UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MARCELO LEMOS ROSSI

MODELAGEM 3D DE ÁREAS URBANAS UTILIZANDO PROCESSAMENTO DIGITAL DE IMAGEM

UBERLÂNDIA
2018

MARCELO LEMOS ROSSI

MODELAGEM 3D DE ÁREAS URBANAS UTILIZANDO PROCESSAMENTO DIGITAL DE IMAGEM

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Engenharia Elétrica, no curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Uberlândia, Área de concentração: Processamento Digital de Imagens.

Orientador: Prof. Dr. Gilberto Arantes Carrijo

UBERLÂNDIA

2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

R834m 2018

Rossi, Marcelo Lemos, 1983-
Modelagem 3D de áreas urbanas utilizando processamento digital de imagem / Marcelo Lemos Rossi. - 2018.
189 f. : il.

Orientador: Gilberto Arantes Carrijo.
Tese (doutorado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.
Disponível em: http://dx.doi.org/10.14393/ufu.te.2018.49
Inclui bibliografia.

1. Engenharia elétrica - Teses. 2. Processamento de imagens - Teses. 3. Planejamento urbano - Teses. I. Carrijo, Gilberto Arantes, 1948-. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 621.3

Maria Salete de Freitas Pinheiro – CRB6/1262

TERMO DE APROVAÇÃO

MARCELO LEMOS ROSSI

MODELAGEM 3D DE ÁREAS URBANAS UTILIZANDO PROCESSAMENTO DIGITAL DE IMAGEM

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Engenharia Elétrica, no curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Uberlândia, Área de concentração: Processamento Digital de Imagens, pela seguinte banca examinadora:

_____

Prof. Dr. Gilberto Arantes Carrijo
Orientador - Departamento de Engenharia Elétrica, UFU

_____

Prof. Dr. Ed' Wilson Tavares Ferreira
Departamento de Área de Informática, IFMT

_____

Prof. Dr. Júlio Cesar Ferreira
Sistemas de Informação, IF Goiano

_____

Profa. Dra. Milena Bueno Pereira Carneiro
Departamento de Engenharia Elétrica, UFU

_____

Prof. Dr. Antônio Cláudio Paschoarelli Veiga
Departamento de Engenharia Elétrica, UFU

Uberlândia, 08 de fevereiro de 2018.

DEDICATÓRIA

À minha família que sempre me apoiou e
me incentivou a continuar batalhando
para realizar esse trabalho

AGRADECIMENTOS

Ao programa de Pós-Graduação da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia pela oportunidade da realização deste curso e à CAPES pela concessão da bolsa de estudos.

Ao Prof. Gilberto Arantes Carrijo pela orientação, ensinamentos, discussões e pela confiança depositada, à Prof. Edna Lúcia Flôres e ao Prof. Antônio Cláudio Paschoarelli Veiga pelos auxílios no trabalho e pela ajuda.

À equipe do Laboratório de Processamento Digital de Sinais sempre prontos para ajudarem e pelo apoio técnico.

À técnica Cinara Fagundes muito simpática e sempre colaborando com informações

AGRADECIMENTO ESPECIAL

À minha querida Família!

RESUMO

De um lado tem-se a crescente atenção dada ao planejamento urbano em especial na expansão urbana, no planejamento e na arrecadação. Diversos trabalhos foram realizados para transformar o planejamento urbano de forma mais científico com a finalidade de melhorar o planejamento urbano. Com isso os governos têm exigido mecanismos cada vez mais inteligentes no gerenciamento das cidades. No Brasil isso não é diferente, de tal forma que pela Lei Estatuto da Cidade, o governo tem exigido que as cidades apresentem alguns mecanismos de planejamento urbano. Apesar da Lei exigida pelo governo brasileiro e os guias disponibilizados por ele apresentarem os mecanismos necessários para elaborar o planejamento urbano, não são descritas as ferramentas nem os métodos necessários para desenvolver os mecanismos de análise das cidades. De outro lado tem-se a visão como o principal mecanismo que o ser humano utiliza para reconhecer o mundo ao seu redor, sendo ela considerada um dos sentidos mais importantes que as pessoas possuem, pois através deste sentido que pode-se reconhecer, classificar e diferenciar as coisas que estão ao redor. Sendo a visão um dos sentidos mais importantes torna-se interessante transferir essa capacidade para computadores e máquinas, tornando-os capazes de identificar o ambiente assim como os seres humanos identificam. Da união da necessidade de ferramentas para contribuir com o planejamento urbano e a importância que a visão tem na realização de ações importantes surge o objetivo deste trabalho que é fornecer mais uma ferramenta, utilizando tecnologias da visão computacional, que contribua para o planejamento urbano fornecendo mais informações para os diretores de cidades, os projetistas e os urbanistas a respeito da volumetria das áreas urbanas. Ou seja, este trabalho apresenta como aproveitar fotografias 2D terrestres de áreas urbanas e, a partir delas, como obter modelos 3D das estruturas urbanas. Contribuindo, assim, como mais uma ferramenta para a elaboração de melhores planos diretores e um crescimento mais sustentável das cidades.

Palavras-chave: Reconstrução 3D; Escaneamento 3D; Visão Computacional; Processamento de Imagem

ABSTRACT

On one hand it is increasing the attention to urban planning, especially in urban expansion, planning and taxes. Several studies were performed with the purpose of transforming urban planning in a more scientific way to improve urban planning. Thus, the governments are demanding the use of more intelligent mechanisms in the management of the cities. In Brazil this is no different. Through the City Statute Law, the Brazilian government is demanding that the cities present some urban planning mechanisms. Although this law and the guides, offered by the Brazilian government, indicate the mechanisms to develop urban planning they don't described the tools or the methods to develop the analysis that is demanded. On the other hand, the vision is the main mechanism that humans use to recognize the world around them in a way that it is considered one of the most important sense that people have. Is through this sense that one can recognize, classify and differentiate the things that are around. As the vision being a so important sense, it is interesting give this ability to computers and machines, making them able to identify the environment as well as human beings identify. By the union of the need for new tools to contribute to the urban planning and the importance that vision has in carrying out important actions emerges the objective of this work: to provide an additional tool using computer vision technologies to the urban planning, providing more information for directors of cities, urban designers and urban planners about the volumes of urban areas. In other words, this work presents how to take advantage of 2D terrestrial photographs of urban areas to get 3D models of urban structures. Thus, contributing as an additional tool for the development better urban master plans and regional plans contributing to a more sustainable growth of cities.

Keywords: 3D Reconstruction; 3D Scanning; Computer Vision; Image Processing

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

SUMÁRIO

CAPITULO 1

1 INTRODUÇÃO

1.1 Motivação

A visão é o principal mecanismo que o ser humano utiliza para reconhecer o mundo ao seu redor. Dessa forma a visão é considerada um dos sentidos mais importantes que as pessoas possuem, sendo através deste sentido que se pode reconhecer, classificar e diferenciar as coisas que estão ao redor. Como esse é um dos sentidos mais importantes é interessante transferir essa capacidade para computadores e máquinas, tornando-os capazes de identificar o ambiente, assim como os seres humanos o identificam. A área que estuda a forma com que as máquinas podem analisar o ambiente ao seu redor é denominada visão computacional.

A visão computacional é utilizada em larga escala para o reconhecimento, a identificação e a autorização de objetos (SMEETS et al., 2010; RAHMAN et al., 2007) e, também, para o reconhecimento, a identificação e autorização de pessoas (ZHANG; HE, 2010; QIAKAI; CHAO; JING, 2012). Quando é utilizada para análise de pessoas, normalmente, ela é realizada pela análise da face, impressão digital, geometria da mão, veias da mão, íris, retina e assinatura.

Devido à grande potencialidade da visão computacional, o seu uso tem excedido o tema de reconhecimento para fins de identificação e autorização, sendo ela utilizada em diversas outras áreas, como classificação de veículos quanto ao tamanho e o número de eixos para as análises do fluxo de veículos, controle do trânsito e de pedágios (ZHU-YU; TIAN-MIN; XIAN-YANG, 2011; WANG, 2009; LOVE; MASAKI; HORN, 2000). De forma mais futurista, a visão computacional pode ser usada para a análise do movimento dos veículos na pista, servindo de computador de bordo em um carro inteligente, auxiliando os motoristas (KATO; NINOMIYA; MASAKI, 2002).

Outro campo que tem crescido com a análise de imagens 2D é o de geoprocessamento, utilizando a análise direta de mapas (EMMERT, 2010) ou pela

análise das imagens de construções, auxiliando no plano diretor das cidades como, por exemplo, a classificação computadorizada de construções residencial, comercial ou industrial (ZHANG; KOŠECKÁ, 2007; TRINH; KIM; JO, 2010).

Entretanto, em alguns momentos as técnicas de análise de uma única imagem em 2D não são suficientes para todas as aplicações. Assim, é necessário realizar a análise dos objetos utilizando modelos em 3D. Como exemplo de aplicações em que uma única imagem em 2D não é suficiente para modelar o objeto/ambiente pode-se citar o trabalho de Carrillo et al (2012) que usou duas câmeras gerando, assim, imagens em 3D para o sistema de navegação de veículos não-tripulado. Kamitomo e Lu (2012) conseguiram melhorias de reconhecimento de face utilizando um modelo em 3D da pessoa. O trabalho de Henn et al (2012) melhorou a classificação e discretização de imóveis quando utilizou modelos em 3D.

1.2 Objetivo Geral

O objetivo geral deste trabalho é fornecer uma ferramenta utilizando tecnologias da visão computacional, que possa fornecer mais informações para os diretores, os projetistas e os urbanistas a respeito da volumetria das cidades. Ou seja, será apresentado neste trabalho como obter modelos 3D de áreas urbanas através de fotografias terrestres. Contribuindo, assim, para a elaboração de melhores planos diretores e um crescimento mais sustentável das cidades.

1.3 Objetivo Específico

O objetivo específico deste trabalho é apresentar uma metodologia que seja capaz de reconstruir em 3D ambientes urbanos utilizando apenas fotografias terrestres da área urbana de interesse e, com isso, fornecer uma ferramenta que os urbanistas possam utilizar para a elaboração do planejamento urbano e que possa ser aproveitado em ferramentas Sistema de Informação Geográfica (SIG). Para isso serão abordados os temas:

a) Obtenção de características da imagem;
b) Encontrar a posição das câmeras por meio das características da imagem;

c) Reconstrução da cena em 3D

1.4 Justificativa

O planejamento urbano tem recebido muita atenção ultimamente na expansão urbana, no planejamento e na arrecadação. No início esses planejamentos eram realizados de forma subjetiva, onde o governante da cidade definia o que era mais interessante para a cidade (LAURINI; 1982A). Diversos trabalhos foram realizados para transformar o planejamento urbano de forma mais científica (LAURINI; 1982B). Com a finalidade de melhorar o planejamento urbano, diversos países têm exigido mecanismos mais inteligentes no gerenciamento das cidades. No Brasil isso não é diferente, de tal forma que pela Lei (BRASIL; 2001) o governo tem exigido que as cidades apresentem alguns mecanismos de planejamento urbano (BARBOSA; COSTA, 2004; CARVALHO et al., 2010).

A Lei exigida pelo governo brasileiro e os guias disponibilizados por ele apresentam os mecanismos necessários para elaborar o planejamento urbano. Entretanto, não é apresentado as ferramentas necessárias para desenvolver os mecanismos de análise das cidades. Dessa forma, as prefeituras utilizam ferramentas de análise de mapas computacionais como o SIG (LAURINI, 2001).

Para a utilização das ferramentas SIG existem diversos mapas 2D ou a possibilidade de obtê-los por imagens de satélites (LAURINI, 2001). Porém, Laurini (2001) apresentou a necessidade de utilizar modelos 3D nas ferramentas SIG. Assim, atualmente, tem sido utilizado o sistema de *Light Detection And Ranging* (LIDAR) para escanear e obter o modelo 3D das cidades. Entretanto, os equipamentos de LIDAR são caros e dispendiosos. Por outro lado, existem veículos equipados com câmeras capazes de obter fotografias de vias e fachadas de edificação, como, por exemplo, o do Google Street View®. O custo dos equipamentos para obter essas fotografias é menor do que os que utilizam o LIDAR.

1.5 Trabalhos Relacionados

Como trabalhos relacionados ao planejamento urbano podem ser citados os trabalhos de Laurini (1982B) que demonstra a necessidade de um planejamento

urbano mais científico, indicando que se deve utilizar ferramentas capazes de fornecer informações que possam contribuir para isso. O próprio Laurini (2001) destaca que informações 3D de áreas urbanas podem contribuir para essa melhoria.

Como trabalhos relacionados às melhorias que a visão computacional trouxe temos os trabalhos de Carrillo et al (2012) que conseguiu melhorar a navegação de veículos não-tripulado ao utilizar duas câmeras e, assim, obter informações 3D do percurso. Outro exemplo da melhoria de resultados utilizando imagens 3D é o trabalho de Kamitomo e Lu (2012) sobre o reconhecimento de face após ter obtido o modelo em 3D da pessoa. Por fim, temos o trabalho de Henn et al (2012) que indicou uma melhoria na classificação e discretização de imóveis quando utilizou um banco de dados com modelos em 3D de áreas urbanas.

1.6 Metodologia

Este trabalho será realizado em três etapas experimentais em que cada etapa é mais complexa que a anterior.

Para a primeira etapa foi desenvolvido uma bancada de testes em que algumas das variáveis do processo são conhecidas *a priori*, como as posições de câmera e o fundo da cena. Utilizando esta bancada é criado um banco de imagens cujas posições de câmera são conhecidas. Dessa forma o foco desta etapa é apresentar metodologias de segmentação da imagem, casamento de pontos característicos e a triangulação para a reconstrução 3D de um objeto.

Na segunda etapa é removido o conhecimento *a priori* das variáveis do processo de reconstrução 3D, sendo o banco de imagens criado colocando a câmera em posições aleatórias, porém capturando vários ângulos da cena. Com isso nesta etapa será abordado uma metodologia capaz de recuperar as posições de câmera, analisando apenas os pontos homólogos encontrados nos pares de imagens, e a reconstrução 3D quando não se conhece as posições de câmera.

Por fim, na última etapa foi realizada a reconstrução 3D de um prédio histórico da cidade de Pelotas/RS. Nesta etapa o banco de imagens foi criado sem que soubesse as posições de câmera. Assim o foco desta etapa é demonstrar que a metodologia apresenta neste trabalho consegue realizar a reconstrução 3D de uma cena urbana.

1.7 Contribuições

Espera-se que as discussões apresentadas neste trabalho possam contribuir como uma ferramenta a ser utilizada pelos diretores de cidades, os projetistas e os urbanistas na elaboração dos planos diretores e nas tomadas de decisões permitindo, assim, um melhor desenvolvimento e aproveitamento das áreas urbanas.

1.8 Conclusões Parciais

Espera-se realizar, ao término deste trabalho, a reconstrução 3D de uma cena urbana de qualidade suficiente para ser utilizada em softwares SIG e que possam auxiliar os urbanistas na elaboração e fiscalização de planos diretores.

CAPITULO 2

2 TÉCNICA UTILIZADA PARA MODELAGEM 3D A PARTIR DE IMAGENS

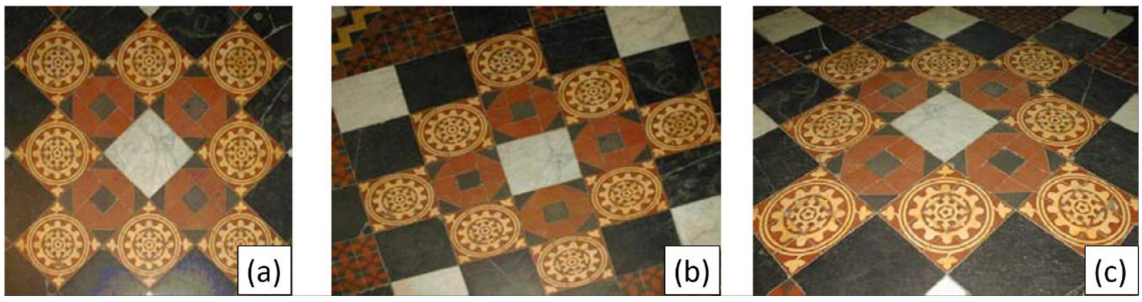
2.1 Introdução

Este capítulo apresenta a técnica utilizada para analisar as imagens e como desenvolver o modelo em 3D utilizando representações 2D das cenas.

Apesar de uma única imagem em 2D não possuir a mesma quantidade de informações que um objeto em 3D possui é possível, por uma sequência adequada de imagens em 2D, construir um modelo em 3D que se aproxima, o melhor possível, do objeto real. A construção do modelo em 3D pode ser realizada ao analisar informações nessa sequência de imagens que permitam identificar a profundidade da cena e do objeto em interesse.

Pode parecer simples retirar informações da imagem para a construção do modelo 3D. Entretanto, as imagens obtidas pela câmera não correspondem ao formato do objeto real. Por exemplo, círculos se tornam elipses, quadrados se transformam em losangos e retas paralelas podem deixar de ser paralelas. As Figura 2.1 e Figura 2.2 apresentam exemplos de deformações causadas pela câmera.

Figura 2.1 – Alteração na forma geométrica das imagens devido à posição da câmera



(a) a forma geométrica real; (b) a posição da câmera altera algumas relações de ângulo, mais mantêm as relações de paralelismo e perpendicularidade; (c) a posição da câmera altera os ângulos e as relações de paralelismo e perpendicularidade.
Fonte: adaptado de Hartley e Zisserman, 2004.

Figura 2.2 – Os trilhos são paralelos, mas na imagem eles não são paralelos e convergem para um ponto

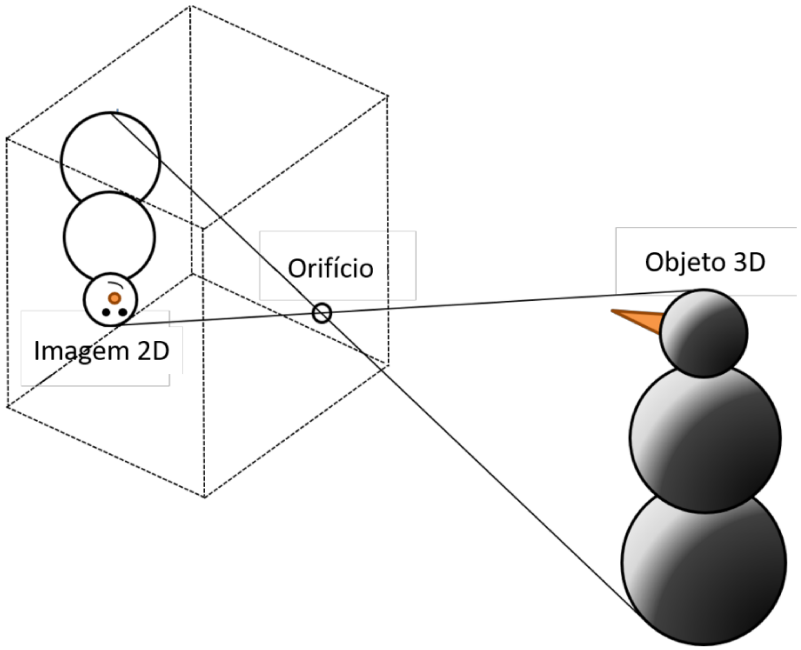
Fonte: Imagem de ferrovia obtida pelo Google

As modificações geométricas do objeto como perda do paralelismo, perdas das relações dos ângulos, etc., surgem devido a uma transformação não-linear causada pela câmera ao transformar uma cena 3D em uma imagem 2D. Assim, a câmera distorce os pontos que definem o objeto. Essa transformação converte as três coordenadas dos pontos do objeto, que ocupam um espaço vetorial de três dimensões ($\mathcal{R}^3$) em uma imagem que corresponde a um sistema de apenas duas coordenadas, ou seja, um espaço vetorial de duas dimensões ($\mathcal{R}^2$).

