma procura por pontos correspondentes entre elas. Essa varredura consiste em encontrar pontos nas duas imagens que possuam seus descritores semelhantes. Baseado nessas correspondências, é possível calcular a matriz fundamental capaz de mapear os pontos de uma imagem na outra (Seção 2.3.3.1), com o auxílio do algoritmo de RANSAC. Posteriormente é realizado uma filtragem dos pontos correspondentes encontrados, utilizando a distância de Sampson, Equação 2.24. Esse filtro é aplicado para evitar que esses pontos prejudiquem o refinamento realizado pelo BA.

Com a manipulação da matriz fundamental, é possível obter a odometria da câmera, como descrito na Seção 2.3.4. Para identificação de uma nuvem de pontos esparsa, uma triangulação dos pontos correspondentes é realizada com o auxílio da matriz de Pose.

Em seguida, o método do BA é aplicado, ajustando tanto a estimação da câmera, quanto da nuvem de pontos. Entretanto, ao realizar o cálculo com um par de imagens, os objetos podem sofrer deformações, uma vez que é possível identificar diferentes configurações que satisfaçam os *matches* encontrados (HARTLEY; ZISSERMAN, 2003). A Figura 32 apresenta a deformação de um cubo, quanto reprojetoado com um conjunto de câmeras estimado incorretamente. Uma maneira de contornar esse erro é aplicar o BA para mais câmeras, com precisão proporcional ao aumento do número de imagens. Para tal, seleciona-se um par inicial, e adiciona-se uma nova imagem para ser ajustada a cada interação do BA.

Figura 32 – Exemplo da deformação de um objeto, ocasionada pela estimação errada da matrizes de Pose das câmeras. O objeto a esquerda corresponde forma real e o da direita o objeto deformado.



Fonte: Elaborado pelo autor.

Como no presente trabalho não foi estipulado uma ordem para inserção das imagens, foi realizado um mapeamento a fim de encontrar a melhor sequência de imagens. Com o intuito de exemplificar o mapeamento, a Tabela 1 demonstra os *matches* obtidos em um conjunto de três imagens. As imagens que não possuem o ponto são representadas pelo sinal de subtração (-).

Tabela 1 – Exemplos de pontos correspondente em um conjunto de três imagens.

C_1		C_2		C_3	
u_1	v_1	u_2	v_2	u_3	v_3
1	5	10	12	-	-
40	80	200	180	-	-
12	52	26	80	-	-
40	80	-	-	78	90
94	350	-	-	150	200
1	5	-	-	18	32
-	-	90	65	150	200
-	-	200	180	78	90
-	-	26	80	42	30
140	280	240	360	-	-
300	500	500	700	-	-
89	160	-	-	340	420

Fonte: Elaborado pelo autor.

É possível observar que alguns pontos estão sendo repetidos na mesma imagem, ou seja, eles estão presentes nas três imagens. Para um mapeamento, os pontos são organizados para obter uma arquitetura similar à mostrada na Tabela 2.

Tabela 2 – Exemplos de pontos correspondente mapeados em um conjunto de três imagens.

C_1		C_2		C_3	
u_1	v_1	u_2	v_2	u_3	v_3
1	5	10	12	18	32
40	80	200	180	78	90
12	52	26	80	42	30
94	350	90	65	150	200
140	280	240	360	-	-
300	500	500	700	-	-
89	160	-	-	340	420

Fonte: Elaborado pelo autor.

Com essa nova configuração, ao projetar um par de pontos correspondentes de duas imagens, a reprojeção em uma terceira imagem pode ser validada. O par de câmeras que possui maior número de correspondência é selecionado como par inicial, ou seja, C_1 e C_2 para o exemplo acima. A próxima imagem a ser inserida, corresponde a imagem que possui o maior número de *matches* com uma das que já fazem parte do grupo selecionado. Esse processo se repete até que todas as imagens tenham sido inseridas.

Com a correspondência de mais imagens para um mesmo ponto projetado no espaço tridimensional, é permitido um ajuste mais robusto no BA. Outra técnica adotada

para um melhor desempenho no BA é ao inserir uma nova imagem para ajuste, realizar ajuste apenas dos parâmetros extrínsecos da câmera. Para isso, são utilizados os pontos 3D ajustados na iteração anterior. Posteriormente é realizado o ajuste em conjunto dos parâmetros com os pontos 3D de todas as imagens selecionadas.

Como dito anteriormente, o processo de SFM não é capaz de identificar a escala real somente baseado na imagens, impedindo a identificação das dimensões de um objeto. Para solucionar esse problema, foi proposto uma nova etapa de BA, utilizando as coordenadas de GPS de cada imagem. Nessa etapa, deve-se minimizar o erro das odometrias calculadas com a coordenada do GPS.

Apartir de uma imagem assumida como referência é possível calcular a distancia euclidiana da variação entre as coordenadas de GPS de cada imagem em relação a imagem de referência (d_{GPS}^i), assim como a distancia euclidiana da variação entre os centros ópticos estimados das mesmas (d_C^i). A diferença entre as distancias euclidianas d_{GPS}^i e d_C^i devem ser minimizado como demonstra 3.1.

$$d_{GPS}^i = d(GPS_{ref} - GPS_i)^2 \quad (3.1a)$$

$$d_C^i = d(C_{ref} - C_i)^2 \quad (3.1b)$$

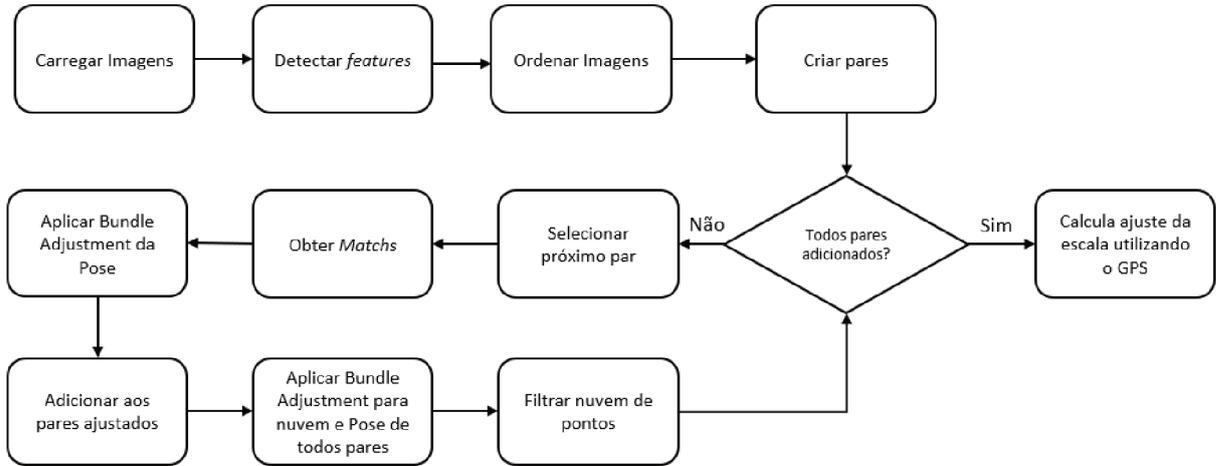
$$\min_{\epsilon} \sum_i d_{GPS}^i - \epsilon d_C^i \quad (3.1c)$$

onde GPS_{ref} representa a coordenada de GPS da imagem de referência; GPS_i corresponde a coordenada de GPS da i -ésima imagem; d_{GPS}^i a distancia euclidiana entre coordenadas de GPS da imagem de referência e a i -ésima imagem; C_{ref} representa a coordenada do centro óptico estimado da imagem de referência; C_i corresponde a coordenada do centro óptico estimado da i -ésima imagem; d_C^i a distancia euclidiana entre coordenadas dos centros ópticos da imagem de referência e a i -ésima imagem; ϵ corresponde ao ajuste da escala.

Ao fim desse procedimento é retornado um ganho (ϵ) capaz de aproximar os dados obtidos na reconstrução de seus valores reais. Durante os experimentos, percebeu-se melhores respostas ao inserir esse ganho no final de todo processo de estimação da nuvem densa.

O Fluxograma 2 foi adotado para a estimação da odometria da câmera.

Fluxograma 2 – Estimação de odometria da câmera.



Fonte: Elaborado pelo autor.

3.3 ESTIMAÇÃO DA NUVEM DENSA

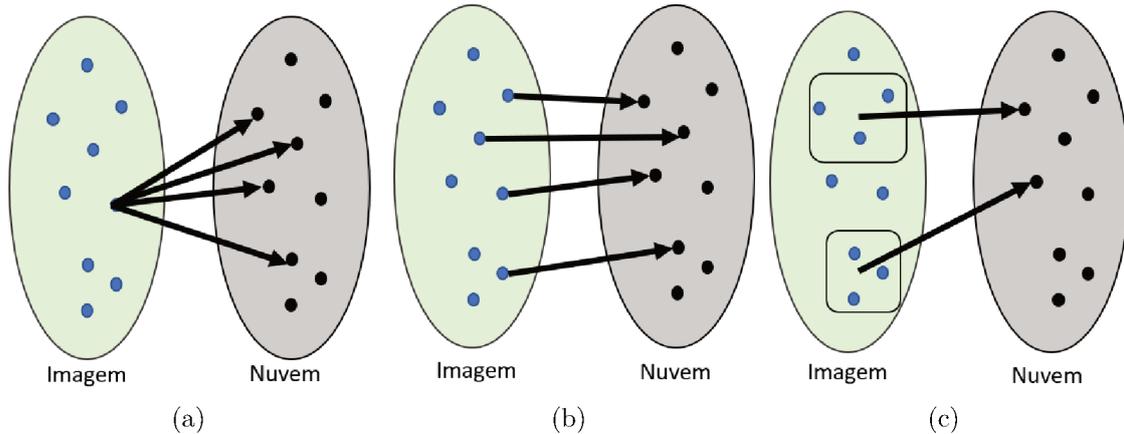
Essa etapa se dedica em expandir os *matches* encontrados, a fim de dar mais detalhes para a reconstrução. Os pontos já encontrados são usados como pontos iniciais (*Seeds*) e para busca de vizinhos ao seu redor. Esses *Seeds* são ordenados em uma lista de prioridades pelo valor de sua pontuação, calculado pela Equação 3.2. Os pontos com maior pontuação são os primeiros a serem propagados. Por possuírem características mais definidas, aumenta-se a chance da obtenção de vizinhos mais precisos.

$$S = (ZNCC - th)/(1 - th) \quad (3.2)$$

onde S corresponde a pontuação para um ponto 3D estimado; $ZNCC$ corresponde à similaridade retornada pela métrica; th corresponde ao valor mínimo de similaridade, para determinar se o ponto é *inlier*.

Nesse contexto existem diversas configurações para a propagação dos pontos tridimensionais. A Figura 33 apresenta diagramas para auxiliar na compreensão dos diferentes comportamentos de cada técnica, que serão abordados nessa seção.

Figura 33 – Exemplo dos diferentes comportamento ao realizar a propagação.



Fonte: Elaborado pelo autor. a) Padrão 1. b) Padrão 2. c) Padrão 3.

Na Figura 34 (a) é apresentado o comportamento do algoritmo baseado em projeção através de pares de imagens. A partir de um ponto específico da imagem, outros pontos na nuvem são encontrados. Na Figura 34 (b), pode-se observar o comportamento da propagação *pixel a pixel*. Para cada *pixel* na imagem, é gerado um ponto na nuvem. Por fim, a Figura 34 (c) mostra a propagação por subáreas. Nela um subconjunto de pontos da imagem possui apenas um ponto na nuvem.

Em uma primeira abordagem, pode-se realizar uma propagação baseada em *Template*, onde para cada par de imagem, os pontos são propagados respeitando a métrica do ZNCC, gerando uma nuvem de pontos. As nuvens geradas posteriormente são concatenadas para gerar a resposta final. Essa metodologia apresenta um comportamento considerável para cenas com pouca variação de planos, onde o algoritmo cria uma nuvem com pouca propagação para imagens que possuem um valor de rotação acentuado.

Para solucionar problemas relacionados à orientação das imagens, utilizou-se da transformada *Affine* para obtenção de *Templates* com maior semelhança entre as imagens. Para a utilização dessa transformada, é necessário o cálculo da normal N do plano que cada *Seed* pertence. Uma alternativa é a utilização de variáveis de ângulo θ e escala λ retornadas pelo descritor AKAZE, como demonstrado na Equação 3.3.

$$\theta_f = \theta_1 - \theta_2 \quad (3.3a)$$

$$\lambda_f = \frac{\lambda_1}{\lambda_2} \quad (3.3b)$$

$$Aff = \begin{bmatrix} \lambda_f \cos(\theta_f) & -\lambda_f \sin(\theta_f) \\ \lambda_f \sin(\theta_f) & \lambda_f \cos(\theta_f) \end{bmatrix} \quad (3.3c)$$

Como esses parâmetros são obtidos com as informações passadas pela AKAZE

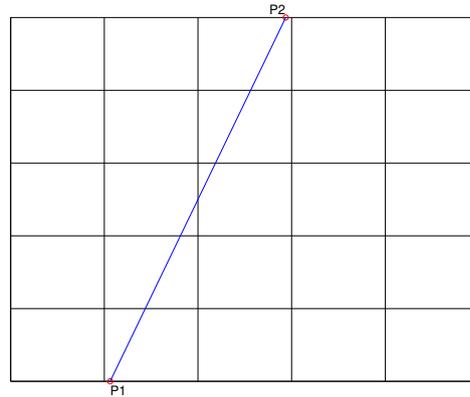
e os demais pontos identificados durante a expansão não possuem essas informações, a Equação 3.3 é utilizada apenas para estimar as condições iniciais das *Seeds*. Os demais pontos obtidos na expansão passam por um processo de refinamento, ajustando a normal N do mesmo. Para isso, foi empregado a metodologia de refinamento demonstrada por (YLIMÄKI et al., 2015), como descrito na Seção 2.4.4. Em um primeiro momento esse refinamento foi aplicado a pares de imagens que posteriormente seriam fundidos para gerar uma nuvem completa, com um grande número de pontos 3D para um mesmo ponto da imagem. Isso gera um conjunto de pontos elevados que não refletem em uma melhora na obtenção das informações do cenário.