O processo em que se obtém imagens 2D de cenas 3D, processo no qual a informação de uma das dimensões é perdida, é denominado de projeção. O método mais comum de modelar esse processo é o da projeção central. Nesse processo de projeção conecta-se todos os pontos do espaço vetorial 3D por traços em um único ponto fixo denominado de centro de projeção. Todos esses traços interceptam um plano escolhido como sendo o plano da imagem. As interseções dos traços com o plano da imagem representam os pontos que compõem a imagem. A Figura 2.3 apresenta o processo de projeção central utilizando um modelo de câmera de orifício.

Como a transformada da projeção remove uma das coordenadas, torna-se impossível de restaurar as coordenadas originais apenas com uma única transformada reversa, devido à ausência de informações necessárias para essa transformação. Em outras palavras, a informação da profundidade dos pontos do objeto 3D em relação ao plano de imagem é perdida e não existe uma operação reversa que possa recuperar essa informação. Dessa forma, a transformada da projeção pode ser considerada uma transformação não-linear.

Figura 2.3 – Exemplo de uma câmera de orifício realizando o processo de processo da projeção central



Fonte: Autor

Devido a essa perda de informação não é mais possível recuperar, de forma exata, a informação da profundidade dos pontos do objeto utilizando apenas uma única imagem 2D. Mas, ao encontrar os pontos do objeto 3D em duas ou mais imagens diferentes, é possível recuperar a informação da profundidade do ponto, conhecendo a posição das câmeras, e, então, reconstruir a forma inicial do objeto construindo, assim, um modelo 3D do mesmo (SONKA; HLAVAC; BOYLE, 2006).

As diversas razões da dificuldade da visão 3D utilizando imagens como entrada são:

- Os sistemas de imagens das câmeras e do olho humano realizam a projeção da perspectiva e isso resulta em uma grande perda de informações da cena. Todos os pontos de uma linha que apontam do centro focal até um ponto da cena são projetados como sendo um único ponto na imagem gerada.
- A realização de uma tarefa inversa, ou seja, pelas medições nas imagens encontrar todas as coordenadas 3D dos pontos da cena. Dessa forma, tem-se um sistema sem restrição (sistema possível e indeterminado – contém várias soluções). Assim, algumas informações adicionais devem ser informadas para resolver as ambiguidades;

- Relacionar a intensidade dos pontos de uma imagem e a geometria do modelo 3D é complicado. Essa intensidade depende de alguns parâmetros da superfície, como refletividade, orientação, tipo e posição dos iluminadores da superfície e, também, da posição do observador (câmera). Reconhecer a geometria (estrutura 3D) das superfícies, como orientação e profundidade, é difícil, pois a transformação projetiva elimina a informação de profundidade;
- A oclusão mútua dos objetos na cena, e mesmo a auto oclusão de um objeto, prejudica o reconhecimento da geometria; e
- A presença de ruídos na imagem e o custo computacional de alguns algoritmos também dificultam o reconhecimento da geometria e o desenvolvimento do modelo 3D.

Este capítulo está organizado da seguinte maneira: Serão apresentados vários paradigmas da visão em 3D, em especial o teorema de Marr da visão 3D. Apesar do teorema de Marr ter sido desenvolvido no final da década de 1970 ele é o mais aceito e mais utilizado. Em seguida, são mostrados os problemas de geometria que devem ser resolvidos para que o computador seja capaz de criar representações em 3D e gerar o modelo 3D da cena. Finalmente, é apresentada a relação de um ponto na imagem e a sua posição no espaço 3D e as considerações finais deste capítulo.

2.2 Visão 3D

O estudo da visão 3D pela projeção projetiva é relativamente novo e não existe uma teoria unificada disponível. Diferentes grupos de pesquisa apresentam diferentes entendimentos do mecanismo de reconhecimento da profundidade dos pontos da projeção. Os diversos algoritmos para a visão 3D e os seus paradigmas relativos apresentados a seguir apresentam várias opções.

Marr (1982) definiu a visão 3D como “A partir de uma imagem (ou conjunto de imagens) de uma cena, obter a descrição correta da geometria em três dimensões de uma cena e determinar de forma quantitativa as propriedades do objeto da cena”. Para Marr, a visão 3D é definida como a reconstrução de um objeto 3D, como exemplo: a descrição da forma 3D em um sistema de coordenadas independente do observador.

Um objeto rígido, que a separação do fundo da cena é realizada facilmente, e o controle do processo é estritamente bottom-up desde a intensidade da imagem até uma representação intermediária. Dessa forma, é apropriado considerar a visão 3D como uma reconstrução da cena. Se a acuidade visual é uma representação precisa de uma cena 3D, então pode ser realizada toda a tarefa que dependa da visão. Como exemplo de tarefas que dependem da visão, pode-se citar: sistemas de navegação de veículos autônomos; inspeção de peças; ou reconhecimento de objetos. O paradigma da reconstrução precisa conhecer a relação entre uma imagem e o mundo 3D correspondente, assim a formação da imagem deve ser descrita;

Aloimonos e Shulman (1989) observam o problema central da visão computacional como "... a partir de uma imagem (ou sequência delas) obtida por um observador monocular (ou poliocular) que se move (ou está estacionário em relação à cena) observando um objeto se movendo (ou estacionário na cena) é possível entender o objeto ou a cena e as suas propriedades tridimensionais". De acordo com essa definição, o conceito de entendimento é que torna essa abordagem da visão computacional diferente. Se apenas um pequeno conhecimento, a priori, está disponível, como na visão humana, então o entendimento do objeto (cena) é complicado. Isso pode ser visto como um caso limitante. Outro exemplo do extremo da complexidade do espectro é a simples verificação de objetos (object matching) em que existe apenas algumas interpretações possíveis;

Wechsler (1990) reforça que deve ser feito a aplicação o princípio do controle de processos diversas vezes, seguindo o princípio: "O sistema visual lança a maioria das tarefas visuais como problemas de minimização e tenta resolve-los usando computação distribuída com aplicações que possuem limitações naturais não acidentais". Para Wechsler, a visão computacional é uma representação paralela e um processamento distribuído de forma paralela, para obter uma percepção ativa da cena. O entendimento é dado através do ciclo "percepção – controle – ação".

Aloimonos (CVGIP B, 1992) questiona quais princípios devem permitir o entendimento do sistema visual de organismos vivos e, após conhece-los, será possível equipar as máquinas com essa capacidade visual. Ele apresentou algumas questões relativas ao entendimento da visão:

- Pergunta empírica – "O que é?": determina como os sistemas visuais são designados;

- Pergunta normativa – “O que deve ser?”: determina as características da visão natural e do sistema de visão ideal que é desejado para a tarefa; e
- Pergunta teórica – “O que pode ser?”: pergunta sobre o mecanismo que pode existir em um sistema inteligente de visão.

A teoria do sistema (KLIR, 2001) fornece uma estrutura geral que permite tratar a compreensão dos fenômenos complexos usando a matemática. A complexidade inerente da tarefa da visão é resolvida neste trabalho pela separação de objetos (ou sistemas ou fenômenos) do fundo da cena (background), em que “objetos” significam qualquer coisa de interesse para a resolução do problema em questão. Os objetos e as suas propriedades precisam ser caracterizados através de um modelo matemático formal que, normalmente, é utilizado para a abstração. O modelo é especificado por um número relativamente pequeno de parâmetros, os quais são, normalmente, estimados pelos dados da imagem.

O desenvolvimento de um sistema de visão por computador deve enfrentar três problemas entrelaçados:

1. Observação das características em uma imagem: É necessário determinar quando uma informação relevante à visão está presente nos dados da primeira imagem;
2. Representação: É a escolha do modelo em vários níveis de complexidade de interpretação do observador;
3. Interpretação: É a semântica dos dados. Em outras palavras, corresponde como os dados são mapeados para o mundo real. A tarefa é fazer com que certas informações sejam explicitas por meio de um modelo matemático registrado de forma implícita.

De acordo com o fluxo de informações e do conhecimento *a priori*, são consideradas duas aproximações para a visão artificial:

1. Reconstrução (bottom-up): O objetivo é reconstruir o formato 3D de um objeto utilizando uma imagem ou conjuntos de imagens. Essas imagens podem ser imagens de intensidade ou imagens distantes do observador. A teoria de Marr (Marr, 1982) estabelece o extremo na qual é estritamente bottom-up considerando pouco conhecimento *a priori* do objeto em questão. O objetivo de algumas aproximações mais práticas é desenvolver

um modelo 3D de um objeto real utilizando imagens distantes do objeto (FLYNN; JAIN, 1991; FLYNN; JAIN, 1992; SOUCY; LAURENDEAU, 1992; BOWYER et al., 1992); e

2. Reconhecimento baseado em visão (top-down): O conhecimento *a priori* do objeto é expresso pelos modelos de objetos, no qual os modelos 3D possuem um interesse particular (BROOKS; CREINER; BINFORD, 1979; BESL; JAIN, 1985; ARMAN; AGGARWAL, 1993). É de grande importância prática o reconhecimento baseado em modelos CAD (NEWMAN; FLYNN; JAIN, 1992). Em muitos casos restrições adicionais podem ser incorporadas nos modelos de forma a tornar um trabalho da visão computacional subdeterminado possível.

Outra possibilidade são sistemas de reconhecimento de objetos utilizando modelos em 3D. As aproximações baseadas em *priming* (utilização de geons) são baseadas nas formas 3D inferidas diretamente nos desenhos em 2D – as características qualitativas são chamadas de geons. Eles imitam o processo humano de reconhecimento no qual constituintes de um único objeto (geons) e o arranjo espacial deles são ponteiros para um pedaço da memória humana, permitindo o reconhecimento da organização espacial do objeto.

O alinhamento de vistas 2D é uma outra opção – linhas ou pontos em 2D podem ser utilizadas para o alinhamento de diferentes vistas em 2D. Primeiro, correspondência dos pontos, linhas ou outras características devem ser estabelecidos. Uma combinação linear das vistas tem sido utilizada para o reconhecimento (ULLMAN; BASRI, 1991). Várias questões relacionadas à representação da cena base da imagem na qual é armazenado uma coleção de imagens com correspondências estabelecidas em vez de um modelo 3D é considerada em (BEYMER; POGGIO, 1996). Uma metodologia que pode ser aplicada para apresentação de uma cena 2D por qualquer ponto de vista é apresentado em (WERNER; HERSCH; HLAVÁC, 1995).

2.2.1 Teoria de Marr

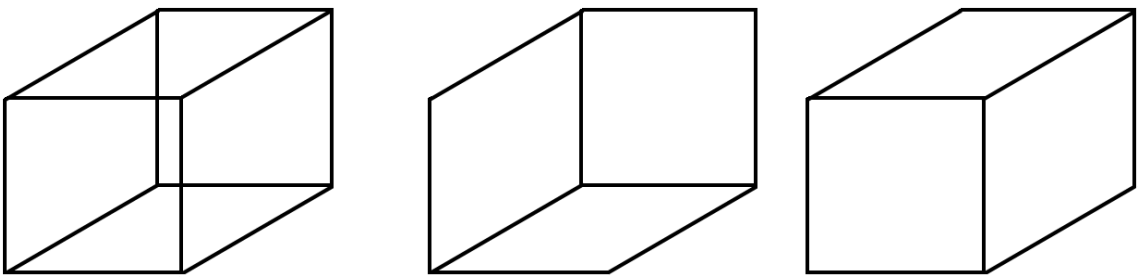
Marr foi o pioneiro no estudo de visão computacional. Apesar de sua morte prematura sua influência tem sido (e continua sendo) considerada. Marr era crítico dos trabalhos anteriores que, embora bem-sucedido em domínios limitados ou classes

de imagem, que eram empíricos ou excessivamente restritos às imagens que eles podiam lidar. Com isso, ele propôs uma aproximação mais abstrata e teórica que permitisse a utilização do trabalho em um contexto maior. Restringindo a interpretações 3D de uma única cena estática Marr propôs que um sistema de visão computacional fosse apenas um exemplo de um dispositivo para processamento de informações que pudesse ser entendido em três níveis:

- Teoria Computacional: Essa teoria descreve o que o dispositivo, supostamente, deve fazer – qual informação ele entrega após processar as informações de entrada. Ela também deve descrever a lógica da técnica utilizada para desenvolver a tarefa;
- Representação e algoritmo: Nessa etapa é apresentado, de forma precisa, o que o computador deve realizar para cumprir a tarefa – em particular, a representação das informações e o algoritmo para manipulá-las; e
- Implementação: Nesse nível é apresentado a realização física do algoritmo – especificamente, programas e hardware.

É importante ser claro sobre qual nível deve ser realizada a tentativa de resolver ou entender um problema em particular. Marr apresentou isso ao verificar que o efeito de uma pós-imagem é um efeito físico (efeito de observar um objeto sobre a mesa). Enquanto a confusão mental, como ocorre ao observar a ilusão causada pelo famoso cubo de Necker (Figura 2.4), parece estar em um nível teórico completamente diferente.

Figura 2.4 – Cubo de Necker e as duas interpretações possíveis devido a ilusão causada pela interpretação do cérebro de suas linhas



Fonte: Imagem do Google

Então, o ponto chave do sucesso é abordar a teoria em vez de algoritmos ou da implementação. Vários detectores de borda podem ser desenvolvidos, para a

solução de um problema específico, mas não estaria próximo de um mecanismo geral de entendimento de como a detecção de bordas deve ou precisa ser realizado, para a solução de todos os problemas. Marr ressaltou que a complexidade do processo de visão determina uma sequência de passos referentes à descrição da geometria das superfícies visíveis. Após ter obtido essa descrição da geometria é necessário remover a dependência do ponto de vista para transformar essa descrição em um objeto de interesse. Para o sucesso do processo é necessário, então, deixar de analisar os pixels e partir para a análise do traçado da superfície. Em seguida para a descrição das características da superfície e, finalmente, para a descrição completa do objeto em 3D. Essas operações podem ser realizadas partindo de uma imagem 2D para um **esboço inicial**, que se torna um **esboço 2,5D** e, então, procede-se para a completa **representação 3D**.

2.2.1.1 Esboço inicial

O objetivo principal do esboço inicial é capturar, de forma mais genérica possível, as informações significantes da alteração de intensidade na imagem. Tais alterações significantes de intensidade da imagem foram são, normalmente, referidas como bordas, mas Marr fez a observação de que esta palavra (bordas) implica um significado físico que não pode ser inferido nesta etapa. A primeira etapa é localizar essas mudanças em uma série de escalas. Informalmente pode-se dizer que é utilizado uma sequência de filtros de suavização na imagem, enquanto um detector de bordas por localização de cruzamento por zero, detecta o cruzamento por zero utilizando a derivada de segunda ordem da imagem a cada filtragem de suavização (MARR; HILDRETH, 1980). O filtro de suavização recomendado é um filtro Gaussiano padrão, mostrado na Equação (2.1). O cruzamento por zero é encontrado usando o operador Laplaciano, apresentado na Equação (2.2). Vários filtros de suavização isolam as características de escalas particulares. O cruzamento por zero realça, na mesma localização, das diversas escalas fortes evidencias das características físicas da cena.

$$G(x,y) = e^{-(x^2+y^2)/2\sigma^2} \tag{2.1}$$

$$\nabla^2 g(x,y) = \frac{\partial^2 g(x,y)}{\partial x^2} + \frac{\partial^2 g(x,y)}{\partial y^2} \tag{2.2}$$

Para completar o esboço inicial, os cruzamentos por zero são agrupados, de acordo com as suas localizações e orientações, de forma a prover informações a respeito das características da imagem como bordas, retas, manchas ou outras formas que podem prover informações da orientação das superfícies da cena. Na fase de agrupamento, deve-se tomar cuidado com as evidências das várias escalas, extraindo informações que são prováveis representações das superfícies no mundo real.

É importante observar que existem fortes evidências da existência de várias componentes usadas para construir o esboço inicial no sistema da visão humana, pois o ser humano também está preocupado em detectar as características das cenas em várias escalas, a localização de mudanças bruscas de intensidade e o posterior agrupamento em características da imagem.

2.2.1.2 O esboço 2,5D

O esboço 2,5D reconstrói as distâncias relativas do observador até as superfícies detectadas na cena, assim ele pode ser chamado, também, de mapa de profundidade (*depth map*). Tendo isso em mente, pode-se dizer que o resultado dessa etapa é utilizado como informação das características detectadas (*input*) para o próximo passo do processo. Mas apenas o resultado obtido pelo esboço 2,5D não permite que seja reconstruída a cena em 3D. Dessa forma, essa etapa é um meio caminho entre as informações em 2D e as em 3D e, em particular, nada pode ser dito a respeito do “outro lado” de qualquer objeto no campo de visão do observador. Mas, com o resultado desse passo, pode ser derivado uma superfície normal associada com cada superfície previamente detectada no esboço inicial, e pode ocasionar uma melhoria implícita na qualidade desta informação.

Existem várias formas de gerar o esboço 2,5D, mas as suas características comuns são a continuação da abordagem bottom-up em que eles não exploram qualquer conhecimento sobre o conteúdo da cena. Mas sim utilizam pistas adicionais, tais como o conhecimento da natureza da iluminação ou o efeito dos movimentos. E são, portanto, de aplicação geral e não de domínio específico. Das abordagens, a mais utilizada é a técnica conhecida como “*Shape from X*”. Ao final dessa etapa a representação ainda é em um sistema de coordenadas centrado no observador.

2.2.1.3 Representação 3D

Na representação 3D, o paradigma Marr coincide com uma aproximação top-down, passando para abordagens baseadas em modelos. Nessa etapa, são utilizados os resultados encontrados na tentativa de identificar os objetos contidos na cena. Isso só pode ser conseguido pelo conhecimento prévio do “objeto” que está na cena e, consequentemente, algum método para descrevê-lo. O ponto importante é que isso é uma translação para um sistema de coordenadas centrado no objeto, permitindo que a descrição do objeto seja independente do observador.

Essa é a etapa mais difícil e o sucesso de sua implementação é remota, especialmente comparado com o sucesso nos processos de geração dos esboços inicial e 2,5D. Uma boa especificação do que se busca tem auxiliado muito na orientação da área de pesquisa de visão computacional desde que o paradigma foi formulado. Ao contrário das etapas anteriores, nesta etapa existe pouca orientação a respeito da forma que pode ser utilizada para desenvolver algoritmos, uma vez que este nível de visão humana ainda não é bem compreendido. Marr observou que o sistema de coordenadas do objeto deve ser modular, no sentido que cada “objeto” deve ser tratado de forma diferente, ao invés de usar um sistema global de coordenadas (que geralmente é centrado no observador) e, desta forma, evita-se a necessidade de considerar a orientação das componentes do modelo em relação ao conjunto. Observa-se ainda que em um conjunto de volumes de formas primitivas é provável que tenha mais sucesso na representação dos modelos (em contraste com a técnica baseada na descrição das superfícies). Representações baseadas nos eixos naturais de um objeto, devido à simetria da forma ou a orientação das características, apresentam grandes chances de sucesso.