No segundo momento, essa técnica foi expandida inserindo o método de propagação proposta por (YLIMÄKI et al., 2015), descrita na Seção 2.4.3. Um dos principais benefícios de sua implantação é a verificação da existência de pontos correspondentes além do par de imagem em análise, evitando o super dimensionamento dos pontos, como citado anteriormente. Apesar de solucionar o super dimensionamento dos pontos, essa técnica tende a capturar um ponto tridimensional para cada *pixel* da imagem, o que gera pontos muitos próximos de forma desnecessária, além de exigir um gasto computacional alto para o refinamento dos pontos.

Como apresentado na Seção 2.4.3, (FURUKAWA; HERNÁNDEZ, 2015) propôs a propagação de pontos pelo espaço tridimensional, procurando por áreas no *grid* da imagem que estejam livres, estimando para cada local um ponto tridimensional, ou seja, realiza a procura de um ponto tridimensional para um sub-agrupamento de *pixels*. Esse procedimento permite alterar a quantidade de pontos encontrados para uma pequena área do mapa variando apenas o tamanho do *grid* da imagem. Além disso, imagens com alta resolução, como a utilizada nesse trabalho, ocorre uma aceleração do processo de obtenção da nuvem de pontos densa com minimização de perdas de informações.

Entretanto, em regiões com pouca textura, ou em cenários que repetem padrões em diferentes partes, essa técnica apresenta erro na propagação, gerando em alguns casos, duplicação de superfícies. Isso ocorre quando pontos de locais diferentes são identificados como *match*, gerando um ponto com a profundidade incorreta. Ao continuar o processo de expansão, esse ponto estimado incorretamente influenciam na obtenção de novos pontos.

A fim de minimizar esse efeito, foi proposto um ajuste no método de refinamento, estipulando os limites da propagação dos pontos durante o processo de otimização. Baseado nos modelos de propagação por *Templates*, é estipulado um limite em *pixels* de propagação dos vizinhos na imagem de referência, como demonstrado na Figura 34.

Figura 34 – Limites para expansão de uma *feature*.

Fonte: Elaborado pelo autor.

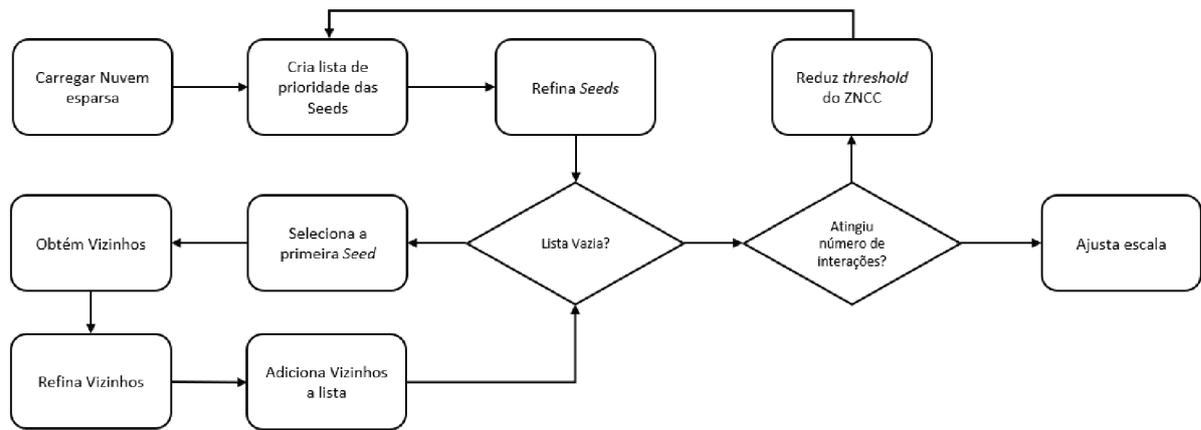
Deve-se, inicialmente, estipular o tamanho do *Template* a ser propagado. Para um ponto tridimensional X que deseja-se realizar sua propagação, é gerando um *Template* ao redor de sua projeção(x) na imagem de referência. O *Template* é projetado em cada imagem que possua esse ponto utilizando a transformação *Affine*.

Em seguida é estimada a linha epipolar(linha em azul na Figura 34) para X em cada imagem(i) e calculado os pontos de interseção(P_1^i e P_2^i) entre essas linhas epipolares com o *Template* projetados. Aplica-se triangulações entre o ponto bidimensional da imagem de referencia(x) com os pontos de interseção calculados(P_1^i e P_2^i), obtendo novos valores tridimensionais. O maior e menor valor do eixo Z são selecionados. Os limites de propagação correspondem a diferença da profundidade de X , para os valores selecionados. Esses limites são utilizados para determinar o *range* da variável λ na Equação 2.44.

Após a propagação dos pontos, os filtros apresentados na Seção 2.4.5 são aplicados à nuvem de pontos para a eliminação dos *outliers*. Para obtenção de uma nuvem de pontos mais rica em detalhes, esse procedimento de propagação é repetido por três vezes, sendo que a cada iteração, o valor de similaridade requerido entre um par de pontos da imagem é reduzido, encontrando assim mais correspondências. Essa metodologia é adotada, para que os pontos mais confiáveis sejam propagados primeiro.

Por fim, é aplicado o ajuste da escala calculado na etapa de estimação da odometria da câmera, para transformar a nuvem para escala real. A etapa de propagação da versão final do algoritmo desenvolvido pode ser vista no Fluxograma 3.

Fluxograma 3 – Obtenção da nuvem densa.



Fonte: Elaborado pelo autor.

4 RESULTADOS E DISCUSSÃO

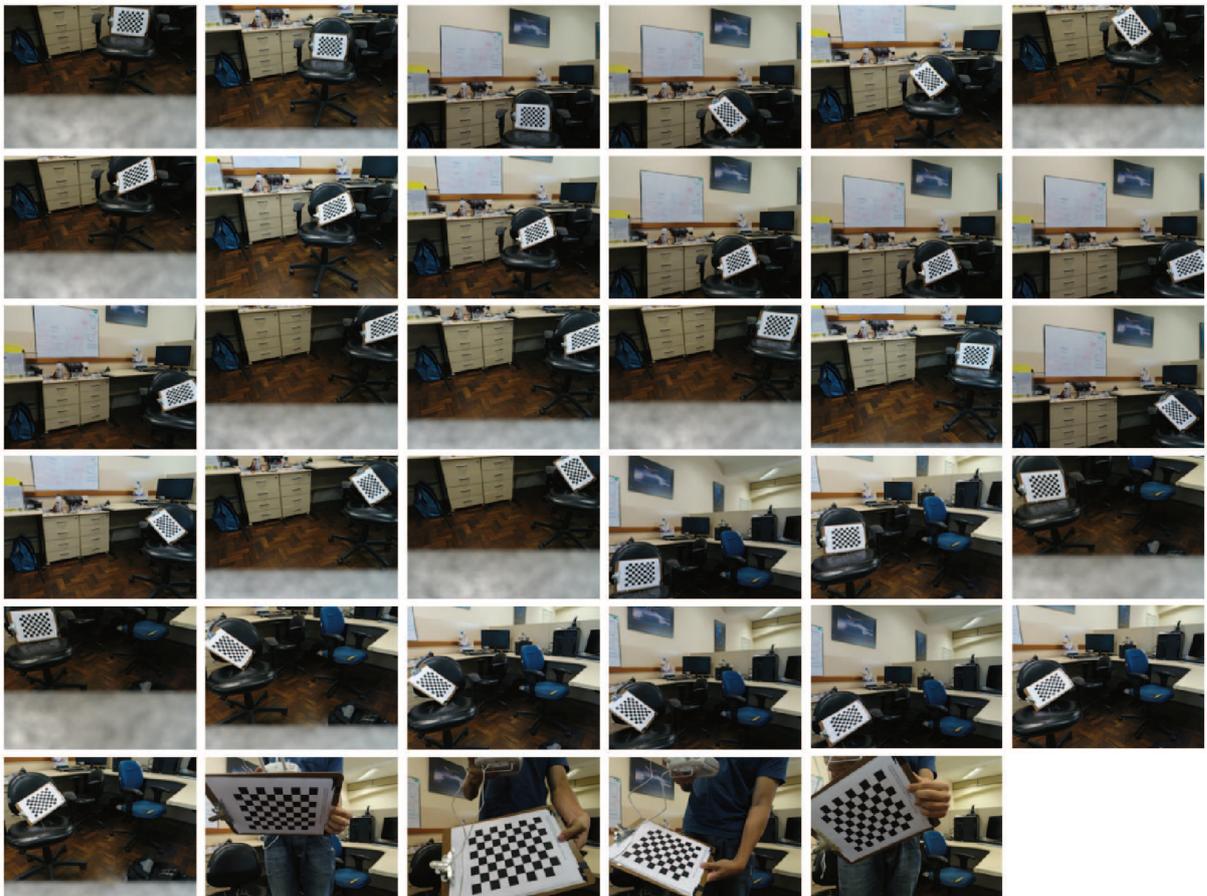
Nesse capítulo serão apresentados resultados obtidos durante o desenvolvimento do algoritmo, demonstrando as evoluções de cada etapa.

O primeiro resultado, preliminar, envolve a calibração da câmera RGB da Seção 4.1. De posse da calibração é possível calcular a odometria das câmeras no momento da captura das imagens demonstrada na Seção 4.2. A Seção 4.3 apresenta os resultados obtidos durante a estimação da nuvem densa. Por fim, a Seção 4.4 faz uma comparação da nuvem obtida com as medidas obtidas no Google Maps.

4.1 CALIBRAÇÃO DA CÂMERA

O primeiro resultado apresenta a busca pela calibração dos parâmetros da câmera, seguindo os procedimentos abordados na Seção 3.1. Para tal, foi realizado a captura de diferentes vistas de um tabuleiro impresso, semelhante à Figura 31. A Figura 35 demonstra as imagens obtidas para a calibração.

Figura 35 – Imagens utilizadas para a calibração da câmera.



Fonte: Elaborado pelo autor.

Os parâmetros obtidos durante a calibração podem ser observados na Tabela 3.

Tabela 3 – Parâmetros intrínsecos da câmera utilizada.

f_x	f_y	P_x	P_y
3696,392	3685,564	2447,985	1797,030

Fonte: Elaborado pelo autor.

Como todas imagens são obtidas da câmera do Phantom 4 PRO, esses parâmetros serão os mesmos para todas as imagens obtidas na missão.

4.2 CÁLCULO DA ODOMETRIA

Com posse dos parâmetros intrínsecos da câmera, foi realizado uma missão com o Phantom 4 sobre a Universidade Federal de Juiz de Fora (UFJF), na área da engenharia, obtendo as imagens da Figura 36.

Figura 36 – Imagens obtidas pelo Phantom 4 em uma missão sobre a engenharia da UFJF.



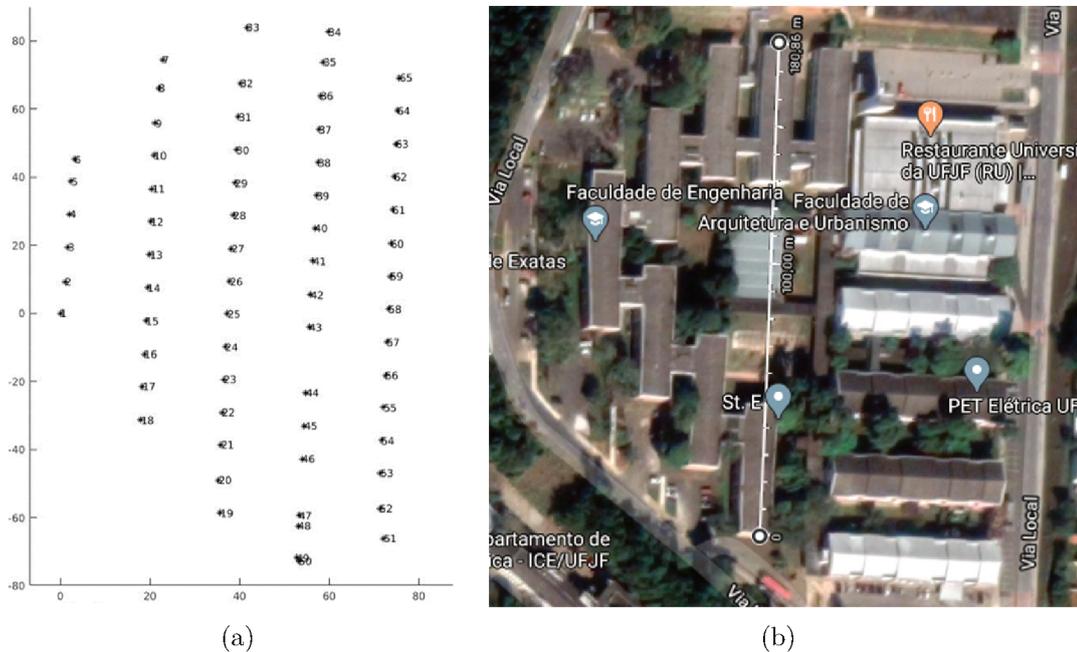
Fonte: Elaborado pelo autor.