O paradigma Marr defende um conjunto de módulos relativamente independentes: O objetivo do módulo de baixo nível é recuperar descrições significativas da imagem desejada; O módulo de nível intermediário utiliza diferentes dicas como mudanças de intensidade, contornos, texturas, movimentos para tentar recuperar a forma ou a localização no espaço. Estudos posteriores (BERTERO; POGGIO; TORRE, 1988; ALOIMONOS; ROSENFELD, 1994) mostraram mais tarde que a maioria das tarefas do módulo de baixo nível e das tarefas do módulo de nível intermediário não são bem formuladas, assim não apresentam uma solução única. Um método popular de deixar as tarefas melhor formuladas é pela regularização

(TIKHONOV; ARSENIN, 1977; POGGIO; TORRE; KOCH, 1985). Limitações que exigem continuidade e suavidade na solução são muitas vezes adicionadas

2.2.2 Outros paradigmas de visão: visão ativa e proposital

Quando informações geométricas consistentes devem ser modeladas explicitamente (como, por exemplo, para a manipulação dos objetos) um sistema de coordenadas centrado no objeto pode ser apropriado. Não é certo que a tentativa de Marr de criar sistemas de coordenadas centradas no objeto é confirmada na visão biológica. Por exemplo, Koenderink (1990) mostrou que o espaço global da visão humana é um sistema centrado no observador e não-euclidiano. Para pequenos objetos, a existência de um quadro com referência centrada no objeto não foi comprovada por estudos psicológicos.

Atualmente existem duas escolas tentando explicar o mecanismo da visão:

- A primeira escola, e mais antiga, procura utilizar métricas explicitas de informações nas primeiras etapas do processo de visão (linhas, curvas, normais, etc.). A geometria é extraída em um processo típico de forma bottom-up sem nenhuma informação a respeito do processo da sua representação. Como resultado, obtém-se um modelo geométrico;
- A segunda escola, e mais nova, não extrai as informações métricas (geometrias) dos dados visuais até que seja necessário para uma tarefa específica. Os dados são coletados de forma sistemática para garantir que todas as características estão presentes nos dados, mas pode permanecer não interpretada até que uma tarefa específica seja incluída. Dessa forma, um banco de dados ou coleção de imagens intrínsecas (ou vistas) é modelado.

Muitos sistemas tradicionais de visão computacional e teorias capturam dados com câmeras que possuem características fixas. O mesmo ocorre com as teorias tradicionais, por exemplo, o observador de Marr é estático. Alguns pesquisadores defendem a visão por percepção ativa (BAJCSY, 1988; LANDY; MALONEY; PAVEL, 1996) e proposital (CVGIP B, 1992). Em um sistema de visão ativa, as características da aquisição são dinamicamente controladas pela interpretação da cena – muitas tarefas visuais tendem a ser mais simples se o observador é ativo e controla o sensor visual. O movimento controlado dos olhos (ou câmeras) é um exemplo, de onde o

observador registra a cena. Não existem dados suficientes para interpretar a cena. Ele pode observar a mesma cena por outro ângulo. Em outras palavras, a visão ativa é um mecanismo de aquisição de dados inteligentes controlados pelas medições parcialmente interpretadas dos parâmetros de uma cena e de seus erros obtidos. A visão ativa é uma área de pesquisas atuais.

A abordagem ativa pode tornar as tarefas de visão, que o observador não esteja bem posicionado, realizáveis. A Quadro 2.1 a mostra uma visão geral de como um observador ativo pode favorecer um observador mal posicionado:

Quadro 2.1 – Comparação entre um observador ativo e um passivo na tentativa de reconhecer uma cena

Tarefa	Observador Passivo	Observador Ativo
Formas por Sombras (Shape from Shading)	Mal posicionado. A normalização ajuda, mas uma solução única não é garantida devido as não-linearidades	Bem posicionado. Estável. Soluções únicas. Equações lineares
Formas por Contornos (Shape from Contour)	Mal posicionado. Soluções regularizadas ainda não formuladas. Soluções existem só para casos muito especiais	Bem posicionado. Solução única para um observador monocular ou binocular
Formas por Texturas (Shape from Texture)	Mal posicionado. Suposições a respeito da textura são necessários	Bem posicionado, sem a necessidade de suposições
Estrutura por Movimento (Structure from Motion)	Bem posicionado, mas instável	Bem posicionado e estável. Constantes quadráticas e de solução simples

Fonte: Autor

Geralmente, tem sido aceito pela comunidade de visão computacional que a recuperação da forma exata da imagem de intensidade é difícil. O paradigma de Marr é uma boa estrutura de trabalho teórica, mas infelizmente ele não obtém um desempenho com muitos sucessos para as aplicações de visão, como, por exemplo, reconhecimento e navegação.

Não existe uma teoria que fornece um modelo matemático (modelo computacional) que explica o “entendimento” dos aspectos da visão humana. Dois desenvolvimentos para uma nova teoria da visão são:

- Visão qualitativa – é um processo que busca a descrição qualitativa dos objetos ou da cena (ALOIMONOS, 1994). A motivação não é representar a geometria que não é necessária nos processos ou na decisão qualitativa (análise não geométrica). Além disso, as

informações qualitativas são mais invariantes para várias transformações indesejáveis (como pequena mudança do ponto de observação da cena) ou ruídos das informações quantitativas. A “qualitatividade” (ou invariância) permite a interpretação de eventos observados em diversos níveis de complexidade. É possível observar que os olhos humanos também não fornecem uma medida precisa, então um algoritmo de visão computacional deve buscar por informações qualitativas nas imagens, como por exemplo manchas (ou regiões) superficiais côncavas e convexas no intervalo de dados (BESL; JAIN, 1988).

- O paradigma da visão intencional – é um processo que pode ajudar a chegar a soluções mais simples para o modelamento do objeto em 3D (CVGIP B, 1992). A questão-chave é identificar o objetivo da tarefa. A motivação é diminuir o processo considerando apenas aquele pedaço da informação que é necessário. Mecanismos de evitar colisões em veículos autônomos é um exemplo no qual a descrição da forma precisa não é necessária. A aproximação pode ser heterogênea e a resposta qualitativa pode ser suficiente em alguns casos. O paradigma da visão intencional ainda não possui uma base teórica sólida, mas os estudos relacionados à visão biológica é uma fonte rica de inspiração. Essa mudança no objetivo das pesquisas tem resultado em sucesso para muitas aplicações de visão onde a descrição perfeita da geometria não é necessária. Como exemplo, pode-se citar os processos de evitar colisão em navegação de sistemas autônomos, rastreamento de objetos, etc. (HOWARTH, 1994; BUXTON; HOWARTH, 1995; FERNYHOUGH, 1997).

Entretanto, existem outras atividades de visão computacional que necessitam da obtenção de modelos 3D completos. Por exemplo, obter um modelo CAD 3D de um objeto real e digitalizar um modelo de barro gerado por um designer humano para obter efeitos especiais em filmes.

2.3 Conceitos Básicos da Geometria Projetiva

O campo da visão computacional tem desenvolvido rapidamente. Pode-se citar, como exemplo, a identificação da geometria utilizando múltiplas vistas do objeto. A matemática utilizada quando envolve geometria utilizando múltiplas vistas do objeto é a relação entre:

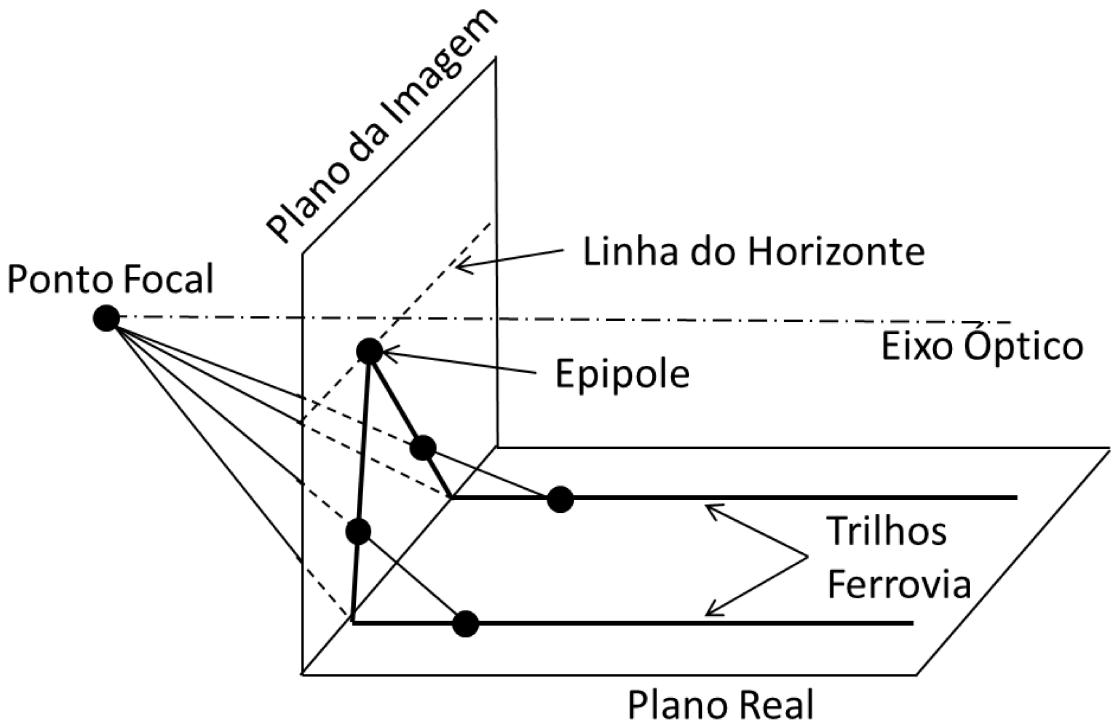
- Pontos 3D de uma cena. De forma mais geral, as linhas e os outros objetos da geometria simples;
- A projeção da câmera ao registrar a cena e;
- A relação entre a projeção de várias câmeras em uma cena.

A **fotogrametria** mede as distâncias 3D das fotografias. Os métodos fotogramétricos típicos utilizam câmeras caras e especiais calibradas precisamente e os pontos na imagem são medidos manualmente com alta precisão. A fotogrametria considera uma classe bastante limitada de tarefas. Em contraste, o objetivo da geometria de múltiplas vistas na visão computacional 3D é utilizar câmeras comuns, que são calibradas parcialmente ou não calibradas em trabalhos com imagens com grandes imprecisões de medidas e utilizando algoritmos automáticos para obter os resultados. Desenvolvimentos recentes permitem a automação total de tarefas como a reconstrução 3D dos pontos observados pela câmera de uma imagem de vídeo sem informações de posição e calibração (http://www.2d3.com) ou a reconstrução automática em 3D de uma cena utilizando um número elevado de diferentes vistas (CORNELIUS et al., 2004). O processo de reconstrução geométrica usando múltiplas vistas tem sido pesquisada e apresentada suas tendências em diversos livros (FAUGERAS, 1993; HARTLEY; ZISSERMAN, 2003; MA et al., 2004).

A base dos cálculos matemáticos para obter a geometria usando múltiplas vistas é a **geometria projetiva**. O sensor básico que provê a visão computacional com informações sobre o ambiente 3D são as imagens estáticas ou os vídeos capturados pela câmera. Este trabalho apresenta os aspectos geométricos e como utilizar as imagens com informações 2D para a medição automática em 3D. As medições das coordenadas 3D dos pontos ou as distâncias a partir das imagens 2D são importantes nesta tese. Para isso é necessário compreender **projeção perspectiva** (também conhecida como **projeção central**). E na projeção descreve a formação das imagens nas câmeras. Nesse processo, as linhas paralelas do mundo real não permanecem paralelas nas imagens da projeção perspectiva. Como exemplo,

pode ser citado imagens de ferrovias muito longas em que os trilhos, apesar de serem paralelos, não apresentam mais essa característica convergindo para um ponto na imagem. A Figura 2.5 mostra como a projeção perspectiva converte as linhas paralelas de uma ferrovia do mundo real em linhas não paralelas na imagem.

Figura 2.5 – Efeito da projeção projetiva da câmera em trilhos de uma ferrovia



Fonte: Autor

2.3.1 Pontos e hiperplanos no espaço projetivo

Para um melhor entendimento nesta seção será apresentado de forma concisa a introdução da notação básica e definições do espaço projetivo (SEMPLE; KNEEBONE, 1998; MOHR, 1993). Considere um espaço linear de dimensão $(d+1)$ sem sua origem em $\mathcal{R}^{d+1} - \{[0,\cdots,0]^T\}$ e definida a relação de equivalência:

$$[\mathrm{x}_1,\cdots,\mathrm{x}_{\mathrm{d}+1}]^{\mathrm{T}} \cong [\mathrm{x}'_1,\cdots \mathrm{x}'_{\mathrm{d}+1}]^{\mathrm{T}} \Leftrightarrow \exists\ \mathrm{a} \neq 0 \colon [\mathrm{x}_1,\cdots,\mathrm{x}_{\mathrm{d}+1}]^{\mathrm{T}} \cong \mathrm{a}[\mathrm{x}'_1,\cdots \mathrm{x}'_{\mathrm{d}+1}]^{\mathrm{T}} \quad (2.3)$$

Isso significa que dois vetores em $\mathcal{R}^{d+1}$ são equivalentes se um é múltiplo escalar do outro, onde esse escalar é diferente de zero $(a \neq 0)$. O espaço projetivo $\mathcal{P}^d$ é o quociente dessa relação. Podemos imaginar o espaço projetivo como sendo um conjunto de linhas em $\mathcal{R}^{d+1}$ em que todas elas passam pela origem.

Dessa forma, qualquer ponto em $\mathcal{P}^d$ irá corresponder a um conjunto infinito de vetores paralelos em $\mathcal{R}^{d+1}$ e é único caso seja conhecido um vetor em $\mathcal{R}^{d+1}$. O vetor em $\mathcal{R}^{d+1}$ que forma o ponto de interesse em $\mathcal{P}^d$ é conhecido como representação homogênea (e, também, de projetiva) do ponto em $\mathcal{P}^d$. Um vetor homogêneo representa o mesmo como qualquer outro vetor em que a diferença entre eles é a multiplicação por um escalar diferente de zero. Esse escalar é escolhido, normalmente, para que o vetor em $\mathcal{R}^{d+1}$ tenha o valor 1 (unidade) na sua posição mais à direita (posição $d+1$), por exemplo $[x_1', \cdots x_d', 1]^T$. Os vetores homogêneos serão representados em negrito, como $\boldsymbol{x}$.

Estamos mais acostumados com o sistema de coordenadas de pontos, o sistema Cartesiano convencional (também conhecido como sistema de coordenadas não homogêneas). Os pontos do sistema de coordenadas Cartesiano fornecem pontos no espaço Euclidiano d-dimensional $\mathcal{R}^d$ que ocupam o plano com a equação $x_{d+1} = 1$ em $\mathcal{R}^{d+1}$. O mapeamento de vetores não-homogêneos de $\mathcal{R}^d$ para $\mathcal{P}^d$ é dado por:

$$[\mathrm{x}_1, \cdots, \mathrm{x}_\mathrm{d}]^\mathrm{T} \longrightarrow [\mathrm{x}_1, \cdots, \mathrm{x}_\mathrm{d}, 1]^\mathrm{T} \qquad (2.4)$$

Os pontos $[x_1, \cdots, x_d, 0]^T$ não apresentam contrapartida no espaço Euclidiano, mas representam pontos no infinito em uma direção em particular. Considere $[x_1, \cdots, x_d, 0]^T$ como um caso limite de $[x_1, \cdots, x_d, a]^T$ que projetivamente equivalente a $[x_1/a, \cdots, x_d/a, 1]^T$ e que podemos assumir que $a \to 0$. Isso corresponde a um ponto em $\mathcal{R}^d$ indo para o infinito na direção do vetor radial $[x_1/a, \cdots, x_d/a]^T \in \mathcal{R}^d$.

Tendo isso em mente, pode-se compreender as **coordenadas homogêneas** de hiperplanos em $\mathcal{P}^d$. Um hiperplano em $\mathcal{P}^d$ é representado pelo vetor de dimensão $(d+1)$ $\mathbf{a} = [a_1, \cdots, a_{d+1}]^T$ tal que todos os pontos de **x** pertencentes ao hiperplano satisfaz a equação $\mathbf{a}^T \odot \mathbf{x} = 0$ (em que $\mathbf{a}^T \odot \mathbf{x}$ representa o produto escalar). Considerando os pontos na forma $\mathbf{x} = [x_1, \cdots, x_d, 0]^T$ nos leva à formula familiar de $a_1 x_1 + \cdots + a_d x_d + a_{d+1} = 0$.

Com isso temos que o hiperplano é definido por d pontos distintos representados por vetores $\mathbf{x}_1, \cdots, \mathbf{x}_d$ pertencentes ao hiperplano que é representado por **a** e o vetor **a** é ortogonal aos vetores $\mathbf{x}_1, \cdots, \mathbf{x}_d$. Esse vetor **a** pode ser computado de diversas formas, como por exemplo, através da Decomposição de Valores Singulares (SVD – Singular Value Decomposition). Simetricamente, os pontos de

intercessão de d hiperplanos distintos $\mathbf{a}_1, \cdots, \mathbf{a}_d$ é o vetor $\mathbf{x}$ que é ortogonal aos hiperplanos.

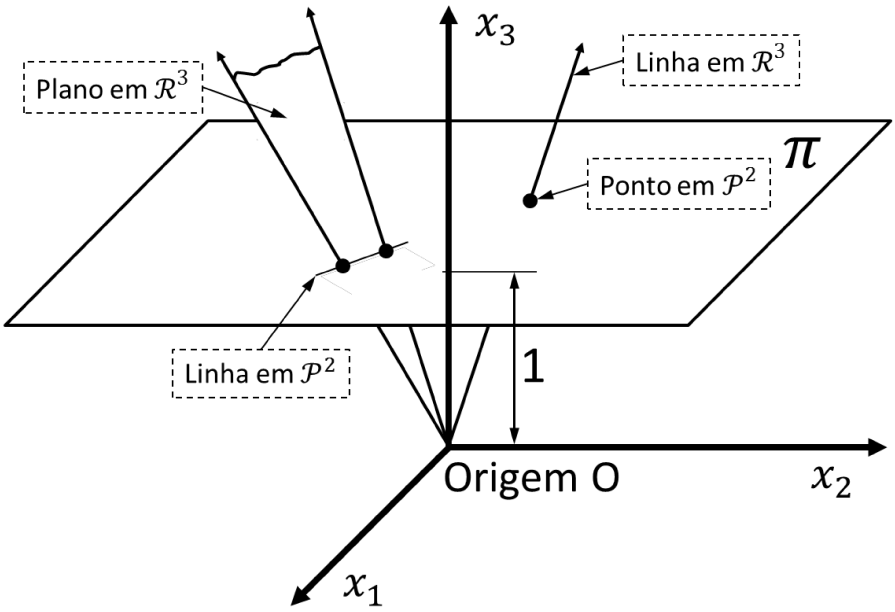
Existem dois casos particulares e de interesse para a visão computacional:

1. O Plano Projetivo $\boldsymbol{\mathcal{P}^2}$. Para a padronização das abreviaturas, os pontos em $\mathcal{P}^2$ serão representados pela letra u em minúsculo, como u $= [u, v, w]^T$, linhas em $\mathcal{P}^2$ pela letra l em minúsculo. Em ^{2} é possível utilizar o produto vetorial para as operações de união e interseção: a linha que passa pelos pontos x e y é representada por $\mathbf{l} = \mathbf{x} \otimes \mathbf{y}$ e o ponto de interseção das duas linhas l e m é $\mathbf{x} = \mathbf{l} \otimes \mathbf{m}$, em que o símbolo $\otimes$ representa o produto vetorial.
2. O Espaço Projetivo $\boldsymbol{\mathcal{P}^3}$. Para a padronização das abreviaturas, os pontos em $\mathcal{P}^3$ serão representados por letras em maiúsculo como X $=$ $[X, Y, Z, W]^T$. Em $\mathcal{P}^3$ hiperplanos tornam-se planos e mais uma entidade ocorre que não tem contrapartida no plano projetivo: a linha 3D. Uma representação homogênea e elegante utilizando um vetor de quatro dimensões, disponível para pontos e planos em $\mathcal{P}^3$, não existe para as linhas. Uma linha 3D pode ser representada tanto por um par de pontos que pertence à linha quanto por uma matriz (Grassmann-)Plücker (HARTLEY; ZISSERMAN, 2003).

A Figura 2.6 apresenta graficamente um modo que pode ser usado para imaginar o espaço projetivo $\mathcal{P}^2$ como linhas em $\mathcal{R}^3$.

Ainda na Figura 2.6, o plano π é formado pela equação $x_3 = 1$. Uma linha em $\mathcal{R}^3$ corresponde a um único ponto em $\mathcal{P}^2$. Um plano em $\mathcal{R}^3$ que passa pela origem $\boldsymbol{O}$ corresponde a uma linha em $\mathcal{P}^2$.

Figura 2.6 – Ilustração do espaço projetivo



$\mathcal{P}^2$. Pontos e linhas em $\mathcal{P}^2$ são representados, respectivamente, por linhas e planos que passam pela origem de um sistema espacial euclidiano $\mathcal{R}^3$
Fonte: Autor

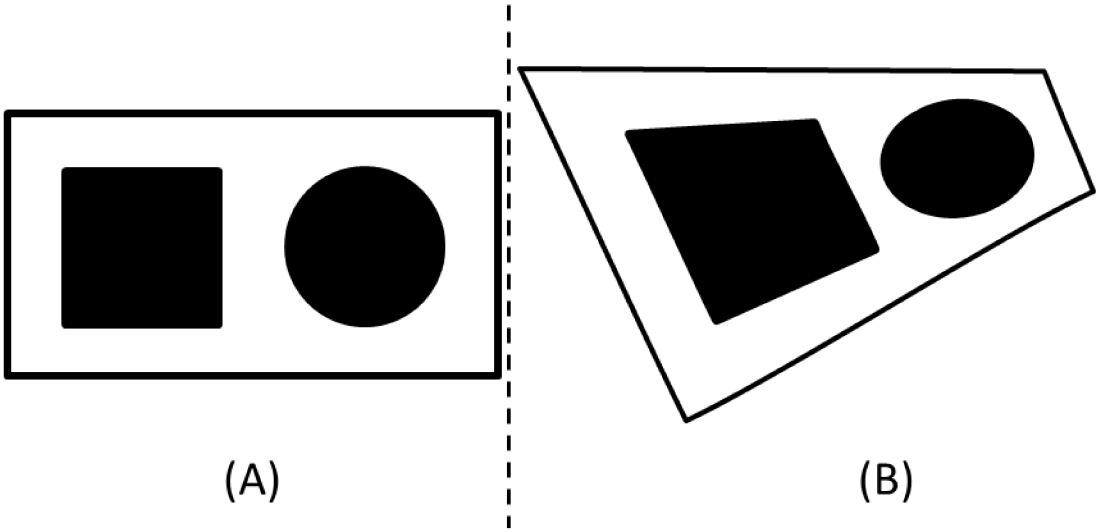
2.3.2 Homografia

Uma **Homografia**, também conhecida como **Colineação** ou **Transformação Projetiva**, é qualquer mapeamento $\mathcal{P}^d \longrightarrow \mathcal{P}^d$ que é linear dentro do espaço $\mathcal{R}^{d+1}$. Isso é, uma homografia é dada por uma escala desconhecida que pode ser escrita como:

$$\mathrm{u}' \cong \mathrm{Hu} \tag{2.5}$$

Em que H é uma matriz $(d+1) \otimes (d+1)$. A transformação mapeia qualquer trio de pontos colineares em outro trio de pontos colineares (daí a origem do nome Colineação). Se H não é singular, então pontos distintos são mapeados em pontos distintos. Um exemplo de uma imagem 2D mapeada com a homografia é apresentado na Figura 2.7.

Figura 2.7 – Exemplo da transformação projetiva

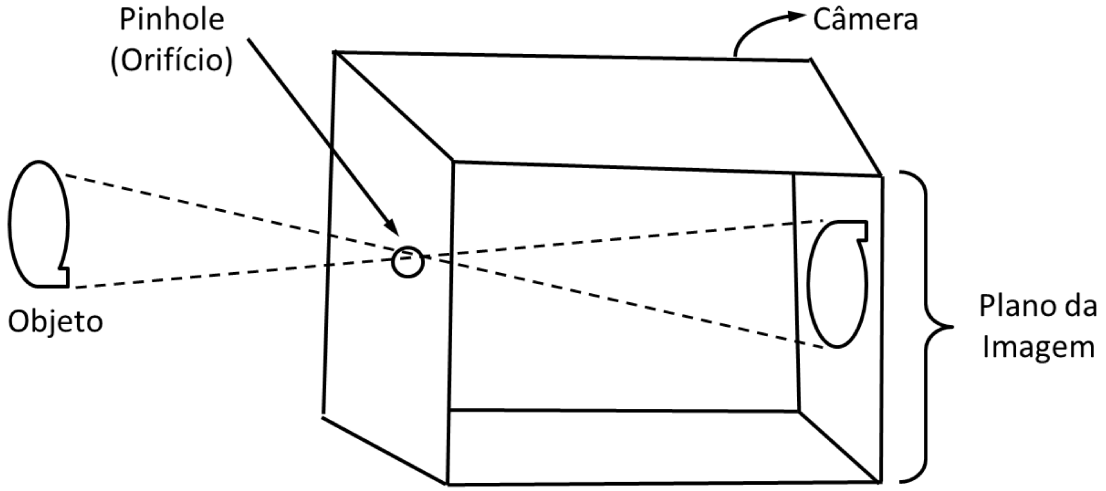


A imagem (B) é uma transformação projetiva da imagem (A)
Fonte: Autor

A transformação projetiva de hiperplanos tem uma forma diferente que a de pontos. Isso pode ser demonstrado do fato que se o ponto original $\mathbf{u}$ e um hiperplano $\mathbf{a}$ são incidentes, $\mathbf{a}^T \cdot \mathbf{u} = \mathbf{0}$, então, após a transformação projetiva, eles devem se manter incidentes após a transformação, $\mathbf{a}'^T \cdot \mathbf{u}' = \mathbf{0}$. Utilizando a Equação (2.5) obtemos $\mathbf{a}' \simeq H^{-T} \cdot \mathbf{a}$, H^{-T} é a matriz trasposta e inversa de H.

Na visão computacional existem dois casos simples em que a homografia acontece. Primeiro, uma projeção de uma cena plana por uma câmera formada por uma câmara negra que possui um furo para a projeção da cena (como mostra na Figura 2.8) é relacionada por uma homografia 2D. Isso pode ser usado para retificar imagens de uma cena planar (como exemplo capturar a face de um edifício). Segundo, duas imagens de uma cena 3D (plana ou não plana) por duas câmeras de câmara escura que compartilham um único centro de projeção é uma homografia 2D. Isso pode ser usado para juntar imagens de fotografias panorâmicas.

Figura 2.8 – Modelo de uma Câmera com Orifício (Modelo Pinhole)



Fonte: Autor

Para deixar a notação homogênea mais familiar, será apresenta como um ponto 2D não homogêneo $[u, v]^T$ (como um ponto em uma imagem) é realmente mapeado a um ponto $[u', v']^T$ não homogêneo em uma imagem, sendo essa transformação realizada por H através da Equação (2.5). Com os componentes e as escalas escritas explicitamente, a equação torna-se:

$$\alpha \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2.6)$$

Ao escolhermos o valor 1 para a última coordenada de u' nós podemos assumir que u' não é um ponto no infinito, se $\alpha \neq 0$. Para calcularmos $[u' \quad v']^T$ é necessário eliminarmos a escala α. Isso nos leva a equação de u' e v' sendo:

$$u' = \frac{h_{11} \cdot u + h_{12} \cdot v + h_{13}}{h_{31} \cdot u + h_{32} \cdot v + h_{33}} \quad (2.7)$$

$$v' = \frac{h_{21} \cdot u + h_{22} \cdot v + h_{23}}{h_{31} \cdot u + h_{32} \cdot v + h_{33}} \quad (2.8)$$

As Equações (2.7) e (2.8) são familiares para trabalhos em processamento de imagem que não utilizam coordenadas homogêneas. É possível notar que ao comparar as Equações (2.7) e (2.8) com a Equação (2.5) a Equação (2.5) é mais simples, linear e permite trabalhar no caso em que o ponto u' é um ponto no infinito. Essas são as vantagens práticas de utilizar as coordenadas homogêneas.

2.3.2.1 Subgrupos da homografia

Além da colinearidade e das relações de tangências, outra invariância bem conhecida da transformação projetiva é a relação de cruzamento entre linhas. O grupo

de transformações projetivas contém importantes subgrupos: transformação afim, transformação de similaridade e transformação métrica (também chamada de transformação Euclidiana). Um resumo das transformações pode ser visto no Quadro 2.2.

Também existem outros subgrupos, mas a Quadro 2.2 apresenta aqueles encontrados frequentemente na visão computacional. Os subgrupos são dados ao impormos as restrições na forma de H.

Qualquer homografia pode ser unicamente composta como $H = H_P \cdot H_A \cdot H_S$, onde:

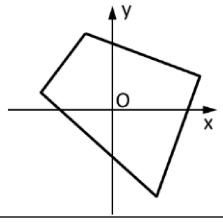
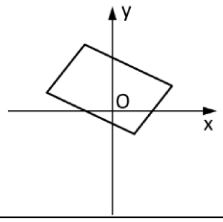
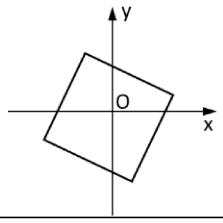
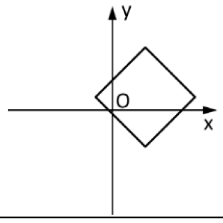
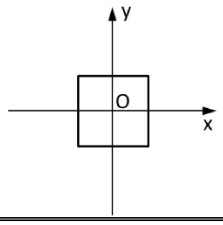
$$\mathrm{H_P} = \begin{bmatrix} \mathrm{I} & 0 \\ \mathrm{a^T} & \mathrm{b} \end{bmatrix} \tag{2.9}$$

$$\mathrm{H_A} = \begin{bmatrix} \mathrm{K} & 0 \\ 0^{\mathrm{T}} & 1 \end{bmatrix} \tag{2.10}$$

$$\mathrm{H_S} = \begin{bmatrix} \mathrm{R} & -\mathrm{Rt} \\ 0^{\mathrm{T}} & 1 \end{bmatrix} \tag{2.11}$$

A matriz K deve ser triangular superior. As matrizes de H_S representam as transformações Euclidiana. A matriz formada por $H_A \cdot H_S$ representa as transformações afins, pois a matriz H_A representa o subgrupo das transformações afim, a transformação “puramente afim”. Para obter a transformação “puramente afim” basta remover a parte da transformação Euclidiana (H_S). A matriz formada por $H_P \cdot H_A \cdot H_S$ representam todo o grupo de transformações projetivas, dessa forma a matriz H_P representa o subgrupo da projeção “puramente projetiva”.

Quadro 2.2 – Subgrupos de transformações projetivas frequentemente encontradas em visão computacional

Nome da transformação	Restrições ou função de transformação (H)	Exemplo em 2D	Invalidações
Projetiva	$\det(H) \neq 0$		Colinearidade Tangencia Relação de Cruzamento de Linhas
Afim	$H = \begin{bmatrix} A & t \\ \mathbf{0}^T & 1 \end{bmatrix}$ $\det(A) \neq 0$		Invariâncias da Projetiva e mais: Paralelismo Relação de comprimento de retas paralelas Razão de Áreas Combinação Linear dos Vetores Centroide
Similaridade	$H = \begin{bmatrix} sR & -Rt \\ \mathbf{0}^T & 1 \end{bmatrix}$ $R^T \cdot R = I$ $\det(R) = 1$ $s > 0$		Invariâncias da Afim e mais: Ângulos Razão de comprimentos
Métrica (ou Euclidiana ou Isométrica)	$H = \begin{bmatrix} R & -Rt \\ \mathbf{0}^T & 1 \end{bmatrix}$ $R^T \cdot R = I$ $\det(R) = 1$		Invariâncias da Similaridade mais: Comprimento Área (volume)
Identidade	$H = I$		Caso trivial, tudo é invariante

Fonte: Autor

Em relação às decomposições, o único paço não trivial é a decompor uma matriz A genérica nos produtos de uma matriz triangular superior K e uma matriz rotação R, através da rotação de matriz, ou seja, encontrar uma matriz que seja ortonormal ($R^T \cdot R$) e não seja reflexiva ($\det(R) = 1$). Isso pode ser obtido através da decomposição RQ, análogo a decomposição QR (PRESS et al., 1992; GOLUB; VAN LOAN, 1989).

2.3.3 Estimando a homografia entre pontos correspondentes

Uma tarefa frequente em visão computacional 3D é computar a homografia das imagens ao comparar pontos correspondentes em mais de uma imagem. Por pontos correspondentes entenda que são um conjunto de pontos ordenados $\{[u_i, u_i']\}_{i=1}^{m}$ tal que cada par sejam correspondentes após as transformações homogêneas. Os pontos correspondentes podem ser obtidos tanto por forma manual ou cálculos computacionais seguindo um algoritmo.

Para que seja possível calcular e encontrar H, precisamos resolver o seguinte sistema linear homogêneo de equações e encontrar H e as escalas a_{i}:

$$\mathrm{a_i} \cdot \mathrm{u_i'} = \mathrm{H} \cdot \mathrm{u_i} \tag{2.12}$$

em que $\mathrm{i} = 1, \cdots, \mathrm{m}$.

Esse sistema possui $m(d+1)$ equações e $m + (d+1)^2 - 1$ incógnitas. Há m incógnitas a_{i}, $(d+1)^2$ componentes em H, enquanto que $(d+1)^2 - 1$ informações é suficiente para encontrarmos todos os elementos de H até um fator de escala global. Dessa forma, temos que $m = d + 2$ correspondencias são necessárias para determinar H de forma única.

Em determinadas ocasiões, a correspondência de uma configuração degenerada significa que H não pode ser encontrado de forma única, mesmo que $m \geq d + 2$. Uma configuração é não degenerada se nenhum dos d pontos de u_i estiver localizado em um único hiperplano e nenhum dos d pontos de u_i' estiver em um único hiperplano.

Quando mais de $d + 2$ pontos correspondentes estiverem disponíveis, o sistema da Equação (2.12) tem nenhuma solução geral devido ao ruído na medição das correspondências. Assim, a forma mais fácil que seria para solucionar o sistema linear torna-se a mais difícil da estimativa ótima de parâmetros de um modelo paramétrico. Assim, não se soluciona a Equação (2.12), mas define-se um critério mínimo adequado, derivado de considerações estatísticas.

Os métodos de estimativa que serão apresentados não estão restritos a homografia, são métodos genéricos aplicados sem modificações conceituais a outros problemas de visão computacional 3D. Dentre os outros métodos podem ser citados: a recessão de câmera, a triangulação, a obtenção da estimativa da matriz fundamental e a obtenção do tensor trifocal.

2.3.3.1 Estimação pela máxima verossimilhança (maximum likelihood – ML)

A abordagem estatística ideal é a Estimação pela Máxima Verossimilhança (ML). Consideremos o caso em que $d = 2$ – estimativa homográfica usando duas imagens como as mostradas na Figura 2.7. Para a estimação dos pontos, pode-se assumir que cada um dos pontos não homogêneos é uma variável aleatórias com distribuição normal e independentes, possuem o valor médio de $[\hat{u}_i, \hat{v}_i]^T$ e $[\hat{u}'_i, \hat{v}'_i]^T$ e, também, possuem a mesma variância. Esta hipótese geralmente leva a bons resultados práticos. Pode ser mostrado que a estimativa por ML leva à minimização de erros de reprojeção de aproximações quadráticas. Ou seja, tem-se que calcular o seguinte processo de minimização com restrições nas $9 + 2m$ variáveis:

$$\min_{H,u_i,v_i} \sum_{i=1}^{m} \left[(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2 + \left(\frac{[u_i, v_i, 1]\boldsymbol{h_1}}{[u_i, v_i, 1]\boldsymbol{h_3}} - \hat{u}'_i \right)^2 + \left(\frac{[u_i, v_i, 1]\boldsymbol{h_2}}{[u_i, v_i, 1]\boldsymbol{h_3}} - \hat{v}'_i \right)^2 \right] \quad (2.13)$$

Em que $\boldsymbol{h_i}$ corresponde a i-ésima linha da matriz H, ou seja, $\boldsymbol{h_1^T}u/\boldsymbol{h_3^T}u$ e $\boldsymbol{h_2^T}u/\boldsymbol{h_3^T}u$ são as coordenadas não homogêneas de um ponto $\boldsymbol{u}$ mapeado por H dado pela Equação (2.6). A função minimizada pela Equação (2.13) é o erro de reprojeção.

O processo descrito pela Equação (2.13) é uma operação não linear, não convexa e normalmente apresenta vários mínimos locais. Um bom local de mínimos (em geral, mas não global) pode ser encontrado seguindo dois passos:

1. Uma estimativa inicial é calculada através da resolução de um problema de minimização estatisticamente não ideal, mas muito mais simples com um único mínimo local;
2. O mínimo local mais próximo da solução por ML (solução ótima) é calculado por um algoritmo de minimização de locais.

Para a solução desse problema de não-linearidade pode-se recorrer ao algoritmo Levenberg-Marquardt (PRESS et al., 1992).

2.3.3.2 Estimação linear

Para encontrar uma boa estimativa, porém não estatisticamente ótima, pode-se resolver o sistema da Equação (2.12) pelo método utilizado em álgebra linear para solucionar sistemas lineares com mais equações do que incógnitas, minimizando a distância algébrica, também conhecida como Transformação Linear Direta (DLT – Direct Linear Transformation) ou simplesmente Estimação Linear. Normalmente a

técnica da DLT fornece resultados satisfatórios, mesmo sem a aplicação de métodos não lineares na sequência.

Para os pontos representado em coordenadas homogêneas, $\boldsymbol{u} = [u, v, w]^T$, pode-se rearranjar a Equação (2.12) em uma forma mais adequada para que a solução possa ser feita manipulando os componentes manualmente. Entretanto, é possível utilizar duas estratégias que permitam que o equacionamento se mantenha na forma de matrizes.