Utilizando as técnicas descritas na Seção 3.2, foi realizado o cálculo da matriz de Pose para cada imagem obtida. A Tabela 12, apresentada no Anexo A, demonstra os parâmetros extrínsecos obtidos.

Como proposto nesse trabalho, foi realizado uma nova etapa do BA para o cálculo do ganho de correção da escala, utilizando as coordenadas do GPS e os centros óticos das imagens. Para esse procedimento, obteve-se um ganho de 9,8415. A Figura 38 (a)

demonstra as odometrias calculadas e ajustadas pelo ganho, e a Figura 38 (b) uma imagem obtida no Google Maps da área sobrevoada.

Figura 37 – Representação 2D da área mapeada.



Fonte: Elaborado pelo autor. a) Posições estimadas das câmeras. b) Área sobrevoada.

4.3 NUVEM DE PONTOS

O processo de propagação dos pontos foi trabalhado de diferentes formas durante a elaboração do trabalho. Essa seção, distribuída em 6 cenários, ilustra como ocorreu a evolução da propagação para alcançar o produto final.

Para todos os casos, com exceção do cenário 1 e 6, o processo de propagação foi realizado 3 vezes consecutivas. A cada execução, o valor de similaridade mínimo é decrescido de um valor correspondente à 0,05, partindo de um valor inicial de 0,7. Para o cálculo do ZNCC, foi utilizado um *Template* de dimensão 95×55 .

4.3.1 Cenário 1

Como primeiro cenário, é apresentado uma reconstrução de uma estátua localizada aos centros das imagens, demonstrada na Figura 38.

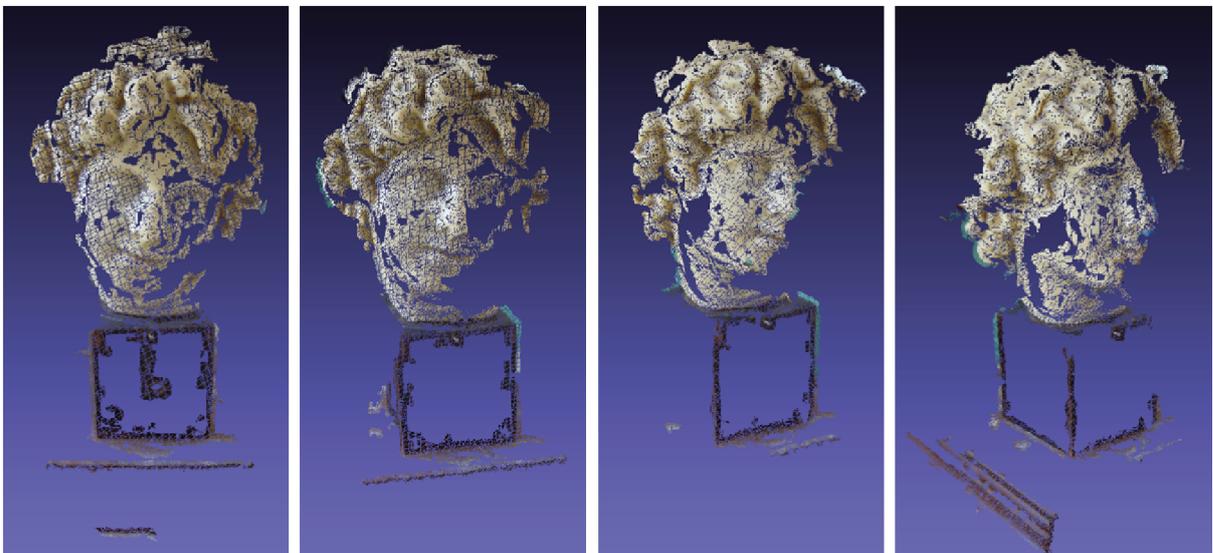
Figura 38 – Imagens utilizadas no primeiro cenário.



Fonte: Elaborado pelo autor.

Para essa reconstrução, foi realizado a propagação em pares. Seguindo a primeira metodologia apresentada na Seção 3.3, esse processo consistiu em expandir as *features* encontradas, sem realizar a transformada *affine*. A Figura 39 demonstra os pares encontrados nesse processo, onde foi adotado um *Template* de dimensão $\mathbb{R}^{7 \times 7}$ para a expansão dos pontos, executado apenas 1 vez.

Figura 39 – Pares de imagens reconstruídos do primeiro cenário.



Fonte: Elaborado pelo autor.

Em cada par de imagem obtido no processo de reconstrução é possível identificar o formato da estátua de diferentes regiões. O número de pontos identificados em cada par é demonstrado na Tabela 4.

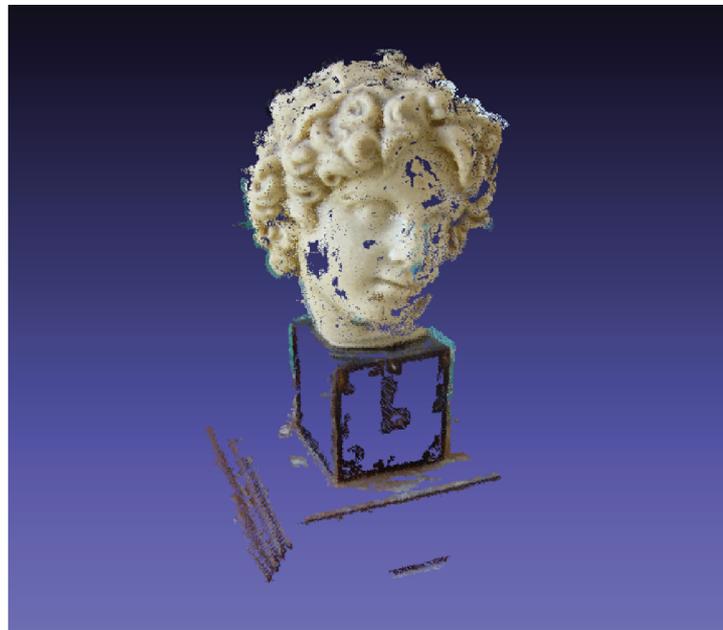
Tabela 4 – Número de pontos obtidos na reconstrução para o primeiro cenário.

Par	Número de pontos
1	58390
2	48016
3	48853
4	48589
Total	203848

Fonte: Elaborado pelo autor.

Os pares identificados na Figura 39 são complementares, e sua junção pode ser observada na Figura 40.

Figura 40 – Nuvem de pontos sobrepostas do primeiro cenário.



Fonte: Elaborado pelo autor.

Apesar de não preencher a base da estátua, a reconstrução obtida conseguiu reconstruir grande parte de sua cabeça. Como as imagens são obtidas de ângulos diferentes e a base possui planos bem definidos, uma similaridade baixa pode ser apresentada ao realizar o cálculo do ZNCC sem a utilização da transformada *affine*.

Essa metodologia foi empregada para as imagens obtidas da UFJF, porém não foi obtido êxito na reconstrução. Um dos principais problemas se deu devido a mudança de rotação das imagens. Na missão o drone sobrevoa a área em uma direção, realiza um giro de 180° e segue na direção oposta. O algoritmo não é capaz de perceber essa rotação, ou seja, assume que todas imagens possuem a mesma rotação e falha ao realiza a busca por novos vizinhos.

4.3.2 Cenário 2

Para um segundo cenário foi adotado uma metodologia similar ao cenário 1, com o acréscimo da transformada *affine*, além do refinamento proposto por (YLIMÄKI et al., 2015). Utilizou-se um conjunto de imagens do campus de engenharia da UFJF, de dimensões 2000x1500, apresentada na Figura 41.

Figura 41 – Imagens retiradas do campo de engenharia da UFJF.



Fonte: Elaborado pelo autor.

Foi adotado um *Template* de dimensão $\mathfrak{R}^{7 \times 7}$ para a expansão dos pontos, onde após aplicar a técnica proposta para esse cenário, tem-se os pares da Figura 42.

Figura 42 – Pares de imagens reconstruídos do segundo cenário.



Fonte: Elaborado pelo autor.

Os telhados das unidades foram detectados pela propagação, sendo propagados por praticamente toda sua extensão. Isso ocorreu devido ao grande número de *features* distinguíveis presentes nessa região. Para o solo, o número de pontos identificados foi inferior aos resultados das demais seções. A Tabela 5 apresenta o número de pontos identificados em cada par de imagens.

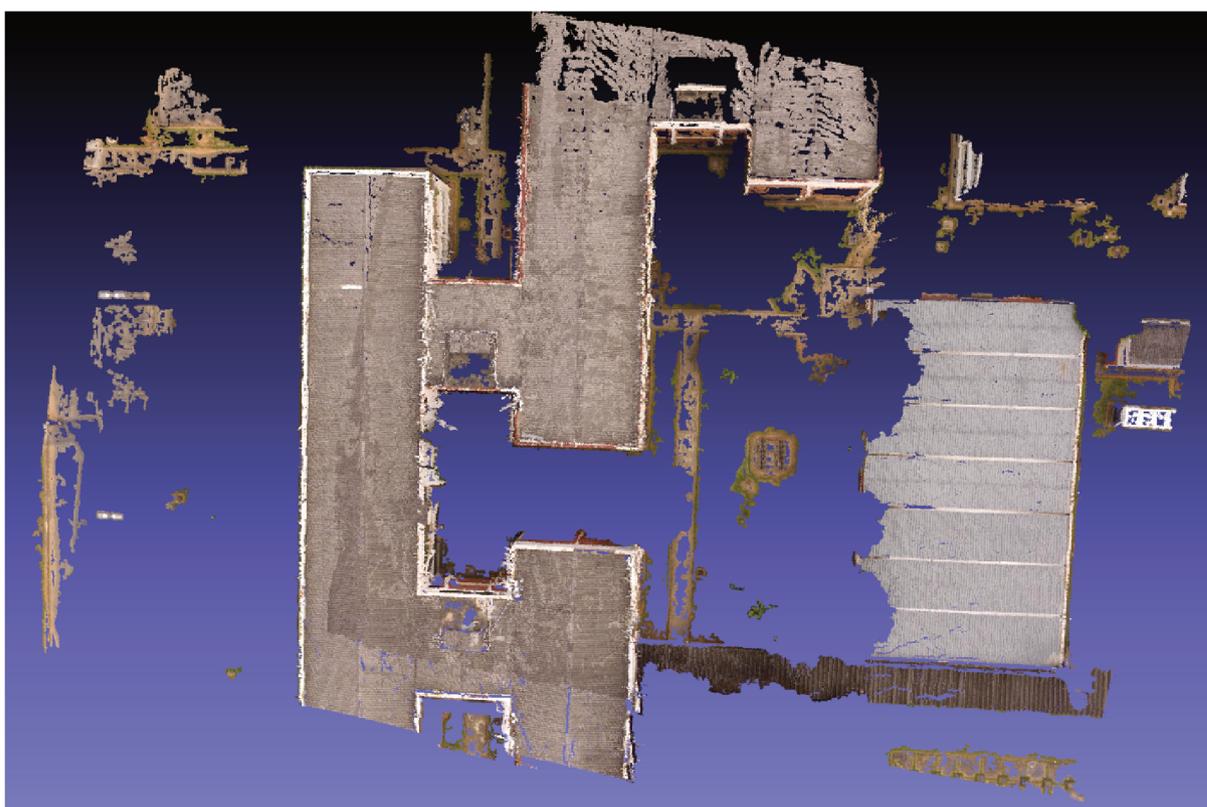
Tabela 5 – Número de pontos obtidos na reconstrução para o segundo cenário.

Par	Número de pontos
1	776933
2	796362
3	470069
4	950690
5	1131782
Total	4125836

Fonte: Elaborado pelo autor.

A Figura 43 apresenta os pares reconstruídos com superposição.

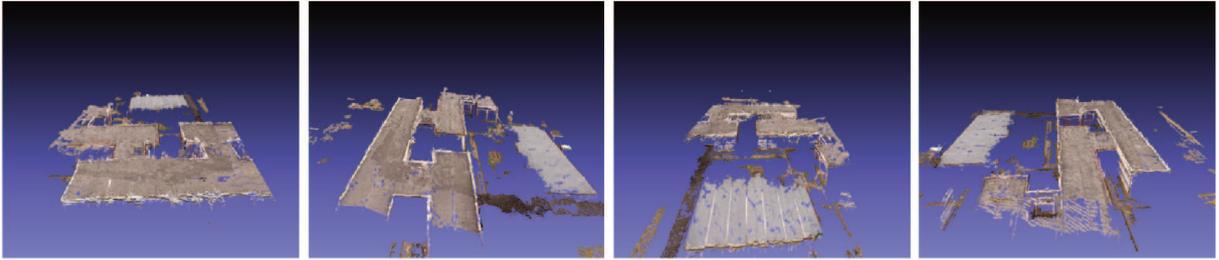
Figura 43 – Nuvem de pontos sobrepostas do segundo cenário.



Fonte: Elaborado pelo autor.

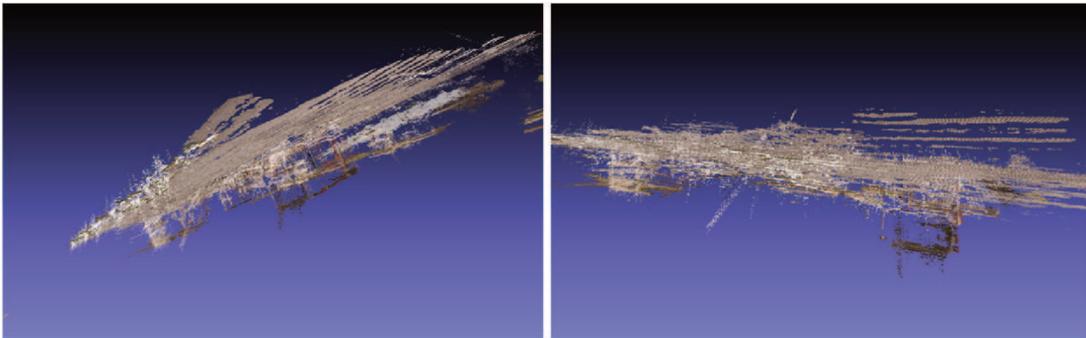
Em uma vista superior, observa-se que os pontos se encaixam e aparentam ser apenas uma nuvem de pontos. As Figuras 44 e 45 apresentam vistas da reconstrução de diferentes orientações.

Figura 44 – Nuvem de pontos sobrepostas do segundo cenário de diferentes vistas.



Fonte: Elaborado pelo autor.

Figura 45 – Nuvem de pontos sobrepostas do segundo cenário.



Fonte: Elaborado pelo autor.

A Figura 45 apresenta um excesso de planos para uma mesma região, ocasionado pela sobreposição dos mesmos. Além de apresentar ruídos na reconstrução, essa sobreposição obtém uma nuvem de pontos carregada, sem muita informação, visto que um conjunto de pontos 3D obtidos representa um ponto do mundo real.

4.3.3 Cenário 3

Para o terceiro cenário, foi selecionado um subconjunto de imagens obtidas na missão do Phantom 4 (Figura 36). Esse subconjunto pode ser visto na Figura 46.

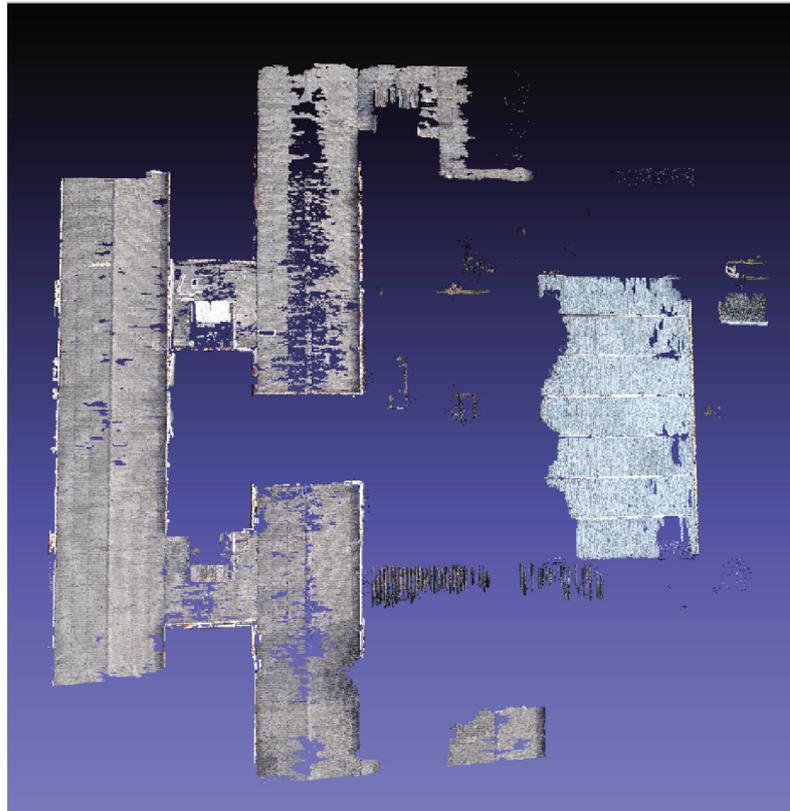
Figura 46 – Subconjunto das imagens obtidas na missão do Phantom 4.



Fonte: Elaborado pelo autor.

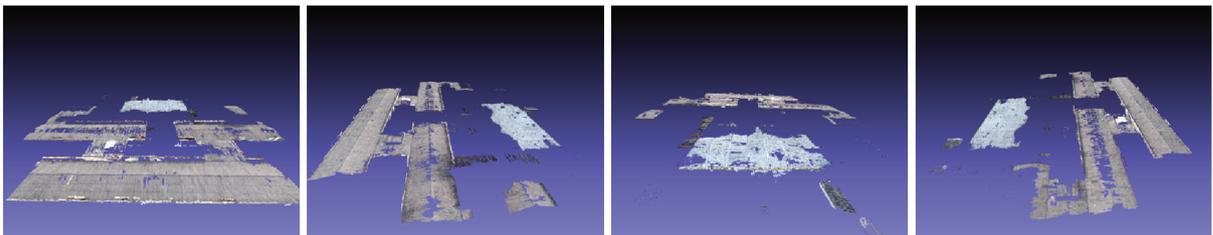
Nesse cenário foi aplicado a técnica de propagação apresentado por (YLIMÄKI et al., 2015), com o intuito de obter apenas uma nuvem para todas as imagens. Com um *Template* de dimensão $\mathbb{R}^{7 \times 7}$, somado ao *Template* do ZNCC, tem-se assim dimensão de $\mathbb{R}^{9 \times 9}$. As Figura 47 e 48 mostram a reconstrução obtida.

Figura 47 – Nuvem de pontos obtida no terceiro cenário.



Fonte: Elaborado pelo autor.

Figura 48 – Nuvem de pontos do terceiro cenário de diferentes vistas.



Fonte: Elaborado pelo autor.

Esse cenário apresentou uma área de reconstrução apenas os telhados da unidade, considerada baixa. Nesse processo, foi determinado que um ponto que deveria pertencer a mais de duas imagens para ser adicionado a nuvem. Além disso, esse procedimento apresentou um grande esforço computacional, devido à obtenção do gradiente da imagem, ocupando um grande espaço na memória. Ainda nessa área, esse método obtém um

número de pontos muito elevado, que não traduz em um grande mapeamento da área. A Tabela 6 apresenta a quantidade de memória e o número de pontos detectados nessa reconstrução.

Tabela 6 – Número de pontos detectados e uso de memória no terceiro cenário.

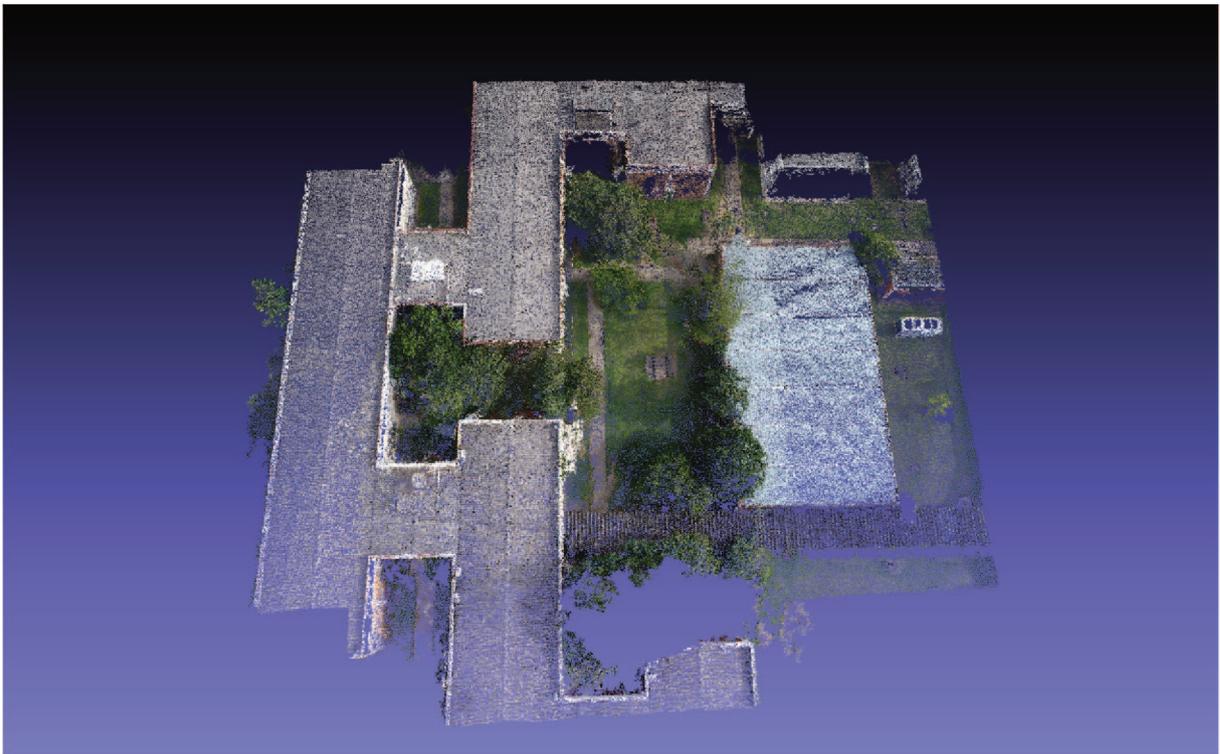
Número de pontos	Consumo de memória
7020001	15,4 Gb

Fonte: Elaborado pelo autor.

4.3.4 Cenário 4

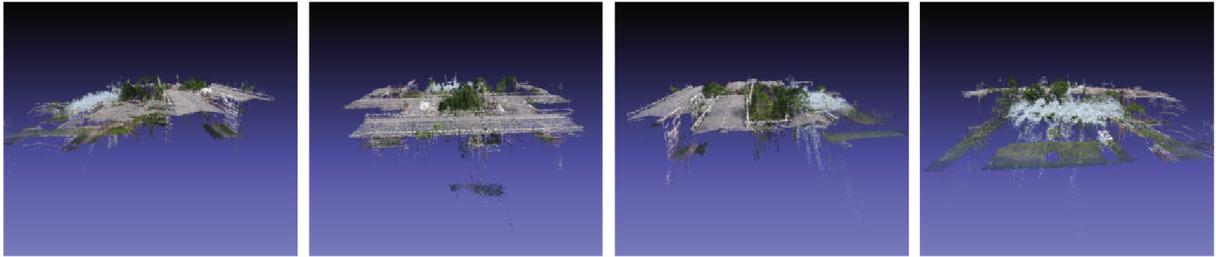
Para o quarto cenário, adotou-se o processo de propagação apresentado por (FURUKAWA; HERNÁNDEZ, 2015). Foi adotado um *grid* de dimensão $\mathbb{R}^{4 \times 4}$, sendo que cada quadrado do *grid* possui apenas 1 ponto 3D. Nesse caso utilizou-se o mesmo conjunto de imagens do cenário anterior e obteve-se a reconstrução apresentada na Figura 49 e 50.

Figura 49 – Nuvem de pontos obtida no quarto cenário.



Fonte: Elaborado pelo autor.

Figura 50 – Nuvem de pontos do quarto cenário de diferentes vistas.



Fonte: Elaborado pelo autor.

É possível perceber que esse método mapeou uma área maior, se comparado ao cenário 3, captando tanto o solo, quanto algumas árvores que existiam na cena, além do telhado das unidades. Vale destacar que ao mesmo passo que consegue cobrir uma área maior na reconstrução, adquire um número de pontos 3D menor como mostrado na Tabela 7.

Tabela 7 – Número de pontos detectados e uso de memória no quarto cenário.

Número de pontos	Consumo de memória
650001	5,2 Gb

Fonte: Elaborado pelo autor.

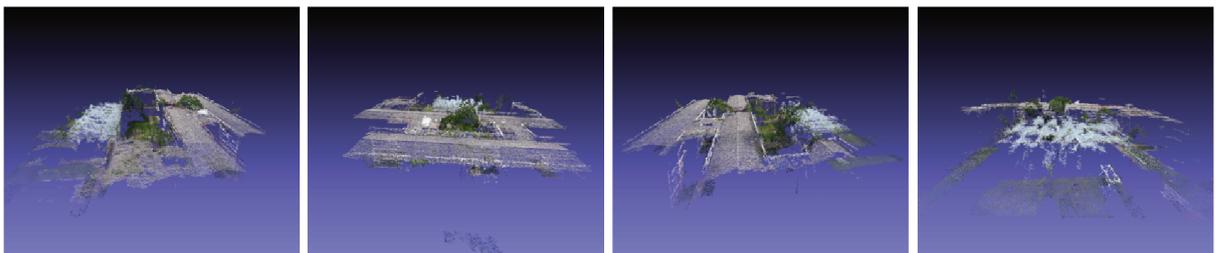
Sendo assim, esse algoritmo obteve uma nuvem de pontos mais leve e expressiva, se comparado ao terceiro cenário, além de apresentar uma grande redução no gasto de memória durante o processo. Entretanto, essa metodologia apresenta muitos pontos espalhados, principalmente embaixo do telhado, como pode ser visto na Figura 50. A fim de remover esses *outliers*, os filtros abordados na Seção 2.4.5 foram implementados, obtendo o resultado mostrado nas Figuras 51 e 52.

Figura 51 – Nuvem de pontos após a aplicação do filtro.



Fonte: Elaborado pelo autor.

Figura 52 – Nuvem de pontos filtrada de diferentes vistas.



Fonte: Elaborado pelo autor.