Primeiramente, para eliminarmos a de $a \cdot \boldsymbol{u}' = H \cdot \boldsymbol{u}$ multiplica-se a equação pela matriz $G(\boldsymbol{u}')$, tal que as suas linhas sejam ortogonais à $\boldsymbol{u}'$. Isso faz com que o lado esquerdo da equação seja nulo, pois $G(\boldsymbol{u}') \cdot \boldsymbol{u}' = 0$ e obtemos $G(\boldsymbol{u}') \cdot H \cdot \boldsymbol{u} = 0$. Se os pontos da imagem possuem a terceira coordenada igual a 1 ($w' = 1$), então as coordenadas dos pontos podem ser expressas como $[u', v', 1]^T$, a matriz $G(\boldsymbol{u}')$ pode ser escolhida como:

$$\mathrm{G(u)} = \mathrm{G([u, v, 1]^T)} = \begin{bmatrix} 1 & 0 & -\mathrm{u} \\ 0 & 1 & -\mathrm{v} \end{bmatrix} = [\mathrm{I} \quad -\mathrm{u}] \tag{2.14}$$

Essa escolha não é adequada caso algum ponto da imagem tenha $w' = 0$, pois, assim, $G(\boldsymbol{u}')$ irá se tornar uma matriz singular se $u' = v'$. Isso pode acontecer se os pontos não são medidos diretamente na imagem, mas sim calculado indiretamente (como o caso dos pontos que se estendem ao infinito como apresentado na Figura 2.5) e, consequentemente, alguns deles podem estar no infinito. Outra escolha, que funciona em situações genéricas, é $G(\boldsymbol{u}) = S(\boldsymbol{u})$, onde:

$$\mathrm{S(u)} = \mathrm{S([u, v, 1]^T)} = \begin{bmatrix} 0 & -\mathrm{w} & \mathrm{v} \\ \mathrm{w} & 0 & -\mathrm{u} \\ -\mathrm{v} & \mathrm{u} & 0 \end{bmatrix} \tag{2.15}$$

Assim, temos que o produto vetorial $\boldsymbol{u} \otimes \boldsymbol{u}' = S(\boldsymbol{u}) \cdot \boldsymbol{u}'$, para qualquer $\boldsymbol{u}$ e $\boldsymbol{u}'$.

Na sequência, para rearranjar a equação $G(\boldsymbol{u}') \cdot H \cdot \boldsymbol{u} = 0$ tal que as incógnitas fiquem a o máximo à direita o possível, utiliza-se a identidade $ABc = (\mathbf{c}^T \circledast \boldsymbol{A})\mathbf{b}$ (LÜTKEPOHL, 1997), em que $\mathbf{b}$ é um vetor construído pelas entradas da matriz B empilhada em colunas de primeira ordem e $\circledast$ corresponde ao produto Kronecker de matrizes. Aplicando isso temos:

$$\mathrm{G(u')} \cdot \mathrm{H} \cdot \mathrm{u} = [\mathrm{u^T} \circledast \mathrm{G(u')}]\mathrm{h} = 0 \tag{2.16}$$

Onde h corresponde ao vetor de 9 posições $[h_{11}, h_{21}, \cdots, h_{23}, h_{33}]^T$ das entradas da matriz H. Para $\mathrm{G(u')} = S(\mathrm{u'})$, em componente, temos:

$$\begin{bmatrix} 0 & -uw' & uv' & 0 & -vw' & -vv' & 0 & -ww' & -wv' \\ uw' & 0 & -uu' & uw' & 0 & -uu' & ww' & 0 & -wu' \\ -uv' & -uu' & 0 & -vv' & vu' & 0 & -wv' & wu' & 0 \end{bmatrix} \mathbf{h} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.17)$$

Considerando todas as m possíveis considerações, nos leva à:

$$\begin{bmatrix} \mathrm{u}_1^{\mathrm{T}} \circledast \mathrm{G}(\mathrm{u}_1') \\ \mathrm{u}_2^{\mathrm{T}} \circledast \mathrm{G}(\mathrm{u}_2') \\ \cdots \\ \mathrm{u}_{\mathrm{m}}^{\mathrm{T}} \circledast \mathrm{G}(\mathrm{u}_{\mathrm{m}}') \end{bmatrix} \mathrm{h} = 0 \quad (2.18)$$

Chamando a matriz $3m \times 9$ do lado esquerdo de W, podemos ler como $W \cdot \mathbf{h} = 0$. Esse sistema, na maioria dos casos, possui mais equações do que incógnitas e não é possível obter uma resposta. Utilizando a Decomposição em Valores Singulares (SVD – Singular Value Decomposition) é possível calcular um vetor $\mathbf{h}$ que minimiza $\|\boldsymbol{W}\mathbf{h}\|$ chegando a $\|\mathbf{h}\| = 1$.

Em detalhes, $\mathbf{h}$ é a coluna da matriz V na SVD da decomposição $W = U \cdot D \cdot V^T$ associada com o menor valor singular. Alternativamente, podemos calcular $\mathbf{h}$ como o autovetor de $W^T \cdot W$ associado com o menor autovalor. Isso é apresentado como sendo numericamente ligeiramente menos preciso do que SVD, mas tem a vantagem que a matriz $W^T \cdot W$ é apenas 9×9, enquanto W é $3m \times 9$. Ambos os casos funcionam relativamente bem na prática.

Para obter um resultado significativo, as componentes dos vetores $\boldsymbol{u}_i$ e $\boldsymbol{u}_i'$ não podem ter muita diferença de magnitude. Magnitudes semelhantes garantem que o mínimo obtido pela minimização algébrica da distância é razoavelmente próximo da solução da Equação (2.13). Magnitude similar pode ser assegurada por um tipo de pré-condição, conhecida no cálculo numérico e utilizada na visão computacional como **normalização** (HARTLEY, 1997). Assim, ao invés de utilizarmos a Equação (2.12), pode ser utilizado o sistema de equações $\overline{\boldsymbol{u}}_i \simeq \overline{H} \cdot \overline{\boldsymbol{u}}_i$ onde podemos substituir $\overline{\boldsymbol{u}}_i = H_{pre} \cdot \boldsymbol{u}_i$ e $\overline{\boldsymbol{u}}_i' = H_{pre}' \boldsymbol{u}_i'$. A homografia H é, então, recuperada como $H = H_{pre}'^{-1} \cdot \overline{H} \cdot H_{pre}$. As homografias pré-condicionadas H_{pre} e H_{pre}' são escolhidas tal que as componentes de $\overline{\boldsymbol{u}}_i$ e $\overline{\boldsymbol{u}}_i'$ possuam magnitudes similares. Assumindo que os pontos originais possuem a forma $[u, v, 1]^T$, uma escolha adequada é o escalonamento e translação anisotrópico dado na forma:

$$\overline{\mathrm{H}} = \begin{bmatrix} \mathrm{a} & 0 & \mathrm{c} \\ 0 & \mathrm{b} & \mathrm{d} \\ 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

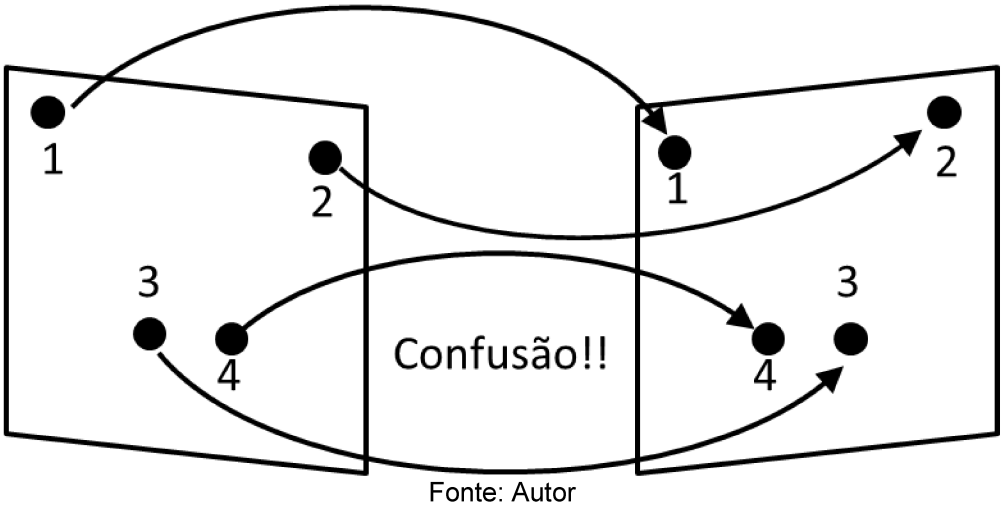
Onde a, b, c, d são tais que a média dos pontos pré-condicionados $\overline{\boldsymbol{u}} = [\bar{u}, \bar{v}, \overline{w},]^T$ seja 0 e a variância seja 1.

É notável a diferença entre o tamanho da solução ótima dada pela Equação (2.13), devido à estimativa por ML, e a solução por linearização, Equação (2.18). Enquanto que a primeira apresenta $9 + 2m$ variáveis, a última apresenta apenas 9 a serem encontradas. Para valores grandes de m há uma diferença considerável no custo computacional. Entretanto, a aproximação ela Equação (2.13) nos fornece a melhor aproximação e é usada na prática. Ainda existem aproximações que permitam diminuir o custo computacional, mas, ainda, não possuem um resultado próximo ao ótimo, como o caso do algoritmo da distância de Sampson (HARTLEY; ZISSERMAN, 2003).

2.3.3.3 Estimação robusta

Normalmente, é assumido que as medidas correspondentes são corrompidas por um Ruído Gaussiano Aditivo. Se o resultado apresenta erros grosseiros, como trocar a posição de pontos (descasamento), como mostrado na Figura 2.9, esse modelo estatístico não consegue mais correlacionar o problema e, com isso, muitos métodos podem levar a resultados sem sentido, como o que está mostrado na Figura 2.9.

Figura 2.9 – Correspondência enganosa de pontos

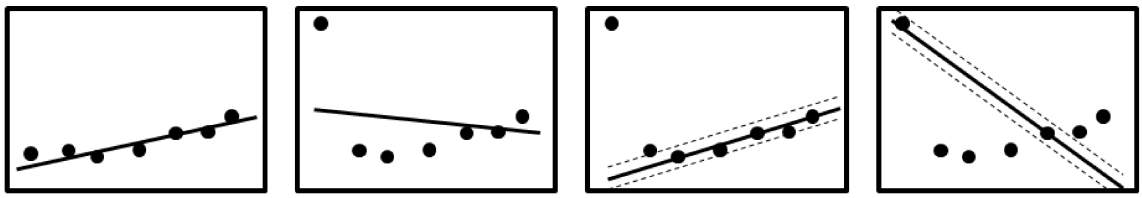


Fonte: Autor

Um exemplo simples é de uma linha pertencente a um plano. Se os pontos são corrompidos por um Ruído Gaussiano Aditivo, então a linha que minimiza a soma dos quadrados das distâncias (equação da regerão linear) dos pontos constituem um bom resultado, como mostrado na Figura 2.10A. Entretanto, se um ou mais pontos estão completamente errados, então minimizando utilizando o mesmo critério leva a

um resultado ruim, como apresentado na Figura 2.10B, pois o ponto distante pode fornecer um efeito arbitrariamente grande na posição da linha. Os resultados ruins são esperados pois estimador de mínimos quadrados é derivado do pressuposto de que o ruído é gaussiano, assim, os nossos dados violam esse modelo. Intuitivamente, a melhor aproximação seria ser ignorar os pontos distantes e utilizar os pontos remanescentes para alocar a linha.

Figura 2.10 – Influência de um ponto discrepante na aproximação por mínimos quadráticos



A – Os dados estão bem comportados e o modelo linear consegue ter um bom resultado; B – Um dos pontos está bem afastado dos outros o que leva a um grande erro no modelo linear quando se usa a aproximação por mínimos quadráticos; C – Caso seja conhecido o ponto discrepante a remoção dele permite o sucesso do modelo linear por mínimos quadráticos (princípio do algoritmo RANSAC); D – Uma avaliação errada do ponto discrepante e a remoção dos outros utilizando um limiar levando a um resultado errado (princípio do algoritmo RANSAC)
Fonte: Autor

Pontos que pertençam e, também, que não pertençam a um modelo de ruído escolhido são chamados de condizente (ou, também, inliers) e discrepantes (ou, também, outliers), respectivamente. Projetos de estimadores insensíveis a pontos discrepantes é a função da **Estatística Robusta**

Estimadores Robustos bem conhecidos são o de Mediana e o M-estimadores. Entretanto, para o condicionamento robusto dos modelos paramétricos utilizados em visão computacional o algoritmo RANSAC tornou-se o método padrão (FISCHLER; BOLLES, 1981).

2.3.4 O algoritmo RANSAC (montagem via consenso de amostras aleatórias)

Suponha que possuímos alguns dados que sabemos ser linearmente relacionada. É razoável e habitual encontrarmos a relação linear através de alguma abordagem de mínimos quadrado, minimizando a soma dos quadrados dos resíduos ao quadrado. Normalmente isso é feito derivando uma expressão para essa soma, diferenciando-se a equação em relação aos parâmetros do ajuste linear,

equacionando para zero e, então, resolvendo o sistema para encontrarmos os parâmetros. Esta abordagem se estende diretamente para muitos outros problemas que não possuam modelos lineares. Nos casos prováveis em que os dados são imperfeitos o modelo resultante também será imperfeito. Na maioria dos casos isso não é problema, se o ruído nos dados é, de certa forma, “bem-comportado”, assim, pode ser que o resultado do modelo é, o melhor possível, representado por um processo estatístico. Por outro lado, se existem vários dados discrepantes é possível que a resposta do modelo estatístico tenha distorções graves.

Reconhecendo essa possibilidade, pode-se tentar identificar as discrepâncias como ponto de dados através de uma grade resíduo em relação ao modelo ajustado. Esses pontos podem ser removidos e, então, o modelo pode ser recalculado. Essa ideia superficialmente atraente é frequentemente utilizada em muitas circunstâncias pode ter o efeito desejado. Por outro lado, ao faz essas hipóteses sobre a natureza dos dados, pode-se cometer erro e levar a um resultado inválido. Isso pode ocorrer, pois os erros são característicos de dois tipos:

- Erros de Medição: Uma observação a partir de uma imagem, ou um parâmetro adquirido de tal observação, não é totalmente correta. É comum que esses erros sejam relativamente pequenos, com média zero (ou relativamente próximo a zero) e, normalmente, apresentam uma distribuição normal;
- Erros de Classificação: Esses erros ocorrem quando algo é mal identificado. Tais erros são, frequentemente, grosseiros e não existe razão de que de maneira geral tenham média zero.

Erros do segundo tipo (erros de classificação) podem distorcer o modelo de tal forma que tentativas de correções normalmente pioram a situação mais do que a corrigem. Isso é bem ilustrado em no caso apresentado na Figura 2.9 em que pode ser visto em uma simples modelo 2D (FISCHLER; BOLLES, 1981).

Subjacente à abordagem dos mínimos quadrados tem um pressuposto de que usando o máximo de dados possível tem-se um efeito benéfico suavização. Porem em muitas circunstâncias onde isso é falso e a aproximação oposta de usando o mínimo o possível pode ser melhor. No caso de um procurar um ajuste linear apenas dois pontos são o suficiente para definir uma linha. Por exemplo, ao selecionar dois pontos ao acaso de uma sequência de dados e que, por hipótese, a linha que une os dois pontos seja o modelo correto. Podemos testar o modelo ao verificarmos quanto

dos pontos restantes estão, de certa forma, “próximos” ao palpite da linha escolhida. A esses pontos chamamos de **Pontos de Consenso** (Consensus Point). Se existe um número significante dos pontos consenso, então recalculando o palpite baseado no conjunto de consenso irá melhorar o modelo sem termos que lidar com os pontos discrepantes.

O apresentado acima descreve de forma informal o algoritmo conhecido como **Amostra Aleatória de Consenso** ou **RANSAC** (Random Sample Consensus) (FISCHLER; BOLLES, 1981). A seguir será apresentado o algoritmo RANSAC de forma mais formal.

2.3.4.1 Amostra aleatória de consenso para ajustes de modelos

Para um melhor entendimento da utilização do RANSAC para ajustes de modelo será apresentado funcionamento do seu algoritmo a seguir:

1. Suponhamos que temos n pontos de dados $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ que gostaríamos de ajustar a um determinado modelo utilizando (no mínimo) m pontos (em que $m \leq n$, e para uma linha temos que no mínimo $m = 2$);
2. Estabelece-se um contador de interações $k = 1$;
3. Escolhe-se, ao acaso, m elementos de X e calcula-se o modelo esperado;
4. Para uma tolerância ϵ, determina-se quantos elementos de X estão dentro de ϵ utilizando o modelo encontrado no passo 3 (esses serão os pontos de consenso). Se esse número exceder um limiar (threshold) t, deve-se recalcular o modelo utilizando os pontos de consenso, utilizando o modelo de mínimos quadráticos ou outro processo semelhante, e encerra-se;
5. Incrementa-se k $(k = k + 1)$. Se $k < K$, para um valor de K pré-determinado, vá para o passo 3. Caso contrário aceita-se o modelo com o maior conjunto de consenso encontrado ou, em última hipótese, o modelo falhou.

Existem muitas melhorias óbvias possíveis para esta apresentação simples do algoritmo RANSAC. O mais simples é observar que a escolha aleatória feita no passo 3 pode, muitas vezes, ser melhorada quando se tem um conhecimento prévio dos dados ou de suas propriedades. Ou seja, podemos ter um conhecimento a priori de que alguns pontos de dados se ajustam melhor a um modelo do que outros.

O algoritmo RANSAC depende da escolha de três parâmetros:

- ϵ, o desvio "aceitável" de um bom modelo: são raras às vezes em que isso pode ser determinada em um sentido analítico, sendo a escolha desse parâmetro feita de forma mais empírica. Empiricamente podemos ajustar um modelo de m pontos, medir os desvios, e, então, escolher ϵ de tal forma que seja um número que corresponda ao desvio padrão aceitável acima da média de um erro aceitável;
- t, é o tamanho do conjunto de consenso considerado ser "suficiente": Esse parâmetro serve, na verdade, para dois propósitos simultaneamente. Ele representa a quantidade de pontos de dados "suficientes" para confirmar a suposição do modelo e a quantidade de pontos de dados "suficiente" para refinar as estimativas parciais para que, no final, chegue na melhor estimativa. O primeiro ponto aqui não é fácil de ser especificado, mas é sugerido utilizar um valor tal que $t - m > 5$ (FISCHLER; BOLLES, 1981);
- K, corresponde a quantidade de iterações que o algoritmo deverá realizar a busca até chegar a um ajuste satisfatório: Tem-se apresentado argumentos (FISCHLER; BOLLES, 1981) de que há formas de calcular o número esperado de tentativas necessárias para selecionar um subconjunto de m pontos de dados "bons". Um argumento estatístico simples nos fornece isso como w^{-m}, em que w é a probabilidade de que um dado escolhido aleatoriamente esteja dentro da condição ϵ do modelo. O desvio padrão dessa estimativa também é da ordem de w^{-m}, assim a escolha de K sendo $K = 2w^{-m}$ ou $K = 3w^{-m}$ é apresentada como sendo uma escolha razoável (FISCHLER; BOLLES, 1981). É claro que isso requer um pouco de uma pré-avaliação dos dados para, pelo menos, se ter uma estimativa aproximada de w.