Como esperado, esse filtro reduz o número de pontos da nuvem, removendo os *outlier* e acaba aumentando o gasto de memória, como demonstrado na Tabela 8.

Tabela 8 – Número de pontos detectados e uso de memória com a aplicação do filtro.

Número de pontos	Consumo de memória
467172	5,6 Gb

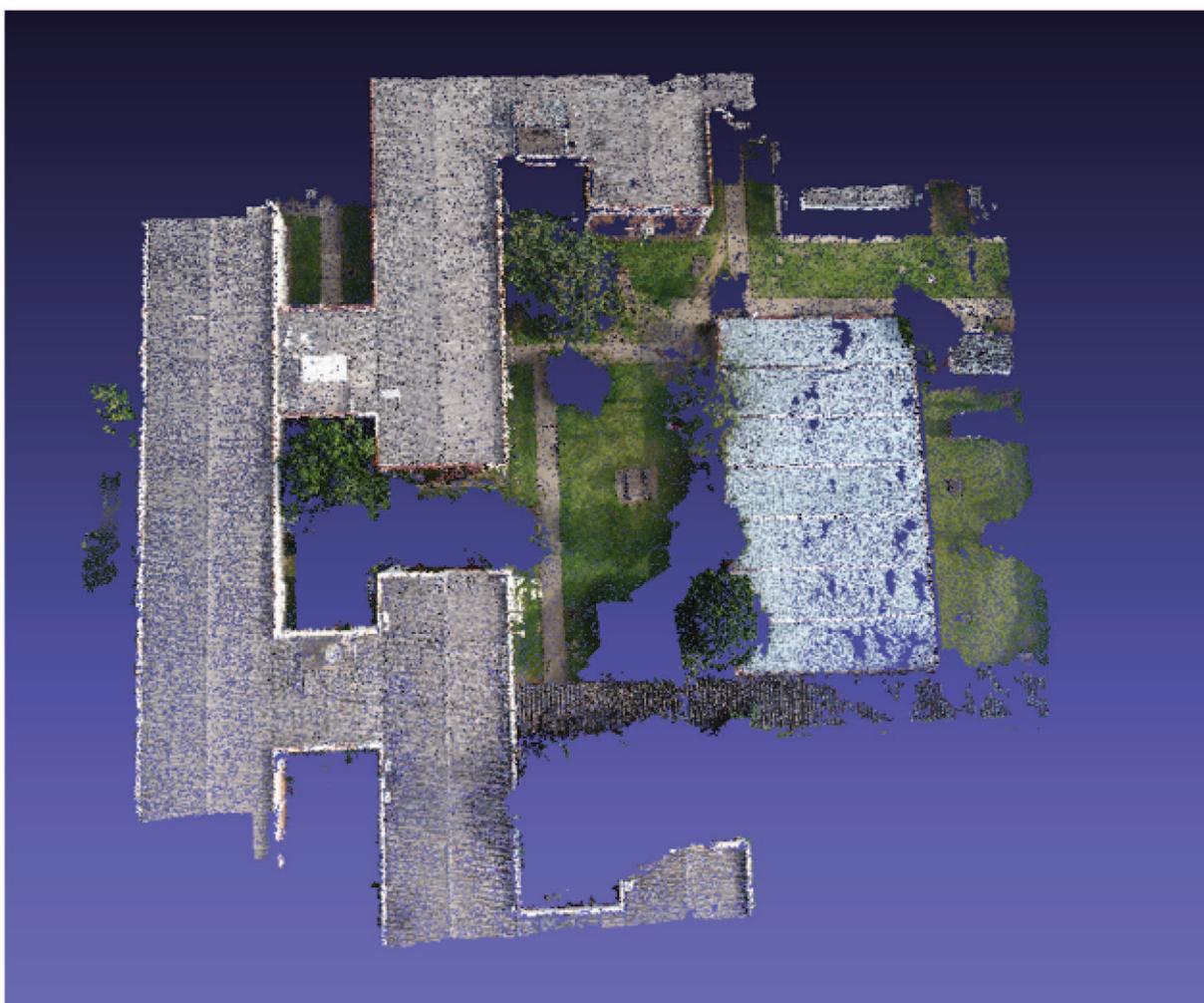
Fonte: Elaborado pelo autor.

É possível observar que alguns pontos espalhados foram removidos, entretanto, os pontos sobre o telhado branco apresentaram um comportamento bastante ruidoso. Isso se dá pelos pontos serem pouco distinguíveis entre si, fazendo com que pontos de diferentes regiões apresentem uma similaridade alta.

4.3.5 Cenário 5

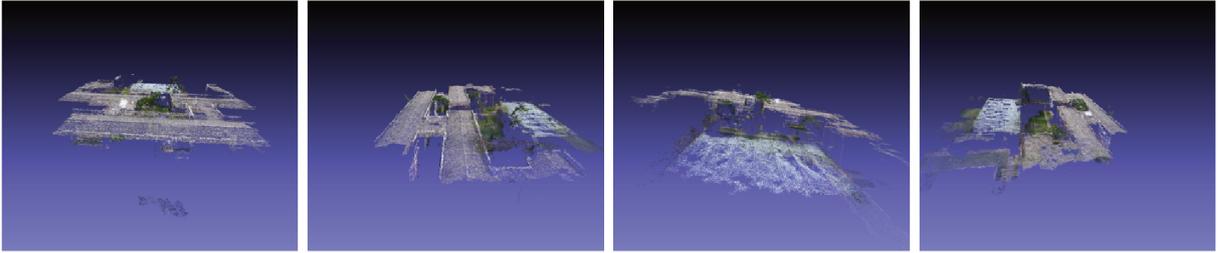
Para o quinto cenário, foi adicionado uma adaptação no método apresentado no cenário anterior, proposta neste trabalho, determinando o limite de propagação dos pontos, a fim de minimizar a quantidade de pontos de áreas distintas sejam detectados como *match*. A adaptação adotada está descrita na Seção 3.3. Os parâmetros de propagação formam mantidos do cenário 4, como o acréscimo do limite de propagação de 7 *pixels*. As Figura 53 e 54 apresentam o comportamento dessa metodologia.

Figura 53 – Nuvem de pontos do quinto cenário.



Fonte: Elaborado pelo autor.

Figura 54 – Nuvem de pontos do quinto cenário de diferentes vistas.



Fonte: Elaborado pelo autor.

É possível perceber que o algoritmo apresenta um melhor desempenho para o telhado branco, entretanto, outras regiões do mapa foram perdidas, como as árvores da cena. Uma justificativa para não detecção das árvores é devido aos pontos iniciais não possuírem pontos dessas estruturas. Como o algoritmo limita a expansão para área conhecida, esses pontos não são detectados. A Tabela 9 mostra o número de pontos encontrados e o consumo de memória.

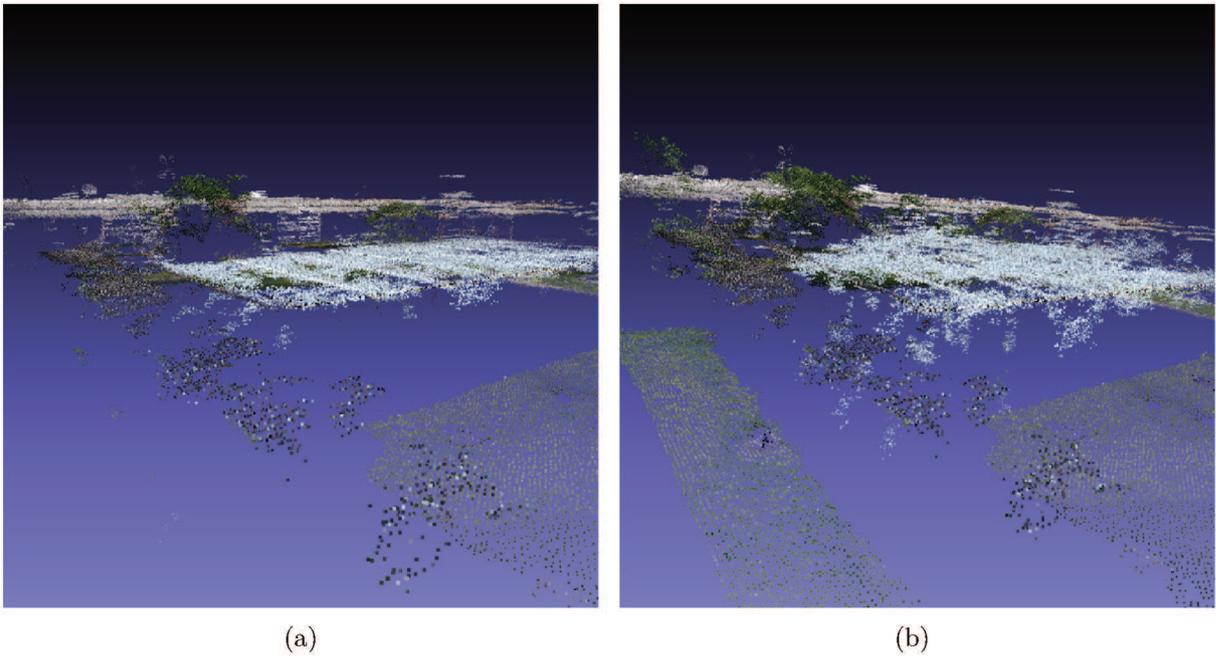
Tabela 9 – Número de pontos detectados e uso de memória no quinto cenário.

Número de pontos	Consumo de memória
282525	5,7 Gb

Fonte: Elaborado pelo autor.

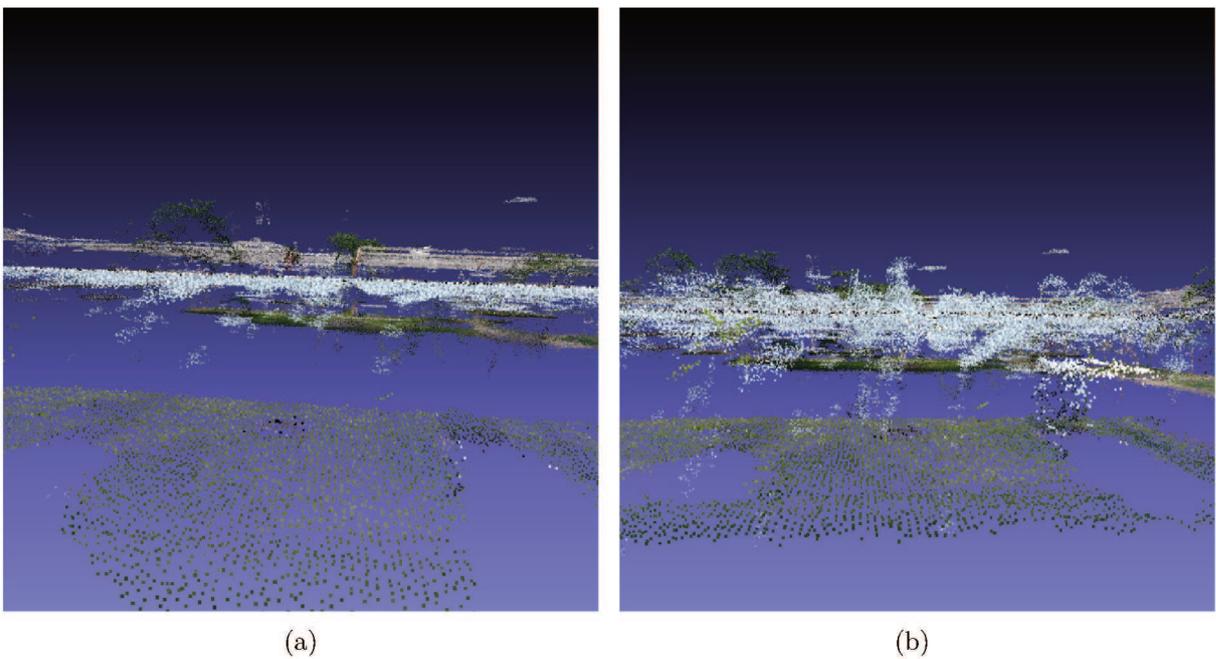
O número de pontos reduziu comparado ao cenário anterior, visto que menos áreas foram preenchidas, entretanto teve um pequeno acréscimo no consumo de memória. Esse aumento ocorre, devido ao cálculo dos limites de propagação. As Figuras 55 e 56 mostram a comparação entre o algoritmo utilizado no quinto cenário (a) e no quarto cenário (b). Essas figuras confirmam o melhor desempenho do algoritmo do cenário 5.

Figura 55 – Comparação entre o desempenho dos algoritmos observados da mesma posição.



Fonte: Elaborado pelo autor. a) Resultado do algoritmo do quinto cenário. b) Resultado do algoritmo do quarto cenário.

Figura 56 – Uma segunda vista comparando o desempenho entre os algoritmos.