O algoritmo RANSAC representa uma mudança de paradigma na escolha de um modelo para o ajuste dos dados. Esse algoritmo tem a característica de que “começa pequeno e cresce”, o que é o contrário da abordagem do método dos mínimos quadráticos e técnicas relacionadas que espera extrair o ajuste através do cálculo das médias dos desvios. RANSAC provou ser uma técnica muito fértil e confiável, particularmente em muitos aspectos relacionados à visão computacional, e

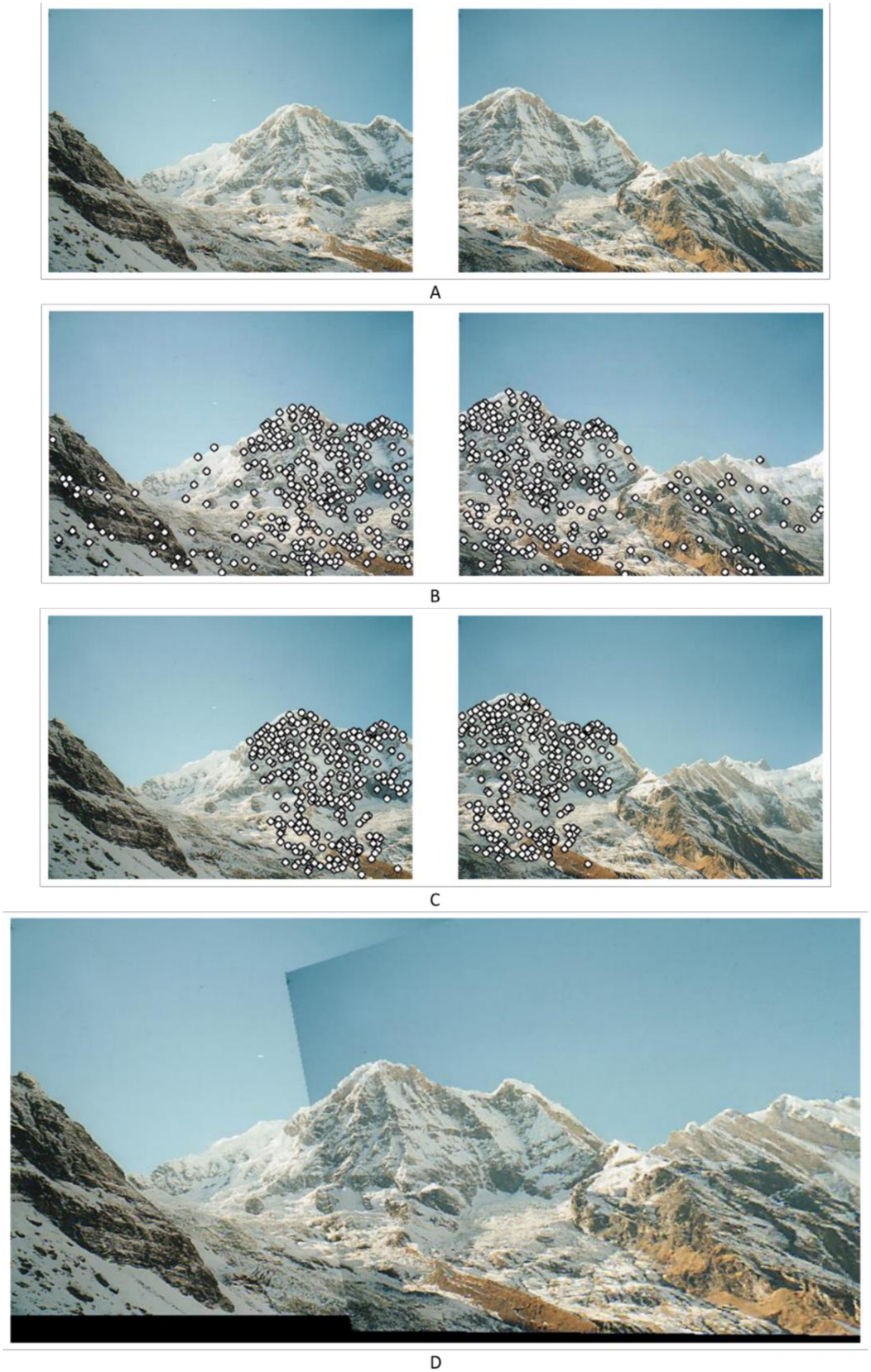
a abordagem provida pelo algoritmo RANSAC tem beneficiado muitas melhorias e aplicações desde o seu desenvolvimento. Uma das aplicações que tem chamado a atenção recentemente é o *Autostitch* (disponível em http://www.cs.bath.ac.uk/brown/autostitch/autostitch.html) que consegue agrupar fotos distintas e criar uma imagem em panorama (BROWN; LOWE, 2003A).

O aplicativo *Autoatitch* trabalha através da identificação de “pontos de interesses” (por exemplo, cantos) nas imagens e representa eles como vetores de características que capturam propriedades de intensidade muito locais. Técnicas eficientes foram implementadas no programa para permitir a localização e a cominação de prováveis pontos comuns em diversas imagens, quando um par de imagens tem uma quantidade significante de combinações, de forma que eles sejam os mesmos pontos, o programa sobrepõe as imagens criando uma combinação.

Se duas imagens são candidatas a serem sobrepostas, a “diferença” entre elas é caracterizada por três possíveis rotações de câmera (em relação aos três eixos cartesianos) e uma modificação do ponto focal, sendo, então, avaliados quatros parâmetros. Busca-se uma homografia que captura essa “diferença” entra as imagens. O problema surge quando um conjunto inicial (relativamente grande) de pontos correspondentes determinados inicialmente possui muitas falsas possibilidades de acertos. Essas falsas possibilidades irão levar a muitas homografias discrepantes da correta. Para corrigir esse problema, é utilizado o algoritmo RANSAC a esse grande número de possíveis combinações com $m = 4$, de forma a encontrar o conjunto de melhores parâmetros possíveis para a homografia. Na sequência é feito um processamento da imagem para verificar a qualidade da montagem, obter uma resolução global da geometria da câmara e é aplicado algumas filtragens de intensidade sofisticado para remover as variações nas bordas das imagens sobrepostas a fim de ter uma imagem única, sem ser possível identificar a posição em que houve a sobreposição das imagens.

O algoritmo obtido é extremante robusto e rápido, sendo capaz de trabalhar com vários panoramas e de identificar imagens não pertencentes ao conjunto de fotos que formarão o panorama. A qualidade final da cominação das imagens é surpreendentemente boa. A Figura 2.11 apresenta uma parte do seu processo de reconhecimento e construção do panorama de forma simples, utilizando um único par de imagens.

Figura 2.11 – Uso do algoritmo RANSAC para criar panoramas através de mosaico de fotos



A – Um par de imagens que se sobrepõem; B – Cada marcação nas imagens representa pontos de interesses, cujos vetores são semelhantes nas duas imagens; C – Após a utilização do algoritmo RANSAC os pontos discrepantes foram removidos, restando apresentados apenas os pontos que pertence às duas imagens; D – Resultado após a sobreposição das imagens
Fonte: Brown e Lowe (2003B)

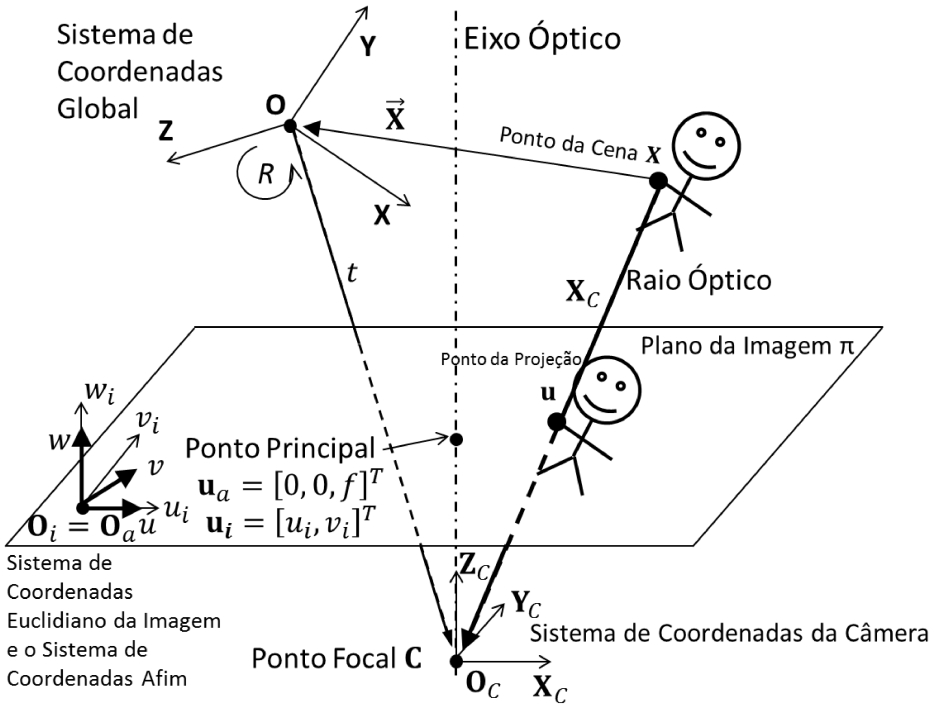
Apesar do *Autostitch* não ser um método de interpretação de imagem por si só, ele apresenta uma forma fácil e eficiente de ilustrar a eficiência e o funcionamento do algoritmo RANSAC. Como apresentado, algoritmo RANSAC representa um algoritmo de ajuste de modelos genéricos aplicável diretamente em diversos processos de interpretação de imagens.

2.4 Câmera com uma Única Perspectiva

2.4.1 Modelo para câmeras

O modelo de câmera *pinhole* apresentado na Figura 2.8 é uma aproximação aceitável para diversas aplicações de visão computacional. A câmera de lentes finas transforma a cena 3D em uma imagem 2D através da projeção central, o efeito geométrico da projeção central é representado na Figura 2.12. Com pode ser observado na Figura 2.12, o plano π é o **plano da imagem** no qual o mundo real é projetado. A linha traço-ponto apresentada na vertical é o **eixo óptico**. A lente é posicionada perpendicularmente ao eixo óptico no **ponto focal C** (também chamado de **Centro Óptico** ou **Centro de Projeção**). O comprimento focal f é um parâmetro da lente.

Figura 2.12 – As características geométricas da perspectiva linear de uma câmera



Fonte: Autor

Será adotada a seguinte notação: os pontos nas imagens 2D serão indicados por letras minúsculas em negrito tanto em coordenadas Euclidiana (espaço não homogêneo), como $\mathbf{u} = [u, v]^T$, quanto as coordenadas homogêneas, como $\mathbf{u} = [u, v, w]^T$. Todas os pontos das cenas 3D serão indicados por letras maiúsculas e em negrito, quando se tratar de coordenadas Euclidiana, como $\mathbf{X} = [X, Y, Z]^T$, e quando se referirem a coordenadas homogêneas, como $\mathbf{X} = [X, Y, Z, W]^T$.

A função das câmeras é realizar uma transformação linear do espaço projetivo 3D $\mathcal{P}^3$ para o espaço projetivo 2D $\mathcal{P}^2$. A projeção é realizada por um raio óptico que ou é originário ou é refletido de um ponto $\boldsymbol{X}$ da cena (processo representado na parte superior direita da Figura 2.12) ou originado em uma fonte luminosa (processo representado na parte superior esquerda da Figura 2.12). O raio óptico se dirige ao centro óptico **C** e atinge o plano de imagem criando o ponto $\mathbf{u}$ projetado.

Para a compreensão da próxima etapa é necessário entender os quatros sistemas de coordenadas apresentado na Figura 2.12:

1. O **Sistema de Coordenadas Euclidiana Global** tem sua origem no ponto O e os pontos $\boldsymbol{X}$ e $\boldsymbol{u}$ são expressos em relação a esse sistema de coordenadas;
2. O **Sistema de Coordenadas Euclidiana da Câmera** (subscrito $_C$) tem a sua origem no ponto focal, ou seja $\mathbf{C} \equiv \mathbf{O_C}$. O eixo de coordenadas Z_C está alinhado com o eixo focal e a sua direção é a partir do ponto focal $\mathbf{C}$ para o plano da imagem. Há uma relação única entre o sistema de coordenada global e o da câmera que pode ser descrito por uma transformação Euclidiana que consiste em uma translação t e uma rotação R;
3. O **Sistema de Coordenadas Euclidiana da Imagem** (subscrito $_i$) tem seus eixos alinhados com o sistema de coordenadas da câmera. Os eixos coordenados u_i, v_i e w_i são paralelos aos eixos de coordenadas X_C, Y_C e Z_C respectivamente, sendo que os eixos u_i e v_i estão no plano da imagem;
4. O **Sistema de Coordenadas Afim** (subscrito $_a$) tem os eixos de coordenadas u, v e w e, também, possui a origem O_a que é coincidente com a origem do sistema de coordenadas da imagem O_i. Os eixos de coordenadas u e w são alinhados com os eixos u_i e w_i, mas o eixo v pode ter uma orientação diferente em relação ao eixo v_i. O motivo pelo qual o sistema de coordenadas afim foi apresentado é devido ao fato que os

pixels podem sofrer cisalhamento, normalmente devido a um desalinhamento do chip fotossensível da câmera. Além do mais, os eixos de coordenadas podem possuir escalas diferentes.

De maneira geral, a transformação projetiva pode ser fatorizada em três transformações simples, as quais correspondem a três translações entre os quatro sistemas de coordenadas diferentes.

A primeira das transformadas, que envolve o sistema de coordenadas global e o da câmera, constitui-se da transição do sistema de coordenadas global (O; X, Y, Z) para o sistema de coordenadas centrado na câmera (O_C: X_C, Y_C, Z_C). O sistema de coordenadas global pode ser alinhado com o sistema de coordenadas da câmera fazendo a translação da origem O para a O_C através do vetor de translação t e através da rotação dos eixos de coordenadas pela matriz de rotação R, transformando, assim, o ponto $\boldsymbol{X}$ para o ponto $\boldsymbol{X}_C$. Expressando esse processo em coordenadas não homogêneas fica:

$$\mathrm{X_C} = \mathrm{R(X - t)} \tag{2.20}$$

A matriz de rotação R apresenta três rotações elementares sobre os eixos de coordenadas, rotação sobre os eixos X, Y e Z. O vetor de translação t apresenta três elementos de translação da origem do sistema de coordenadas global em relação ao sistema de coordenadas da câmera. Assim, existem seis parâmetros extrínsecos da câmera, três rotações e três translações, sendo que a matriz R e o vetor t são chamados de **Parâmetros Extrínsecos da Calibração da Câmera**.

Podemos expressar a Equação (2.20) em coordenadas homogêneas por meio do subgrupo de homografia H_S a partir da Equação (2.11). De forma mais formal, temos:

$$\mathrm{X_C} = \begin{bmatrix} \mathrm{R} & -\mathrm{R} \cdot \mathrm{t} \\ 0^{\mathrm{T}} & 1 \end{bmatrix} \mathrm{X} \tag{2.21}$$

A segunda transformada, que envolve o sistema de coordenadas da câmera e o sistema de coordenadas da imagem, projeta o ponto $\boldsymbol{X}_C$, expresso no sistema de coordenadas da câmera (O_C: X_C, Y_C, Z_C), da cena 3D para criar o ponto $\boldsymbol{u}_i$ no plano π (plano da imagem) de coordenadas (O_i: u_i, v_i, w_i).

A projeção de $\mathcal{R}^3 \longrightarrow \mathcal{R}^2$ em coordenadas não homogêneas nos fornece duas equações não lineares em Z_C:

$$u_i = \frac{X_C \cdot f}{Z_C} \tag{2.22}$$

$$v_i = \frac{Y_C \cdot f}{Z_C} \tag{2.23}$$

Em que f corresponde à distância focal. Se a projeção dada pela Equação (2.22) e pela Equação (2.23) é incorporada no espaço projetivo, então a projeção $\mathcal{P}^3 \longrightarrow \mathcal{P}^2$ pode ser descrita de forma linear em coordenadas homogêneas como:

$$\mathrm{u_i} \simeq \begin{bmatrix} \mathrm{f} & 0 & 0 & 0 \\ 0 & \mathrm{f} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathrm{X_C} \tag{2.24}$$

Uma câmera que possua uma distância focal especial de $f = 1$ (também conhecida como **Câmera com o Plano de Imagem Normalizado** – FORSYTH; PONCE, 2002) irá levar a simplificação da Equação (2.24), tornando-a em:

$$\mathbf{u}_i \simeq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}_C \tag{2.25}$$

A terceira transformada envolve o sistema de coordenadas da imagem e o sistema de coordenadas afim. A vantagem dessa transformada é que ela consegue reunir todos os parâmetros intrínsecos a uma câmera (sendo o comprimento focal f um deles) em uma matriz K de 3×3 denominada de **Matriz de Calibração Intrínseca**. K é uma matriz triangular superior e expressa o mapeamento $\mathcal{P}^2 \longrightarrow \mathcal{P}^2$ que é um caso especial da transformação afim. Esse caso especial é chamado, também, de **uma Transformação Afim Fatorizada por Rotações**, e ela cobre o escalonamento não isotrópico e cisalhamento da imagem. Ela pode ser realizada dentro do plano da imagem, como apresentado na Figura 2.12. Essa transformação $\mathcal{P}^2 \longrightarrow \mathcal{P}^2$ pode ser descrita como:

$$\mathbf{u} \simeq K \cdot \mathbf{u}_i = \begin{bmatrix} f & s & -u_0 \\ 0 & g & -v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{u}_i \tag{2.26}$$

Os parâmetros da matriz de calibração intrínseca são: f fornece uma mudança de escala em relação ao eixo u enquanto que g fornece uma mudança de escala em relação ao eixo v. Frequentemente, ambos os valores são iguais à distância focal ($f = g$). O parâmetro s fornece o grau de cisalhamento dos eixos de coordenadas no plano da imagem. Normalmente presume-se que, na imagem, o eixo v do sistema de coordenadas afim é coincidente com o eixo v_i do sistema de coordenadas euclidiano da imagem. O valor de s representa o quanto o eixo u está inclinado na direção do eixo v. O parâmetro de cisalhamento s é introduzido, na

prática, para lidar com as distorções. Um exemplo de distorção seria a provocada pela colocação de um chip fotossensível de forma não perpendicular ao eixo óptico durante a montagem da câmara.

Agora é possível especificar o efeito de projeção de uma câmera de lente fina em toda a sua generalidade. Já foi descrito que projeção da câmera é uma transformação linear do espaço projetivo 3D $\mathcal{P}^3$ para o espeço projetivo 2D $\mathcal{P}^2$. A transformada é o produto de três fatores já apresentados e que podem ser expressos matematicamente pelas Equações (2.21), (2.25) e (2.26). Ao juntarmos os três fatores pode-se representar a projeção da câmera como:

$$\mathrm{u} \simeq \mathrm{K}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\mathrm{R} & -\mathrm{R}\cdot\mathrm{t}\\ 0^{\mathrm{T}} & 1\end{bmatrix}\mathrm{X} \qquad (2.27)$$

O produto do segundo com o terceiro fator apresenta uma estrutura interna útil, assim, podemos reescrever a Equação (2.27) como:

$$\mathbf{u} \simeq K\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}R & -R\cdot t\\ \mathbf{0}^T & 1\end{bmatrix}\mathbf{X} = K[R| \quad -R\cdot t]\mathbf{X} = M\cdot\mathbf{X} \qquad (2.28)$$

Se expressarmos os pontos da cena em coordenadas homogêneas, pode-se descrever a projeção perspectiva de forma linear utilizando uma única matriz M que é 3×4 chamada de **Matriz de Projeção** ou **Matriz da Câmera**. Os 3×3 elementos mais a esquerda da matriz M fazem parte de uma submatriz que descreve as rotações e a coluna mais a direita da matriz M descreve a translação. O delimitador | na notação explicita que a matriz M é composta de duas submatrizes (a de rotação e a de translação). É possível de verificar pela Equação (2.29) que na notação da matriz M contém todos os parâmetros intrínsecos e extrínsecos.

$$M = K[R| \quad -R\cdot t] \qquad (2.29)$$

Esses parâmetros podem ser obtidos pela decomposição de M em K, R e t, sendo essa decomposição única para cada câmera. Se escrevermos $M = [\boldsymbol{A}| \quad \mathbf{b}]$, temos que $\boldsymbol{A} = K\cdot R$, $\mathbf{b} = -\boldsymbol{A}\cdot t$ e $t = -\boldsymbol{A}^{-1}\cdot\mathbf{b}$. A decomposição de $\boldsymbol{A} = K\cdot R$, em que K é uma matriz diagonal superior e R é de rotação, pode ser realizada através da "Decomposição RQ", similar ao processo mais conhecido de "Decomposição QR" (PRESS et al., 1992; GOLUB; VAN LOAN, 1989) como descrito na seção da página 46 em Subgrupos da homografia.