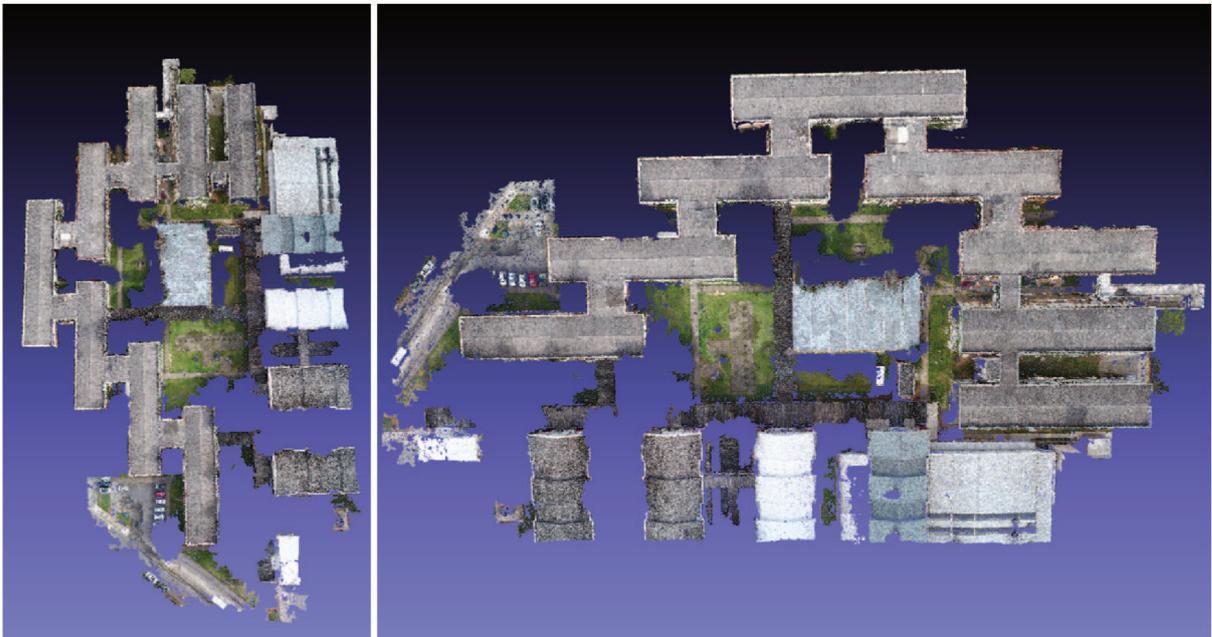
Fonte: Elaborado pelo autor. a) Resultado do algoritmo do quinto cenário. b) Resultado do algoritmo do quarto cenário.

Ao analisar as Figuras 55 e 56 é possível perceber um acréscimo de performance da metodologia proposta no cenário 5 em relação ao cenário 4, com o retorno de pouca ondulação sobre o telhado branco. É válido ressaltar que esse telhado possui padrões semelhantes em toda sua extensão, o que dificulta a expansão de pontos nessa região.

4.3.6 Cenário 6

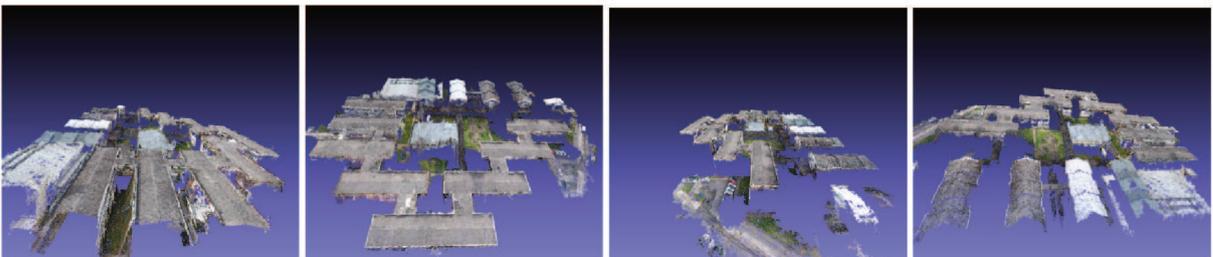
Por fim foi realizado uma reconstrução utilizando todas as imagens apresentadas na Figura 36. O algoritmo do Cenário 5 foi selecionado e seus valores do parâmetros de configuração foram mantidos. As Figuras 57 e 58 apresentam o desempenho o algoritmo.

Figura 57 – Reconstrução utilizando todas imagens obtida pela missão do Phantom 4.



Fonte: Elaborado pelo autor.

Figura 58 – Diferentes vistas da nuvem obtida na reconstrução com todas imagens.



Fonte: Elaborado pelo autor.

Uma grande área do campus de engenharia da UFJF foi detectada pela reconstrução, apresentando de forma expressiva a extensão reprojeta. Apesar do grande desempenho,

as árvores não foram reconstruídas, deixando muitas áreas vazias na reconstrução. Para essa reconstrução, o processo de expansão foi executado apenas 1 vez, o que pode ter influenciado em seu desempenho. Foi escolhido apenas uma iteração, devido ao tempo necessário para processar todas as imagens ser elevado.

A Tabela 10 demonstra o número de pontos encontrados e o consumo de memória para esse cenário.

Tabela 10 – Número de pontos detectados e uso de memória no sexto cenário.

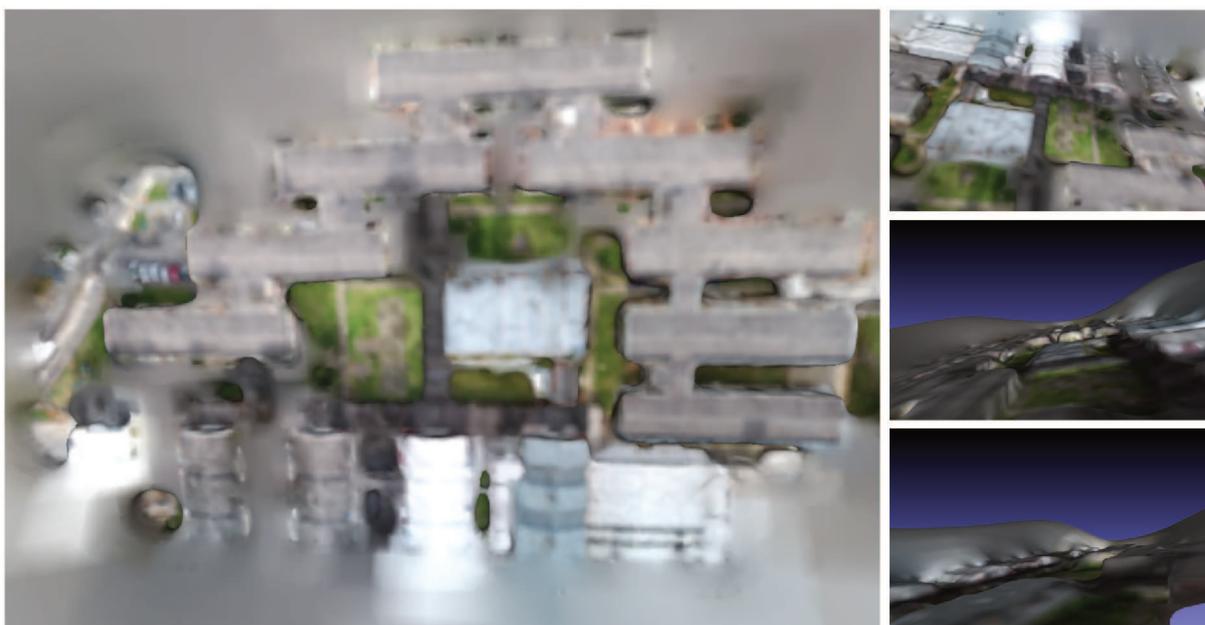
Número de pontos	Consumo de memória
2365988	13,3 Gb

Fonte: Elaborado pelo autor.

Para o cenário 6, houve um grande número de pontos encontrados, isso se deu devido ao tamanho da área reconstruída. Apesar disso, o terceiro cenário apresentou uma quantia maior que o sexto cenário, o que comprova que essa metodologia apresenta pontos mais representativos da cena.

Utilizando o MeshLab, com implementação da técnica de *Poisson* sobre a nuvem, obteve-se estruturas mais sólidas da reconstrução. Seu resultado pode ser observado na Figura 59.

Figura 59 – Reconstrução após a aplicação da técnica de *Poisson*.



Fonte: Elaborado pelo autor.

A técnica de *Poisson* demonstra o bom desempenho do algoritmo para o cálculo da profundidade dos prédios. Os edifícios podem ser facilmente identificados nessa técnica,

lembrando que um próximo passo para essa metodologia seria o processo de texturização que não foi realizado nesse trabalho. Com a texturização, pequenos pedaços das imagens são inseridos ao resultado do *Poisson*, apresentando cenários com nível de detalhamento elevado.

4.4 Comparação entre Medidas do Google Maps e a Nuvem Calculada

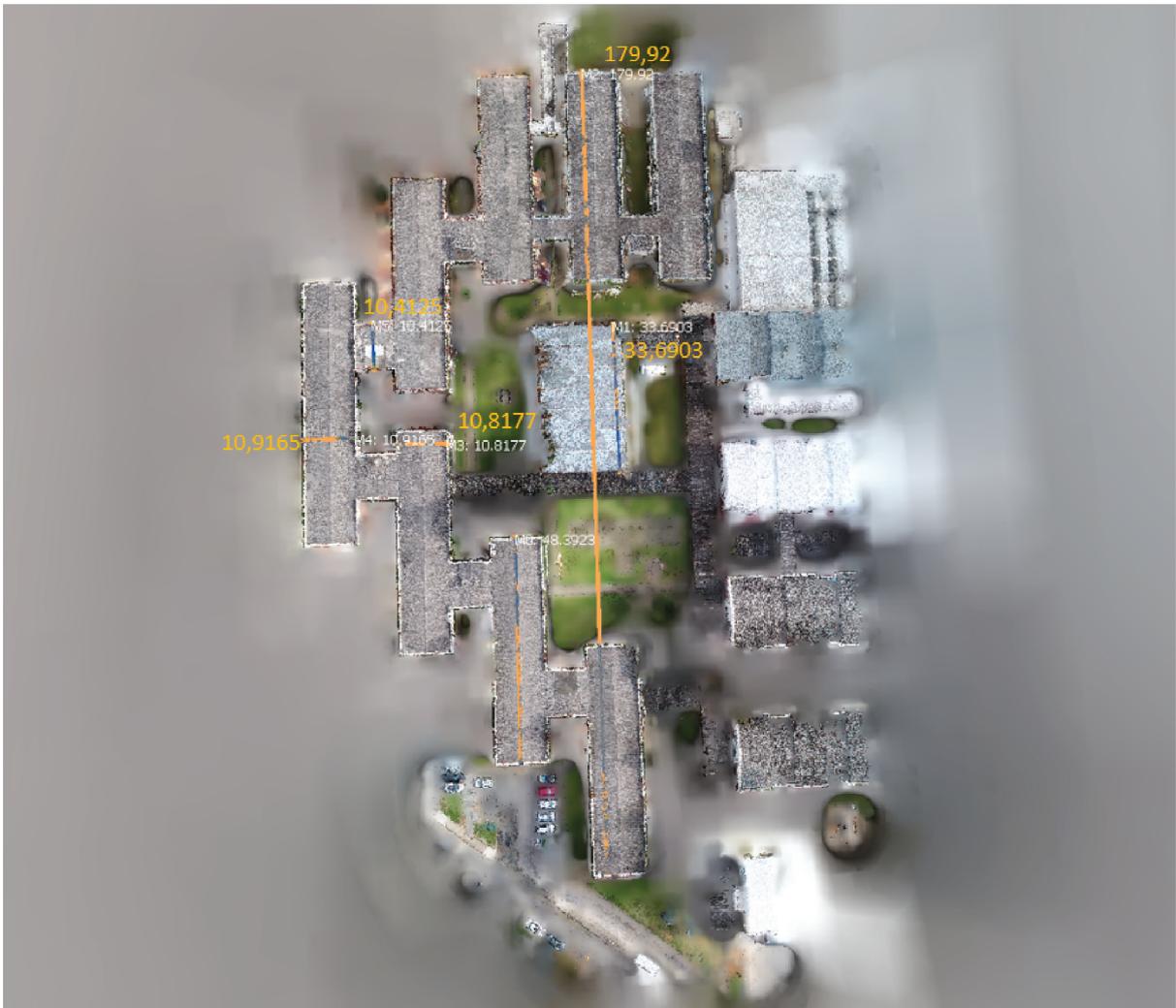
Como última análise dos resultados, foi realizado uma comparação entre imagens retiradas do Google Maps (Figura 60) e medidas da cena reconstruída (Figura 61). Foi utilizado a ferramenta para medir distâncias do Google Maps, para determinar a distância entre pontos específicos. Essa mesma medição foi realizada na nuvem obtida com o auxílio do MeshLab.

Figura 60 – Imagens obtidas no Google Maps com a medida do comprimento entre pontos da cena em amarelo.



Fonte: Elaborado pelo autor.

Figura 61 – Medidas realizadas sobre a nuvem de pontos obtidas.



Fonte: Elaborado pelo autor.

A Tabela 11 mostra os valores obtidos em cada medição.

Tabela 11 – Comparação entre as mediadas obtidas no Google Maps e os valores estimados na reconstrução.

Reta	Google Maps(m)	Nuvem de Pontos(m)	Erro(m)	Porcentagem do erro(%)
1	180,86	179,9200	0,94	0,5197
2	33,55	33,6903	0,1403	0,4182
3	48,12	48,3923	0,2723	0,5659
4	10,73	10,8177	0,0877	0,8173
5	10,84	10,9165	0,0765	0,7130
6	10,25	10,4125	0,1625	1,5853

Fonte: Elaborado pelo autor.

Com um erro máximo menor que 2%, as medidas se mostraram satisfatórias. É válido ressaltar que os pontos são selecionados manualmente, o que diminui a precisão das medidas selecionadas. Mas em um aspecto geral, a escala obtida da cena pode ser considerada aplicável.

5 CONCLUSÕES

Este trabalho apresentou uma metodologia de reconstrução de um ambiente real em 3D utilizando técnicas de visão computacional, capaz de retirar informações de imagens 2D. Com o resultado é possível obter dados de reconstrução de uma área extensa, sobrevoada por um UAV.

Ao longo do trabalho as técnicas foram detalhadas para a obtenção de um modelo final a partir do registro de nuvens de pontos. A metodologia SFM em conjunto com um método de ordenação de pares é apresentada com a finalidade de obter a localização das câmeras para um conjunto de imagens desordenados. Como essa metodologia não apresenta uma escala real, foram utilizados os dados de coordenadas do GPS para corrigi-las.

Para a obtenção da nuvem de pontos, foram apresentadas três técnicas de MVS, que ao serem comparadas, obtiveram diferentes desempenhos quanto ao número de pontos 3D identificados, extensão das áreas expandidas e consumo de memória. Nessa etapa o algoritmo apresentou dificuldades para a reconstrução de áreas das imagens que apresentam baixa variação de cor. O Cenário 5, que consistiu em uma propagação no espaço tridimensional com o cálculo de seus limites de propagação, o que ocasionou em um melhor desempenho diante deste problema, entretanto apresentou menor área reconstruída se comparado ao Cenário 4, onde não era realizado o cálculo do limite de propagação.

Por fim, foi realizado uma reconstrução utilizando grande quantidade de imagens obtidas em uma missão de voo do UAV (65 imagens), resultando em uma reconstrução satisfatória, com medidas dos edifícios do cenário similares aos encontrados no Google Maps, apresentando um erro máximo de 1,58% entre as aferições.

5.1 TRABALHOS FUTUROS

Trabalhos futuros são vislumbrados frente aos resultados apresentados, principalmente no que diz respeito à qualidade do modelo final em alguns aspectos.

Devem ser realizados estudos para o ajuste de parâmetros do algoritmo, com o intuito de identificar uma melhor configuração para obtenção de uma propagação que atinja mais áreas.

A fim de obter aferições mais precisas, é vislumbrado a utilização de pontos de controle durante o experimento, para obter o georreferenciamento preciso dos dados coletado.

Ainda no processo de propagação da nuvem de pontos, apesar dos pontos serem organizados em fila de prioridade e seus vizinhos serem analisados durante a execução para evitar a duplicação de dados, é possível paralelizar o algoritmo, reduzindo o tempo consumido para a expansão.

Para a correção de ruídos apresentados nas nuvens de pontos, a técnica do *Deep Learning* pode ser aplicada, com o intuito de reconhecer formatos e reorganizar os pontos nos formatos reconhecidos.

Por fim, uma técnica de clusterização de imagens é almejada, visto que o número de imagens utilizados durante o processo implica diretamente no consumo de memória. Com a clusterização das imagens, cada conjunto pode ser analisado separadamente e ao final suas nuvens podem ser sobrepostas. Vale ressaltar que os conjuntos devem ter sobreposição entre si, possibilitando a utilização de uma câmera de referência para o processo.

REFÊRENCIA

- ALBUQUERQUE, E. S. **Desenvolvimento de um módulo para template matching baseado em ZNCC com prototipação em FPGA**. 2017. Dissertação. (Mestrado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife, 2017.
- ALCANTARILLA, P. F.; BARTOLI, A.; DAVISON, A. J. KAZE features. In: FITZGIBBON, A. et al. **12th European Conference on Computer Vision**. Florence, Itália: Springer, out. 2012. p. 214-227.
- ALCANTARILLA, P. F.; SOLUTIONS, T. Fast explicit diffusion for accelerated features in nonlinear scale spaces. **IEEE Trans. Patt. Anal. Mach. Intell**, Atlanta, Estados Unidos, v. 34, n. 7, p. 1281–1298, set. 2013.
- ALIDOOST, F; AREFI, H. An image-based technique for 3D building reconstruction using multi-view UAV images. **The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences**, Kish, Irã, v. 40, n. 1, p. 43, nov. 2015.
- ATKINSON, K. B. **Close range photogrammetry and machine vision**. Escócia: Whittles Publishing, 1996.
- BADURA, G. P. et al. A novel approach for deriving LAI of salt marsh vegetation using structure from motion and multiangular spectra. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 12, n. 2, p. 599–613, mar. 2019.
- BARATH, D.; HAJDER, L. Novel ways to estimate homography from local affine transformations. **Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications**, v. 11, n. 3, p. 434–445, fev. 2016.
- BAY, H. et al. Speeded-up robust features (SURF). **Computer vision and image understanding**, v. 110, n. 3, p. 346–359, jun. 2008.
- BEDRUZ, R. A. R. et al. Vehicle Classification Using AKAZE and Feature Matching Approach and Artificial Neural Network. In: **10TH IEEE Conference TENCON**. Jeju, Coreia do Sul, out. 2018. P. 1824–1827.
- BEVILACQUA, A.; GHERARDI, A.; CAROZZA, L. Automatic perspective camera calibration based on an incomplete set of chessboard markers. In: **SIXTH IEEE Indian**

Conference on Computer Vision, Graphics & Image Processing. Bhubaneswar, Índia, dez. 2008. P. 126–133.

BROWN, M.; LOWE, D. G. Unsupervised 3D object recognition and reconstruction in unordered datasets. In: **FIFTH IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM'05).** Ottawa, Canadá, jun. 2005. v. 5. P. 56–63.

CAMPOS, S. R. Os cinco sentidos da hospitalidade. **Revista Acadêmica Observatório de Inovação do Turismo**, Rio de Janeiro, v. 3, n. 1, p. 9–10, mar. 2008.

CARRIVICK, J. L.; SMITH, M. W. Fluvial and aquatic applications of Structure from Motion photogrammetry and unmanned aerial vehicle/drone technology. **Wiley Interdisciplinary Reviews: Water**, v. 6, n. 1, p. 1328–1345, jan. 2019.

CARVAJAL-RAMIREZ, F. et al. Virtual reconstruction of damaged archaeological sites based on Unmanned Aerial Vehicle Photogrammetry and 3D modelling. Study case of a southeastern Iberia production area in the Bronze Age. **Measurement**, v. 136, p. 225–236, mar. 2019.

CECH, Jan; SARA, Radim. Efficient sampling of disparity space for fast and accurate matching. In: **IEEE Conference on Computer Vision and Pattern Recognition.** Jun. 2007. P. 1–8.

CLUB, VIRTUAL KENNEL. **Color and Tint.** 2020. Disponível em: <https://www.virtualkennelclub.com/page.php?id=8>. Acessado em 20 jan. 2020.

CORKE, P. **Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised.** Berlim, Alemanha: Springer International Publishing, 2017.

COUTO, L. N. **Sistema para localização robótica de veículos autônomos baseado em visão computacional por pontos de referência.** 2012. Tese. (Doutorado em Ciências de Computação e Matemática Computacional) – Universidade de São Paulo, São Carlos, 2012.

DE FARIAS, T. S. M. C. **Metodologia para reconstrução 3d baseada em imagens.** 2012. Tese (Doutorado em Ciência da Computação) – Universidade Federal de Pernambuco, Pernambuco, 2012.

DEGLINT, J. et al. Auto-calibration of a projector–camera stereo system for projection mapping. **Journal of the Society for Information Display**, v. 24, n. 8, p. 510–520, ago. 2016.

DI STEFANO, L.; MATTOCCIA, S.; TOMBARI, F. ZNCC-based template matching using bounded partial correlation. **Pattern recognition letters**, v. 26, n. 14, p. 2129–2134, out. 2005.

DJI. **Phantom 4 PRO**. 2020. Disponível em:

<https://www.dji.com/br/phantom-4-pro-v2>. Acessado em 20 mar. 2020.

DUBROFSKY, E. **Homography estimation**. 2009. Dissertação. (Mestrado em Ciência da Computação) – Universidade da Colúmbia Britânica, Vancouver, Canadá, 2009.

ESCALERA, A.; ARMINGOL, J. M. Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration. **Sensors**, v. 10, n. 3, p. 2027–2044, mar. 2010.

EWERTOWSKI, M. et al. Operational framework for rapid, very-high resolution mapping of glacial geomorphology using low-cost unmanned aerial vehicles and structure-from-motion approach. **Remote Sensing**, v. 11, n. 1, p. 65–83, jan. 2019.

FAVILLA, M et al. Avaliação da visão funcional de uma criança prematura com cegueira congênita. **Cadernos de Terapia Ocupacional da UFSCar**, São Carlos, v. 22, n. 2, p. 429–434, mai. 2014.

FERRARI, V.; TUYTELAARS, T.; VAN GOOL, L. Simultaneous object recognition and segmentation by image exploration. In: PAJDLA, T.; MATAS, J. **8th European Conference on Computer Vision**. Prague, República Tcheca: Springer, mai. 2004. p. 40-54.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, v. 24, n. 6, p. 381–395, jun. 1981.

FORSMOO, J. et al. Drone-based structure-from-motion photogrammetry captures grassland sward height variability. **Journal of applied ecology**, v. 55, n. 6, p. 2587–2599, mar. 2018.

FÖRSTNER, W.; WROBEL, B. P. **Photogrammetric computer vision**. Suíça: Springer, 2016.

FRASER, B.; CONGALTON, R. Evaluating the Effectiveness of Unmanned Aerial Systems (UAS) for Collecting Thematic Map Accuracy Assessment Reference Data in New England Forests. **Forests**, v. 10, n. 1, p. 24–41, jan. 2019.

FURUKAWA, Y.; HERNÁNDEZ, C. Multi-view stereo: A tutorial. **Foundations and Trends® in Computer Graphics and Vision**, Hanôver, Estados Unidos, v. 9, n. 1-2, p. 1–148, jun. 2015.

GEORGOULAS, C.; ANDREADIS, I. FPGA based disparity map computation with vergence control. **Microprocessors and Microsystems**, v. 34, n. 7-8, p. 259–273, nov. 2010.

GOESELE, M.; CURLESS, B.; SEITZ, S. M. Multi-view stereo revisited. In: **IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)**. Nova Iorque, Estados Unidos, jun. 2006. v. 2. P. 2402–2409.

GÓMEZ-CANDÓN, D.; DE CASTRO, A. I.; LÓPEZ-GRANADOS, F. Assessing the accuracy of mosaics from unmanned aerial vehicle (UAV) imagery for precision agriculture purposes in wheat. **Precision Agriculture**, v. 15, n. 1, p. 44–56, nov. 2014.

GRAHAM, A. et al. Evaluation of ground surface models derived from unmanned aerial systems with digital aerial photogrammetry in a disturbed conifer forest. **Remote Sensing**, v. 11, n. 1, p. 84–103, jan. 2019.

GRYCUK, R.; SCHERER, R.; GABRYEL, M. New image descriptor from edge detector and blob extractor. **Journal of Applied Mathematics and Computational Mechanics**, v. 14, n. 4, p. 31–39, abr. 2015.

HABBECKE, M.; KOBELT, L. A surface-growing approach to multi-view stereo reconstruction. In: **IEEE Conference on Computer Vision and Pattern Recognition**. Mineápolis, Estados Unidos, jun. 2007. P. 1–8.

HARRIS, C. G; STEPHENS, M. A combined corner and edge detector. In: **CITeseer. ALVEY vision conference**. Manchester, Inglaterra, set. 1988. v. 15. P. 147–151.

HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. Nova Iorque, Estados Unidos: Cambridge University Press, 2003.

IPMA. **Imagens de Satélite - Visualização**. 2020. Disponível em: <https://www.ipma.pt/pt/educativa/observar.tempo/index.jsp?page=satelite03.xml>. Acessado em 20 jan. 2020.

JIANG, T; MERICKEL, M. B. Identification and boundary extraction of blobs in complex imagery. **Computerized Medical Imaging and Graphics**, v. 13, n. 5, p. 369–382, out. 1989.

KAZHDAN, M.; BOLITHO, M.; HOPPE, H. Poisson surface reconstruction. In: **EUROGRAPHICS symposium on Geometry processing**. 2006. v. 7. P. 61–70.

KUHN, A. et al. DeepC-MVS: Deep Confidence Prediction for Multi-View Stereo Reconstruction. **ArXiv**, abs/1912.00439, n. 3, dez. 2019.

KURZ, F. et al. Deep Learning Segmentation and 3D Reconstruction of Road Markings Using Multiview Aerial Imagery. **ISPRS International Journal of Geo-Information**, v. 8, n. 1, p. 47–63, jan. 2019.

LHULLIER, M.; QUAN, L. A quasi-dense approach to surface reconstruction from uncalibrated images. **IEEE transactions on pattern analysis and machine intelligence**, v. 27, n. 3, p. 418–433, mar. 2005.

LI, S. et al. Efficient multi-view surface refinement with adaptive resolution control. In: LEIBE, B. et al. **14th European Conference on Computer Vision**. Países Baixos: Springer, out. 2016. p. 349-364.

LOWE, D. G. Object recognition from local scale-invariant features. In: **SEVENTH IEEE International Conference on Computer Vision**. 1999. P. 1150–1157.

_____. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, v. 60, n. 2, p. 91–110, nov. 2004.

LUCIEER, A.; ROBINSON, S. A.; TURNER, D. Using an unmanned aerial vehicle (UAV) for ultra-high resolution mapping of Antarctic moss beds. **Australasian Remote Sensing Photogrammetry Conference**, Alice Springs, Austrália, v. 15, jan. 2010.

LUCIEER, A.; TURNER, D. et al. Using an Unmanned Aerial Vehicle (UAV) to capture micro-topography of Antarctic moss beds. **International Journal of Applied Earth Observation and Geoinformation**, v. 27, p. 53–62, abr. 2014.