2.4.2 Projeção e retroprojeções em coordenadas homogêneas

A Equação (2.28) nos fornece um resultado importante, que em coordenadas homogêneas a projeção de um ponto $\mathbf{X}$ da cena para um ponto $\mathbf{u}$ da imagem pela câmera é devido ao mapeamento linear simples:

$$\mathrm{u} \simeq \mathrm{MX} \tag{2.30}$$

Pela Equação (2.30) é possível notar que o mapeamento feito pela câmera é semelhante ao mapeamento feito pela homografia apresentada na Equação (2.5). Entretanto, para homografias a matriz H era quadrada e, em geral, não singular. Isso garante que o mapeamento seja 1-para-1, ou seja, o mapeamento é único. No caso das projeções a matriz M não é quadrada e, dessa forma, temos o mapeamento de muitos-para-1. Em outras palavras, podemos dizer que todos os pontos de uma cena pertencente a um raio são projetados em um único ponto da imagem.

Existe um único ponto da cena que não possui imagem na câmera, esse ponto é definido como Centro de Projeção $\mathbf{C}$. O Centro de Projeção tem a propriedade de que $M \cdot \mathbf{C} = \mathbf{0}$. E isso nos permite recuperar M utilizando, por exemplo, SVD. Em outras palavras $\mathbf{C}$ é um vetor ortogonal às linhas de M. Desta forma, temos que $\mathbf{C}$ é único e importante.

A Equação (2.28) também nos permite obter uma derivação simples para a expressão da retroprojeção dos pontos e das linhas por uma câmera M. Por meio da retroprojeção tem-se a possibilidade de calcular toda a cena 3D que se projeta para uma dada imagem pela transformação dada por M.

Dado um ponto $\mathbf{u}$ em uma imagem em coordenadas homogênea, pode-se encontrar a pré-imagem de uma cena. Essa pré-imagem não é obtida de forma única, uma vez que todos os pontos em um raio da cena serão projetados em $\mathbf{u}$. Um ponto nesse raio é o centro de projeção $\mathbf{C}$. Outro ponto no raio pode ser obtido de $\mathbf{u} = M \cdot \boldsymbol{X}$ como

$$\mathrm{X} = \mathrm{M}^{+} \cdot \mathrm{u} \tag{2.31}$$

Em que $\mathrm{M}^{+} = \mathrm{M}^{\mathrm{T}}(\mathrm{M} \cdot \mathrm{M}^{\mathrm{T}})^{-1}$, denominada **pseudoinversa**, é a generalização da inversão de uma matriz não quadrada. Ela tem a propriedade de que $\mathrm{M} \cdot \mathrm{M}^{+} = I$.

Por exemplo, suponha uma dada imagem de uma linha $\mathbf{l}$**,** em coordenadas homogênea, a qual gostaríamos de encontrar a sua pré-imagem. Neste caso teremos uma solução que não é única, pois todo o plano $\mathbf{a}$ da cena será projetado em $\mathbf{l}$. Um ponto $\mathbf{X}$ pertencente a $\mathbf{a}$ satisfaz a $\mathbf{a}^{T} \cdot \mathbf{X} = 0$ e a sua projeção é $\mathbf{u} = M \cdot \mathbf{X}$. Essa

projeção precisa estar contida em $\mathbf{l}$, o que nos leva à $\mathbf{I}^T \cdot \mathbf{u} = \mathbf{I}^T \cdot M \cdot \mathbf{X} = 0$. Dessa forma, temos:

$$\mathbf{a} = \mathrm{M}^T \cdot \mathbf{l} \tag{2.32}$$

Em que esse plano contém o centro de projeção, $\mathbf{a}^T \cdot \mathbf{C} = 0$.

2.4.3 Calibração da câmera por uma cena conhecida

Nesta etapa será apresentado como calcular a matriz M da projeção da câmera através da correspondência de um conjunto de pontos de imagem, ou seja, dado um conjunto $\{(\mathbf{u}_i, \mathbf{X}_i)\}_{i=1}^m$ onde $\mathbf{u}_i$ é um vetor homogêneo de 3 coordenada representando os pontos da imagem e $\mathbf{X}_i$ é um vetor homogêneo de 4 posições que representa os pontos da cena. Esse procedimento também é conhecido como **Resseção de Câmera (Camera Resectioning)**.

Essa situação é similar a estimação de uma homografia, como descrito na subseção 2.3.3. Precisamos solucionar o sistema linear homogêneo

$$\alpha_i \cdot \mathbf{u}_i' = M \cdot \mathbf{X}_i, \qquad i = 1, \cdots, m \tag{2.33}$$

Para M e para α_i. M é determinado até uma escala, sendo que M possui, no máximo, 11 parâmetros livres. Dos parâmetros de M, apenas 6 (também dito $5\frac{1}{2}$) correspondências são necessárias para calcular a matriz M. De forma similar ao cálculo da homografia, existem configurações degenerativas que corrompem a formas da homografia e previnem que M tenha uma solução única, mesmo que $m \geq 6$. As configurações degenerativas são mais complexas e não podem ser resolvidas com a técnica da homografia (HARTLEY, 1997; HARTLEY; ZISSERMAN, 2003).

A estimação linear de M pela minimização da distância algébrica é completamente análogo à feita para a homografia. Ao multiplicarmos a equação $\mathbf{u} \simeq M \cdot \mathbf{X}$ por $S(\mathbf{u})$ faz com que o lado esquerdo da equação suma, levando a $0 = S(\mathbf{u}) \cdot M \cdot \mathbf{X}$. Rearranjando essa equação, temos $[\mathbf{X}^T \circledast S(\mathbf{u})]\mathbf{m} = 0$, onde $\mathbf{m} = [\mathrm{m}_{11}, \mathrm{m}_{21}, \cdots, \mathrm{m}_{24}, \mathrm{m}_{34}]^T$ e $\circledast$ é o produto de Kronecker. Considerando todas as correspondências de m, isso nos dá um sistema na forma:

$$\begin{bmatrix} \mathbf{X}_1^T \circledast S(\mathbf{u}_1) \\ \cdots \\ \mathbf{X}_m^T \circledast S(\mathbf{u}_m) \end{bmatrix} \mathbf{m} = \mathrm{W} \cdot \mathbf{m} = \mathbf{0} \tag{2.34}$$

Em que a distância algébrica $\|\mathrm{W} \cdot \mathbf{m}\|$ pode ser minimizada fazendo com que $\|\mathbf{m}\| = 1$ utilizando, por exemplo, SVD. Para isso uma pré-condição é necessária, a

garantia que as componentes dos vetores $\mathbf{u}_i$ e $\mathbf{X}_i$ tenham magnitudes similares. Opcionalmente, pode-se decompor M em parâmetros intrínsecos e extrínsecos, tais como os dados pela Equação (2.29).

Ao obter a estimativa inicial pelo método linear, pode-se, em seguida, melhorar o resultado pelo processo de aproximação pela maior vero semelhança utilizando o método dos mínimos quadráticos não-linear. Entretanto deve-se tomar cuidado nesta etapa ao especificar um ruído apropriado para o modelo dos pontos da cena, sendo que isso depende do cenário particular no qual a calibração da câmera foi usada.

2.5 Considerações Finais

Este capítulo apresenta a técnica utilizada para analisar as imagens e como desenvolver o modelo em 3D utilizando representações 2D das cenas.

O próximo capítulo mostra a técnica utilizada para analisar as imagens e como desenvolver o modelo em 3D utilizando representações 2D de cenas.

CAPITULO 3

3 RECONSTRUÇÃO DA CENA ATRAVÉS DE MÚLTIPLAS VISTAS

3.1 Introdução

Aqui será apresentado como calcular os pontos 3D da cena através de projeções de várias câmeras. Essa tarefa é fácil se os pontos da imagem e a matriz da câmera são dados, dessa forma apenas os pontos 3D da cena devem ser calculados. Se a matriz da câmera é desconhecida, a tarefa aumenta, já que é necessário encontrar os pontos 3D da cena e as matrizes das câmeras. Isso é consideravelmente mais difícil e é a tarefa central da formação de geometria por múltiplas vistas.

3.2 Triangulação

Assumido que a matriz da câmera M e que os pontos da imagem $\mathbf{u}$ são dados pode-se encontrar os pontos $\mathbf{X}$ da cena utilizando relação de triângulos. Representando as diferentes imagens pelo subscrito j e assumido que n vistas estão disponíveis para encontrarmos os pontos $\mathbf{X}$, basta resolvermos o sistema linear homogêneo

$$\alpha^j \cdot \mathbf{u}^j = M^j \cdot \mathbf{X}, \qquad j = 1, \cdots, n \tag{3.1}$$

Essa técnica é conhecida como **Triangulação**. O nome é derivado de uma técnica de fotogrametria na qual o processo original era a interpretação em termos de triângulos similares.

O procedimento é relativamente simples, pois a Equação (3.1) é linear. É muito similar à estimativa homográfica (apresentado na subseção 2.3.3) e à calibração da câmera utilizando uma cena conhecida (subseção 2.4.3).

Geometricamente, a triangulação consiste em encontrar as interseções comuns de n raios dados pela retroprojeção dos pontos da imagem pela câmera. Se não houver ruído nas medições de $\mathbf{u}^j$ e na determinação de M^j, então esses raios só podem se interceptar em um único ponto e o sistema da Equação (3.1) deverá

apresentar uma única solução. Na verdade, os raios não deverão se interceptar (skew) e o (super determinado) sistema da Equação (3.1) não deverá ter solução.

Devemos calcular $\mathbf{X}$ como sendo o ponto da cena mais próximo o possível da intercepção dos raios (skew). Para $n = 2$ câmeras, esse processo se resume em encontrarmos o ponto médio do menor segmento de reta entre dois raios. Entretanto, isso não é ótimo estatisticamente. A aproximação mais correta seria a estimativa pela maior vero semelhança, o que permite minimizar o erro de projeção. Denominando por $\left[\hat{\mathrm{u}}^j, \hat{\mathrm{v}}^j\right]^T$ os pontos da imagem em coordenadas não homogênea, podemos solucionar o problema de otimização por:

$$\min_{\mathbf{X}} \sum_{j=1}^{m} \left[\left(\frac{\mathbf{m}_1^{j\,T} \cdot \mathbf{X}}{\mathbf{m}_3^{j\,T} \cdot \mathbf{X}} - \hat{\mathrm{u}}^j \right)^2 + \left(\frac{\mathbf{m}_2^{j\,T} \cdot \mathbf{X}}{\mathbf{m}_3^{j\,T} \cdot \mathbf{X}} - \hat{\mathrm{v}}^j \right)^2 \right] \tag{3.2}$$

Em que $\mathbf{m}_i^j$ representam a *i*-ésima linha da matriz da câmera M^j. Esta equação assume que somente os pontos da imagem estão corrompidos por ruído, enquanto que a as matrizes das câmeras não apresentam ruído.

O problema de otimização não convexo apresentado na Equação (3.2) é conhecido por ter vários mínimos locais, o que impossibilita a obtenção direta do resultado desejado. Entretanto uma forma fechada da solução para o caso simples de $m = 2$ câmeras já é conhecida (HARTLEY, 1997). Podemos solucionar este caso primeiramente encontrando uma estimativa inicial por um método linear e, então, utilizar mínimos quadráticos não linear.

Para formulação de um método linear podemos multiplicar a equação $\mathbf{u} \simeq M \cdot \mathbf{X}$ por $S(\mathbf{u})$, dessa forma o lado esquerdo da equação é anulado $0 = S(\mathbf{u}) \cdot M \cdot \mathbf{X}$. Considerando todas as n câmeras, obtermos o sistema:

$$\begin{bmatrix} S(\mathbf{u}^1) \cdot M^1 \\ \cdots \\ S(\mathbf{u}^n) \cdot M^n \end{bmatrix} \mathbf{X} = \mathrm{W} \cdot \mathbf{X} = \mathbf{0} \tag{3.3}$$

O sistema de equações apresentado na Equação (3.3) pode ser solucionado pela minimização da distância algébrica utilizando SVD. Porém, a pré-condição para que a solução por SVD seja feito é garantir que as componentes de $\mathbf{u}^j$ e M^j não tenham magnitudes muito diferentes. Às vezes basta substituir $\mathbf{u} \simeq M \cdot \mathbf{X}$ por $\overline{\mathbf{u}} \simeq \overline{M} \cdot \mathbf{X}$. Em que $\overline{\mathbf{u}} = H_{pre} \cdot \mathbf{u}$ e $\overline{M} = H_{pre} \cdot M$. Aqui H_{pre} é obtido como descrito na subseção 2.3.3. Entretanto, às vezes isso não remove as grandes diferenças nas entradas de M. Então, é necessário substituir $\overline{M} = H_{pre} \cdot M \cdot T_{pre}$, em que T_{pre} é uma

matriz 4×4 adequada que representa uma homografia 3D. Nesse caso, não existe um método único conhecido para determinar T_{pre} e H_{pre} parece ser bom em todas as situações e o pré-condicionamento ainda é um tipo de arte.

3.2.1 Observação na reconstrução de uma linha 3D

Dependendo do formato do objeto que do qual será realizada a reconstrução 3D é melhor reconstruir entidades geométricas do que pontos. Para reconstruir uma linha 3D a partir de sua projeção $\mathbf{l}^j$ em uma câmera M^j, deve-se recorrer à Equação (2.32). Nas medidas com ausência de ruídos os planos devem possuir uma única linha em comum. Podemos representar essa linha por dois pontos $\mathbf{X}$ e $\mathbf{Y}$ que estejam contidos na linha, dessa forma satisfazendo $\mathbf{a}^T[\mathbf{X}|\mathbf{Y}] = [0,0]$. Para garantir que os pontos sejam distintos é necessário que $\mathbf{X}^T \cdot \mathbf{Y} = 0$. A intersecção pode ser encontrada ao solucionar o seguinte sistema linear

$$W \cdot [\mathbf{X}|\mathbf{Y}] = \begin{bmatrix} (\mathbf{l}^1)^T \cdot M^1 \\ \cdots \\ (\mathbf{l}^n)^T \cdot M^n \end{bmatrix} [\mathbf{X}|\mathbf{Y}] = 0, \qquad \mathbf{X}^T \cdot \mathbf{Y} = 0 \tag{3.4}$$

Podemos encontrar W facilmente, fazendo $W = U \cdot D \cdot V^T$ sendo $U \cdot D \cdot V^T$ a decomposição SVD de W. Os pontos $\mathbf{X}$ e $\mathbf{Y}$ são obtidos pelas as duas colunas de V associadas com os dois menores valores singulares.

Esse método linear pode ser seguido pela *estimativa* por máxima *verossimilhança*. *P*ara refletir onde o ruído entra no processo corretamente, um bom critério é o de minimizar o erro de imagem retroprojeção dos pontos finais dos segmentos de linha de imagem medidos.

O pré-condicionamento é necessário porque assim pode-se garantir que os componentes de $\mathbf{l}^j$ e M^j possuem magnitudes similares.

3.3 Reconstrução projetiva

Suponha que hajam m pontos da cena na forma $\mathbf{X}_i$ $(i = 1, \cdots, m)$, sendo estes pontos diferenciados pelos sobrescritos, e m câmeras M^j $(j = 1, \cdots, m)$, sendo estas câmeras diferenciadas pelos sobrescritos. Os pontos da cena projetam nas câmeras a imagem como:

$$\alpha_i^j \cdot \mathbf{u}_i^j = M^j \cdot \mathbf{X}_i, \qquad i = 1, \cdots, m \qquad j = 1, \cdots, n \tag{3.5}$$

Na Equação (3.5) pode-se indicar o *i-ésimo* ponto da imagem na *j-ésima* imagem utilizando os sobrescritos e os subscritos, como $\mathbf{u}_i^j$.

Considere um processo onde os pontos da cena $\mathbf{X}_i$ e as matrizes das câmeras M^j são desconhecidos e devem ser calculados a partir dos pontos da imagem $\mathbf{u}_i^j$. Diferente da triangulação mostrado na subseção 3.2 deste capítulo, o sistema de equações da Equação (3.5) não é linear em relação às incógnitas e não é possível encontrar uma solução simples para ele. Pode-se tentar solucionar esse sistema utilizando um conjunto redundante de pontos da imagem, tentando contornar os ruídos. Dessa forma, a Equação (3.5) se torna sobredeterminado e, com isso, mais é difícil de solucionar o problema de reconstrução 3D.

Esse problema pode ser solucionado em duas etapas:

1. Enumerar uma estimativa inicial (de forma não muito precisa) das matrizes M^j calculada pelos pontos da imagem $\mathbf{u}_i^j$. Isso pode ser realizado usando a estimativa dos coeficientes das **restrições correspondentes** ao solucionar o sistema linear de equações e, então, calcular a matriz da câmera M^j utilizando esses coeficientes. Essa mudança de um sistema não-linear para um linear, inevitavelmente, ignora algumas relações não-lineares entre os componentes de M^j. As restrições correspondentes são apresentadas na subseção 3.4 deste capítulo para qualquer número de vistas e mais detalhadamente na subseção 3.7 para duas vistas e na subseção 3.8 para três vistas.
2. Um subproduto desse processo é, geralmente, uma estimativa inicial dos pontos $\mathbf{X}_i$ da cena. Assim, M^j e $\mathbf{X}_i$ são calculados com precisão utilizando a estimação de máxima verossimilhança (ajuste do pacote), como o descrito na subseção 3.5 deste capítulo.

3.3.1 Ambiguidade projetiva

Sem solucionar o sistema dado pela Equação (3.5), alguma informação sobre a singularidade de sua solução pode ser facilmente derivada. Se tomarmos M^j e $\mathbf{X}_i$ como a solução da Equação (3.5) e tomarmos T como sendo uma matriz arbitrária e não singular 3×4, então as câmeras $M'^j = M^j \cdot T^{-1}$ e os pontos da cena $\mathbf{X}_i' = T \cdot \mathbf{X}_i$ são solução da Equação (3.5) pois:

$$M'^{j} \cdot \mathbf{X}'_i = M^j \cdot T^{-1} \cdot T \cdot \mathbf{X}_i = M^j \cdot \mathbf{X}_i \quad (3.6)$$

Dessa forma, ao multiplicarmos por T significa que estamos modificando M^j e $\mathbf{X}_i$ por uma transformação projetiva 3D. Este resultado pode ser interpretado como a incapacidade de recuperar de forma verdadeira as matrizes de câmeras e pontos 3D mais preciso do que, de forma geral, uma transformação projetiva 3D. Qualquer solução particular $\{M'^{j}, \mathbf{X}'_i\}$ que satisfaz a Equação (3.5), ou o processo de calcular $\{M'^{j}, \mathbf{X}'_i\}$, é chamada de **Reconstrução Projetiva** ou **Reconstrução Projetiva 3D**.