LUONG, Q.; FAUGERAS, O. D. The fundamental matrix: Theory, algorithms, and stability analysis. **International journal of computer vision**, v. 17, n. 1, p. 43–75, jul. 1996.

MA, Y. et al. **An invitation to 3-d vision: from images to geometric models**. Nova Iorque, Estados Unidos: Springer Science & Business Media, 2012.

MALIHI, S. et al. 3D building reconstruction using dense photogrammetric point cloud. **International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, Praga, República Checa, v. 3, p. 71–74, jul. 2016.

- MANCINI, F. et al. Using unmanned aerial vehicles (UAV) for high-resolution reconstruction of topography: The structure from motion approach on coastal environments. **Remote Sensing**, v. 5, n. 12, p. 6880–6898, dez. 2013.
- MATTOCCIA, S.; TOMBARI, F.; DI STEFANO, L. Reliable rejection of mismatching candidates for efficient ZNCC template matching. In: **15TH IEEE International Conference on Image Processing**. San Diego, Estados Unidos, out. 2008. P. 849–852.
- MEINEN, B.U.; ROBINSON, D. T. Mapping erosion and deposition in an agricultural landscape: Optimization of UAV image acquisition schemes for SfM-MVS. **Remote Sensing of Environment**, v. 239, p. 111666, mar. 2020.
- MUNDY, J. L. et al. **Geometric invariance in computer vision**. Estados Unidos: The MIT press, 1992.
- NESBIT, P. R.; HUGENHOLTZ, C. H. Enhancing UAV–SfM 3D Model Accuracy in High-Relief Landscapes by Incorporating Oblique Images. **Remote Sensing**, v. 11, n. 3, p. 239–263, jan. 2019.
- OPENCV. **Shi-Tomasi Corner Detector Good Features to Track**. 2020.
Disponível em:
https://docs.opencv.org/3.1.0/d4/d8c/tutorial_py_shi_tomasi.html. Acessado em 20 jan. 2020.
- PAGÁN, J. I. et al. Monitoring the dune-beach system of Guardamar del Segura (Spain) using UAV, SfM and GIS techniques. **Science of the total environment**, v. 687, p. 1034–1045, out. 2019.
- PAREEK, A.; ARORA, N. Evaluation of Feature Detector-Descriptor Using RANSAC for Visual Tracking. **Available at SSRN 3354470**, 2019.
- PIJL, A. et al. Design of Terrace drainage networks using UAV-based high-resolution topographic data. **Water**, v. 11, n. 4, p. 814, abr. 2019.
- RIQUELME, A. et al. Digital landform reconstruction using old and recent open access digital aerial photos. **Geomorphology**, v. 329, p. 206–223, mar. 2019.
- SCHMID, C.; ZISSERMAN, A. The geometry and matching of curves in multiple views. In: BURKHARDT, H.; NEUMANN B. **European Conference on Computer Vision**. Freiburg, Alemanha: Springer, jun. 1998. p. 394-409.

- SCHONBERGER, J. L.; FRAHM, J. M. Structure-from-motion revisited. In: **IEEE Conference on Computer Vision and Pattern Recognition**. Las Vegas, Estados Unidos, jun. 2016. P. 4104–4113.
- SEITZ, S. M. et al. A comparison and evaluation of multi-view stereo reconstruction algorithms. In: **IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)**. Nova Iorque, Estados Unidos, jun. 2006. v. 1. P. 519–528.
- SHAH, R.; DESHPANDE, A.; NARAYANAN, P. J. Multistage sfm: Revisiting incremental structure from motion. In: **2ND IEEE International Conference on 3D Vision**. Tóquio, Japão, fev. 2014. P. 417–424.
- SMITH, M. W.; CARRIVICK, J. L.; QUINCEY, D. J. Structure from motion photogrammetry in physical geography. **Progress in Physical Geography**, Londres, Inglaterra, v. 40, n. 2, p. 247–275, nov. 2016.
- SNAVELY, N.; SEITZ, S. M.; SZELISKI, R. Photo tourism: exploring photo collections in 3D. In: **ACM SIGGRAPH 2006 Papers**. Nova Iorque, Estados Unidos, jul. 2006. v. 25. P. 835–846.
- SOLEM, J. E. **Programming Computer Vision with Python: Tools and algorithms for analyzing images**. Sebastopol, Estados Unidos: O'Reilly Media, Inc., 2012.
- STRECHA, C. et al. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: **IEEE Conference on Computer Vision and Pattern Recognition**. Anchorage, Estados Unidos, jun. 2008. P. 1–8.
- STROBL, K. H.; HIRZINGER, G. More accurate pinhole camera calibration with imperfect planar target. In: **IEEE International Conference on Computer Vision Workshops (ICCV Workshops)**. Barcelona, Espanha, nov. 2011. P. 1068–1075.
- SWIRAD, Z. M; ROSSER, N. J.; BRAIN, M. J. Identifying mechanisms of shore platform erosion using Structure-from-Motion (SfM) photogrammetry. **Earth Surface Processes and Landforms**, v. 44, p. 1542–1558, jan. 2019.
- SZELISKI, R. **Computer vision: algorithms and applications**. Londres, Inglaterra: Springer Science & Business Media, 2010.
- THOENI, K. et al. The potential of low-cost RPAS for multi-view reconstruction of sub-vertical rock faces. **International Archives of the Photogrammetry, Remote**

Sensing & Spatial Information Sciences, Praga, República Checa, v. 41, p. 909–916, jul. 2016.

TOLA, E.; LEPETIT, V.; FUA, P. A fast local descriptor for dense matching. In: **IEEE conference on computer vision and pattern recognition**. Anchorage, Estados Unidos, jun. 2008. P. 1–8.

VOGIATZIS, G. et al. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 29, n. 12, p. 2241–2246, nov. 2007.

WARFIELD, A. D.; LEON, J. X. Estimating Mangrove Forest Volume Using Terrestrial Laser Scanning and UAV-Derived Structure-from-Motion. **Drones**, v. 3, n. 2, p. 32, abr. 2019.

WESTOBY, M. J. et al. Structure-from-Motion photogrammetry: A low-cost, effective tool for geoscience applications. **Geomorphology**, v. 179, p. 300–314, dez. 2012.

WU, C. Towards linear-time incremental structure from motion. In: **IEEE International Conference on 3D Vision-3DV**. Seattle, Estados Unidos, jun. 2013. P. 127–134.

XINGFANG, Y.; YUMEI, H.; FENG, G. A simple camera calibration method based on sub-pixel corner extraction of the chessboard image. In: **IEEE International Conference on Intelligent Computing and Intelligent Systems**. Xiamen, China, out. 2010. v. 3. P. 688–692.

YANG, J.; LI, H.; JIA, Y. Optimal essential matrix estimation via inlier-set maximization. In: FLEET, D. et al. **13th European Conference on Computer Vision**. Zúrich, Suíça: Springer, set. 2014. p. 111–126.

YI, G. et al. Survey of structure from motion. In: **IEEE International Conference on Cloud Computing and Internet of Things**. Changchun, China, dez. 2014. P. 72–76.

YLIMÄKI, M. et al. Fast and accurate multi-view reconstruction by multi-stage prioritised matching. **IET Computer Vision**, v. 9, n. 4, p. 576–587, mar. 2015.

ZACH, C. Fast and high quality fusion of depth maps. In: **CITeseer International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)**. Atlanta, Estados Unidos, jun. 2008. v. 1.

ZACH, C.; POCK, T.; BISCHOF, H. A globally optimal algorithm for robust TV-L1 range image integration. In: **11th IEEE International Conference on Computer Vision**. Rio de Janeiro, out. 2007. P. 1–8.

ZIMMER, B. et al. Using differential structure-from-motion photogrammetry to quantify erosion at the Engare Sero footprint site, Tanzania. **Quaternary Science Reviews**, v. 198, p. 226–241, out. 2018.

APÊNDICE A – Parâmetros Extrínsecos das Câmeras

A Tabela 12 apresenta os valores de odometria estimados utilizando a metodologia SFM. Vale ressaltar que os valores de translação não possuem escala. A fim de uma melhor visualização, foi realizado a transformação da matriz de rotação para os ângulos de Euler.

Tabela 12 – Parâmetros extrínsecos de cada câmera.

Câmera	Rotação			Translação		
	α	β	γ	x	y	z
1	0.2784	-0.0069	0.0040	33.205	19.963	0.0190
2	0.0584	-0.0062	0.0049	35.176	21.546	0.0236
3	0.0361	-0.0054	0.0043	34.604	31.041	0.0262
4	0.0407	-0.0038	0.0044	33.659	41.052	0.0340
5	0.0306	-0.0036	0.0023	33.442	50.487	0.0355
6	0.0253	-0.0039	0.0012	32.734	56.849	0.0378
7	-31.064	0.0257	-0.0080	-11.136	-86.185	-0.1211
8	-31.376	0.0243	-0.0070	-14.790	-77.331	-0.1126
9	-31.382	0.0211	-0.0055	-15.730	-66.890	-0.0844
10	-31.405	0.0209	-0.0039	-16.168	-57.141	-0.0551
11	-31.408	0.0201	-0.0021	-16.607	-47.201	-0.0387
12	-31.335	0.0206	-0.0022	-16.647	-37.540	-0.0375
13	-31.373	0.0221	-0.0000	-17.087	-27.670	-0.0288
14	-31.409	0.0198	-0.0004	-17.612	-17.715	-0.0221
15	-31.414	0.0201	0.0011	-17.958	-0.7836	-0.0197
16	-31.403	0.0199	0.0028	-18.389	0.2192	-0.0103
17	-31.331	0.0204	0.0032	-18.899	11.649	-0.0170
18	-31.338	0.0202	0.0034	-19.359	21.624	-0.0229
19	-0.0601	-0.0067	0.0137	-0.1778	-49.492	-0.0855
20	0.0054	-0.0049	0.0117	0.1763	-39.716	-0.0645
21	-0.0002	0.0001	0.0110	0.1019	-29.173	-0.0395
22	-0.0056	-0.0016	0.0048	0.0665	-19.623	-0.0199
23	0.0036	-0.0008	0.0035	0.0441	-0.9705	-0.0096
24	0.0000	-0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0070	-0.0003	-0.0023	-0.0519	0.9890	-0.0088
26	0.0087	0.0003	-0.0019	-0.1096	19.564	-0.0157
27	0.0146	0.0007	-0.0042	-0.1715	29.311	-0.0222
28	0.0074	0.0027	-0.0045	-0.2036	39.286	-0.0303
29	0.0050	0.0041	-0.0093	-0.2410	48.924	-0.0581

30	0.0061	0.0039	-0.0080	-0.2866	58.806	-0.0670
31	0.0024	0.0030	-0.0086	-0.3128	68.631	-0.0801
32	0.0046	0.0025	-0.0083	-0.3764	78.608	-0.0890
33	0.0104	0.0021	-0.0089	-0.5801	95.308	-0.1106
34	-30.749	0.0291	-0.0172	29.634	-92.238	-0.1662
35	-31.343	0.0286	-0.0139	22.832	-84.778	-0.1256
36	-31.413	0.0285	-0.0105	21.573	-74.896	-0.0911
37	-31.386	0.0240	-0.0154	21.436	-64.858	-0.1270
38	-31.390	0.0245	-0.0120	21.142	-55.022	-0.0966
39	31.366	0.0246	-0.0105	20.459	-45.315	-0.0870
40	31.386	0.0250	-0.0073	20.246	-35.436	-0.0777
41	-31.390	0.0241	-0.0011	19.894	-25.574	-0.0469
42	-31.355	0.0184	-0.0040	19.383	-15.473	-0.0701
43	-31.376	0.0211	0.0007	19.047	-0.5816	-0.0779
44	-31.368	0.0247	0.0027	18.188	13.807	-0.0659
45	-31.271	0.0267	0.0050	17.557	23.821	-0.0769
46	-31.252	0.0273	0.0062	16.881	33.751	-0.0801
47	-31.287	0.0250	0.0125	16.031	50.479	-0.0925
48	-31.208	0.0254	0.0104	15.466	53.727	-0.0828
49	-31.188	0.0258	0.0066	14.734	63.232	-0.0519
50	-31.368	0.0262	0.0070	16.366	64.079	-0.0463
51	-0.0960	-0.0005	0.0132	-41.160	-53.518	-0.1526
52	0.0165	-0.0004	0.0097	-34.185	-48.802	-0.1142
53	0.0088	0.0009	0.0069	-34.767	-38.027	-0.0905
54	0.0048	0.0028	0.0055	-35.293	-28.058	-0.0705
55	0.0046	0.0039	0.0053	-35.739	-18.164	-0.0607
56	0.0138	0.0054	-0.0033	-36.149	-0.8989	-0.0331
57	0.0202	0.0038	-0.0048	-36.630	0.0755	-0.0346
58	0.0134	0.0043	-0.0057	-37.178	10.773	-0.0310
59	0.0141	0.0038	-0.0068	-37.671	20.532	-0.0458
60	0.0076	0.0031	-0.0080	-37.773	30.594	-0.0748
61	0.0009	0.0027	-0.0073	-37.902	40.885	-0.0872
62	0.0020	0.0029	-0.0073	-38.386	50.695	-0.0978
63	0.0058	0.0026	-0.0072	-38.977	60.348	-0.1154
64	0.0089	0.0025	-0.0062	-39.641	70.168	-0.1218
65	0.0092	0.0026	-0.0072	-40.139	79.910	-0.1444