De forma a esclarecer o significado da expressão "incapacidade de recuperar de forma verdadeira as matrizes de câmeras e pontos 3D mais preciso do que, de forma geral, uma determinada transformação G" suponhamos que exista uma verdadeira reconstrução desconhecida que nos forneça $\{M^j, \mathbf{X}_i\}$ e que a nossa reconstrução, $\{M'^{j}, \mathbf{X}'_i\}$, difere da verdadeira por uma transformação desconhecida que pode ser fornecida por um grupo de transformação G. Isso significa que sabemos algo a respeito da cena verdadeira e da câmera verdadeira, mas não sabemos tudo. No caso da ambiguidade projetiva é possível chegarmos ao conhecimento de que alguns pontos no conjunto $\mathbf{X}'_i$ são, por exemplo, colineares, assim, os pontos verdadeiros no conjunto dos pontos $\mathbf{X}'_i$ também são colineares. Entretanto, distâncias, ângulos ou volumes calculados na reconstrução projetiva são diferentes, de forma geral, dos da cena verdadeira porque essas características geométricas não são invariantes à transformação projetiva, como já apresentado na subseção 2.3.2.

Sempre é possível escolher T de tal forma que a primeira matriz de câmera tenha a forma simples como

$$M^1 = [\boldsymbol{I}|\mathbf{0}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.7)$$

Esta simplificação é sempre conveniente nas derivações. Em detalhes, é possível afirmar que para uma matriz de câmera arbitrária M existe uma homografia 3D T de tal forma que $M \cdot T^{-1} = [\boldsymbol{I}|\mathbf{0}]$. Dessa forma, T pode ser escolhido como:

$$T = \begin{bmatrix} M \\ \mathbf{a}^T \end{bmatrix} \quad (3.8)$$

Onde $\mathbf{a}$ é qualquer vetor de quatro posições tal que T tenha um ranking completo. Podemos escolher $\mathbf{a}$ convenientemente de forma a satisfazer $M \cdot \mathbf{a} = 0$, por exemplo, $\mathbf{a}$ representando o centro da projeção. Assim, $M = [\boldsymbol{I}|\mathbf{0}] \cdot T$, dessa forma satisfazendo a afirmação.

3.4 Restrições correspondentes

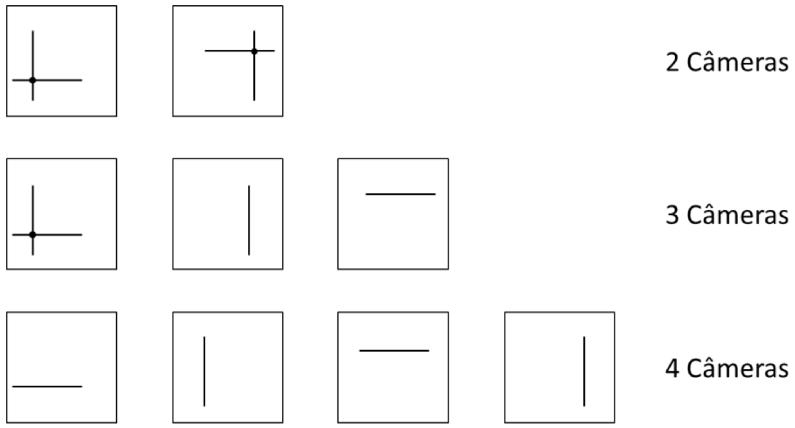
Restrições correspondentes são relações satisfeitas por coleções de pontos correspondentes em n imagens. Eles possuem a propriedade de que uma função multilinear, do sistema de coordenadas homogêneo da imagem, deve desaparecer. Os coeficientes dessas funções formam o **Tensor de Múltiplas Vistas** (*Multiview Tensors*). Exemplos de tensores multilineares são as matrizes fundamentais e o tensor trifocal. Esses elementos são apresentados nas próximas seções deste capítulo. Uma função $f(x_1, \cdots, x_n)$ é multilinear se ela é linear para qualquer variável x_i e se todas as outras são mantidas fixas.

Seja $\mathbf{u}^j$ pontos de uma cena localizados em $j = 1, \cdots, n$ imagens que possuem matrizes da câmera M^j. Para as restrições correspondentes é necessário que um único ponto $\mathbf{X}$ da cena se projete em $\mathbf{u}^j$, ou seja, $\mathbf{u}^j \sim M^j \cdot \mathbf{X}$ para todo j. Foi apresentado na subseção 2.3.3 que toda a formulação matemática pode ser expresso usando matrizes homogênea como mostrado na Equação (3.3).

É possível notar que as linhas de $S(\mathbf{u})$ representam as três linhas da imagem passando por $\mathbf{u}$, onde duas linhas são apresentadas na cena e, ao menos, é uma apresentada no infinito. Pela Equação (2.32), as linhas da matriz $S(\mathbf{u}) \cdot M$ representam os três planos da cena que interceptam no raio retroprojetado a partir de $\mathbf{u}$ pela câmera M. Dessa forma, as linhas da matriz W na Equação (3.3) representam os planos da cena que possuem o ponto $\mathbf{X}$ em comum.

A Equação (3.3) apresenta solução somente se W é uma matriz de rank deficiente, ou seja, todas as parcelas dos cofatores 4×4 devem ser nulas. Isso significa que qualquer um dos $3n \times 4$ planos da cena, representados pelas linhas de W, possuem um ponto em comum. Esses quatro planos são denominados de $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$. Escolhendo diferentes conjuntos dos quatro planos ($\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$) conduz a diferentes restrições correspondentes. Isso indica que todos eles são multilineares, apesar de alguns só se tornarem após serem divididos por um fator comum. A Figura 3.1 ilustra uma interpretação geométrica das restrições em termos de quatro planos

Figura 3.1 – Interpretação geométrica das restrições de bilinearidade, trilinearidade e quadrilinearidade em termos de quatro planos



Fonte: Autor

3.4.1 Para duas vistas

Qualquer quadruplas $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$ contém planos retroprojetados de, pelo menos, duas vistas diferentes. Sejam essas vistas $j = 1, 2$, sem perda de generalidade, no caso em que $\mathbf{a}$, $\mathbf{b}$ e $\mathbf{c}$ são da vista 1 e $\mathbf{d}$ é da vista 2. Esses quatro planos não são o suficiente para serem utilizados como restrição, porque eles sempre apresentam um ponto em comum. Entretanto, se $\mathbf{a}$ e $\mathbf{b}$, forem da vista 1 enquanto $\mathbf{c}$ e $\mathbf{d}$ forem da vista 2, como mostrado na Figura 3.1 (com as linhas no infinito removidas) obtém-se a restrição de bilinearidade. Com isso forma-se $3^2 = 9$ quádruplos com essa propriedade e que cada um dos nove determinantes correspondentes é dividido pelo monômio linear. Após a divisão, todos esses determinantes se tornam iguais, produzindo uma única restrição bilinear. Isso é vastamente conhecido como restrição epipolar que é apresentada em mais detalhes na subseção 3.7.1 deste capítulo.

3.4.2 Para três vistas

Como mostrado na Figura 3.1 tem-se $\mathbf{a}$ e $\mathbf{b}$ pertencentes à vista 1, $\mathbf{c}$ à vista 2 e $\mathbf{d}$ à vista 3. Para esse caso existem $3^3 = 27$ escolhas possíveis para formar as restrições. Cada um dos 27 determinantes são divisíveis por um monômio linear. Após a divisão, restam apenas 9 determinantes diferentes e isso implica na obtenção de 9 restrições trilineares. Pode-se, também, escolher $c = (M^2)^T \cdot \mathbf{l}^2$ e $d = (M^3)^T \cdot \mathbf{l}^3$, onde $\mathbf{l}^2$ e $\mathbf{l}^3$ são linhas quaisquer nas imagens da vista 2 e da vista 3. O mesmo não é válido se forem considerados pontos da imagem $\mathbf{u}^2$ e $\mathbf{u}^3$. Isso conduz a uma restrição trilinear

ponto-linha-linha única. Na verdade, essa é a essência geométrica da restrição trilinear. A restrição por três vistas será apresentada em mais detalhes na subseção 3.8 deste capítulo.

3.4.3 Para quatro vistas

Tem-se que $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ e $\mathbf{d}$ estejam, respectivamente, nas vistas 1, 2, 3, 4. Nesta hipótese existem $3^4 = 81$ escolhas, permitindo 81 restrições quadrilineares. Novamente, pode-se considerar quatro linhas nas imagens $\mathbf{l}^1$, $\mathbf{l}^2$, $\mathbf{l}^3$, $\mathbf{l}^4$, ao invés de pontos $\mathbf{u}^1$, $\mathbf{u}^2$, $\mathbf{u}^3$, $\mathbf{u}^4$, o que conduz a apenas uma restrição quadrilinear nas quatro imagens das linhas. Essa é a essência das restrições quadrilineares. Note que as restrições não necessitam que exista uma linha da cena que se projete nessas imagens de linha. É suficiente que existe um ponto da cena cuja projeção está contida nas linhas das imagens.

3.4.4 Para cinco ou mais vistas

Restrições correspondentes em cinco ou mais vistas podem ser compreendidas e resolvidas com a união de conjuntos de restrições de menos de cinco vistas.

As Restrições Correspondentes encontram-se principalmente no fato que os seus coeficientes podem ser estimados usando as correspondências das imagens. De fato, os pontos das imagens correspondentes (ou linhas) proporcionam restrições lineares para esses coeficientes.

3.5 Ajuste de pacotes

Quando é calculada uma reconstrução projetiva pelas correspondências da imagem, utilizando a Equação (3.5) para encontrar $\mathrm{X_i}$ e $\mathrm{M^j}$, normalmente estão disponíveis mais do que a quantidade mínima de correspondências necessárias. Dessa forma a Equação (3.5), geralmente, não apresenta solução sendo necessário minimizar os erros de retroprojeção, processo similar ao método de estimar a homografia, como o apresentado na subseção 2.3.3 deste capítulo.

$$\min_{\mathbf{X_i},\mathbf{M^j}} \sum_{\mathbf{i=1}}^{\mathbf{m}} \sum_{\mathbf{j=1}}^{\mathbf{n}} \left[\left(\frac{\mathbf{m_1^j X_i}}{\mathbf{m_3^j X_i}} - \hat{\mathbf{u}}_\mathbf{i}^\mathbf{j} \right)^2 + \left(\frac{\mathbf{m_2^j X_i}}{\mathbf{m_3^j X_i}} - \hat{\mathbf{u}}_\mathbf{i}^\mathbf{j} \right)^2 \right], \mathbf{i} = \mathbf{1}, \cdots, \mathbf{m}; \mathbf{j} = \mathbf{1}, \cdots, \mathbf{n} \quad (3.9)$$

Para solucionar esse problema deve-se, primeiro, encontrar um estimador inicial utilizando um método linear e, então, usar uma aproximação por mínimos quadráticos não lineares, como por exemplo, o algoritmo de Levenberg-Marquardt. Os métodos de aproximação por mínimos quadráticos especializados para essa tarefa são conhecidos na fotogrametria como Ajuste de Grupo. Às vezes, este termo também é utilizado, ingenuamente, para outras soluções de problemas dentro da reconstrução geométrica por múltiplas vistas que usam, também, a minimização por mínimos quadráticos não lineares, como por exemplo o estimador de homografia ou a triangulação.

A aproximação não linear por mínimos quadráticos pode ser considerada como proibida computacionalmente para muitos pontos e muitas câmeras. Entretanto, as implementações mais modernas e inteligentes que utilizam as matrizes esparsas tem aumentando significativamente a eficiência do processo (TRIGGS et al., 2000 HARTLEY; ZISSERMAN, 2003).

Não existe um método ótimo para calcular uma reconstrução projetiva utilizando correspondências de muitas imagens. O método a ser escolhido depende fortemente da quantidade de dados a serem utilizados. Um método diferente deve ser usado quando se trabalha com uma sequência de imagens vindas de uma câmera de vídeo, pois as mudanças entre as sequências das imagens são pequenas (FITZGIBBON; ZISSERMAN, 1998), o que pode não ser possível descrever as posições das câmeras ao avaliar os pontos nas imagens semelhantes (CORNELIUS et al., 2004).

Uma aproximação adequada para as sequências de vídeo ou uma grande quantidade de fotografias tiradas em sequência pode ser considerada como: primeiro realiza-se a reconstrução projetiva a partir de duas imagens. Isso pode ser feito estimando a matriz fundamental, decompondo as matrizes de câmera (como apresentado na subseção 3.7) e, então, calcula-se os pontos 3D por triangulação (como apresentado na subseção 3.2) e, finalizando, com o ajuste de grupo. Na sequência, a terceira matriz de câmera é calculada pela técnica de recessão (apresentada na subseção 2.4.3) partindo dos pontos 3D já reconstruídos e dos pontos correspondentes na terceira imagem seguindo do ajuste de grupo. Este último passo é repetido para todos as imagens subsequentes do vídeo.

3.6 Atualizando a reconstrução projetiva com o auxílio da auto calibração

De forma geral a solução da Equação (3.6) traz uma ambiguidade projetiva que é inerente da solução da equação. Entretanto, é possível remover essa ambiguidade sem que sejam conhecidas as informações adicionais. Por outro lado, tendo um conhecimento adicional adequado a respeito da verdadeira cena e/ou das propriedades da câmera é possível definir as restrições que minimizam a quantidade de informações desconhecidas para o processo de reconhecimento da cena, permitindo uma maior similaridade entre o que foi reconstruído e a cena real.

Informações que podem ser adicionadas, como conhecimento *a priori*, que permitem que a ambiguidade projetiva seja redefinida como uma transformada ou afim, ou de similaridade ou euclidiana. Métodos que utilizam conhecimentos adicionais para calcular uma reconstrução de similaridade ao invés de uma projetiva são conhecidos como **auto calibração**, pois isso é, de fato, equivalente a encontrar os parâmetros intrínsecos da câmera (como apresentado na subseção 2.4.1). Os métodos de auto calibração podem ser divididos em dois grupos: Restrições na câmera e restrições na cena. Eles, normalmente, resultam em problemas não lineares. Dessa forma cada um dos métodos necessita de um algoritmo diferente. Exemplo de restrição para câmeras são:

- Restrições ou parâmetros intrínsecos da câmera na calibração da matriz K (como apresentado na subseção 2.4.1)
    - A matriz de calibração K é conhecida para cada câmera. Nesse caso, a cena pode ser reconstruída até uma escala global, com a adição de uma ambiguidade quadrupla. Isso será descrito mais detalhadamente na subseção 3.7.2;
    - As matrizes de calibração intrínseca das câmeras (K) são desconhecidas e deferentes para cada câmera, mas possuem uma forma restrita e sem inclinação (os seus pixels são retangulares). Dessa forma a matriz K pode ser aproximada pela Equação (3.10);

$$K = \begin{bmatrix} f & 0 & -u_0 \\ 0 & g & -v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

- As pesquisas apresentam que é possível reduzir a ambiguidade para uma mera similaridade quando três ou mais vistas da cena estão disponíveis (POLLEFEYS; KOCH; VAN GOOL, 1998; HARTLEY, 1997). O algoritmo fica mais fácil de trabalhar caso seja aumentada a restrição de K fazendo $f = g$. Em outras palavras, seria obrigar os pixels a serem quadrados e fazendo $u_0 = v_0 = 0$, obrigando o ponto principal ser o centro da imagem. Essas restrições, apesar de serem aproximações, são válidas para câmeras reais, pois esse método funciona razoavelmente bem na prática; e
- As matrizes de calibração das câmeras (K) possuem parâmetros intrínsecos desconhecidos, mas são os mesmos para todas as câmeras. Em teoria, isso permite restringir a ambiguidade em uma transformação de similaridade (MAYBANK; FAUGERAS, 1992) utilizando-se as equações de Kruppa. Entretanto, o sistema de equações polinomiais resultante é muito instável e de solução difícil, dessa forma esse método não é utilizado na prática.

- Restrições nos parâmetros extrínsecos da câmera R e $\mathbf{t}$, que são relativos ao movimento das câmeras:
  - São conhecidas rotação R e a translação $\mathbf{t}$ da câmera (HORAUD et al., 1995);
  - É conhecido apenas a rotação R da câmera (HARTLEY, 1994);
  - É conhecida somente a translação $\mathbf{t}$. Para essa restrição Pajdla e Hlaváĉ (PAJDLA; HLAVÁC, 1998) desenvolveram uma solução linear.

Na subseção 2.3.2 foram listadas algumas invariantes do subgrupo das transformações projetivas. Para elas, as restrições da cena podem ser, frequentemente, interpretadas como especificar um número suficiente de invariantes apropriadas para a cena. Assim, é possível recuperar os grupos das transformações correspondentes. Exemplos de restrições para as cenas são:

- A forma mais simples é especificar a coordenada 3D de pelo menos cinco pontos que podem ser identificados na imagem. Sendo que

quatro deles não podem ser coplanares. Denominados $\mathbf{X}_i$ os pontos conhecidos (no mínimo cinco) e a reconstrução deles como $\mathbf{X}'_i$, para $\mathbf{i} = \mathbf{1}, \cdots, 5$, então é possível calcular T usando o sistema $\mathbf{X}'_i \simeq T\mathbf{X}_i$, como descrito na subseção 2.3.3;

- As invariantes afins podem ser suficientes para restringirem a ambiguidade de uma transformação projetiva para uma afim. Isso é equivalente a calcular um plano especial da cena em $\mathcal{P}^3$, denominado de **Plano no Infinito**, no qual todas as linhas paralelas e planas se interpretam. Dessa forma, pode-se especificar certos índices de comprimento para as linhas ou se certas linhas são paralelas na cena;
- Invariantes por similaridade ou métrica podem ser suficientes para restringir a ambiguidade projetiva ou afim para uma ambiguidade de similaridade ou métrica. Isso é equivalente a calcular um cone especial (especial por ser complexo) localizado em um plano do infinito denominado de **Cone Absoluto**. Para fazer isso, pode ser o suficiente especificar um conjunto apropriado de ângulos ou distâncias.

De forma particular, em um ambiente construído pelo homem pode-se utilizar **Pontos de Fuga**. Esses pontos são pontos na imagem localizados no infinito. Eles, normalmente, são três (um vertical e dois horizontais) em direções da cena mutualmente ortogonais.

As restrições da câmera e da cena apresentadas nessa subseção podem ser incorporadas ao ajuste de pacote apresentado na subseção 3.5.

3.7 Visão Estereoscópica Utilizando Duas Câmeras

Uma das grandes vantagens que o sistema de visão humana fornece, ao relacionar com o que já foi apresentado, é a presença de dois olhos e, assim, (*a priori*) é enviado ao cérebro o dobro de informações que uma única imagem fornece, de tal forma que a utilização de duas imagens ligeiramente diferentes permite criar ilusões em 3D. Esse efeito era tão comum que na década de 1950 começaram os primeiros filmes em 3D. Reciprocamente, é de se esperar que em uma cena 3D, em que duas vistas são apresentas à dois olhos diferentes, possa permitir capturar a informação de

profundidade quando a informação é combinada com algum conhecimento a respeito da geometria e/ou posição do sensor (no caso os olhos).

A visão estereoscópica é de grande importância. Ela tem possibilitado uma vasta gama de pesquisas a respeito de sistemas para visão computacional, quando se utiliza dois inputs (sensores) e usa-se o conhecimento da geometria relativa dos sensores. Esses trabalhos possibilitam obter a informação da profundidade da cena ao analisar as diferenças nas imagens.

A calibração de uma câmera e o conhecimento de um ponto na imagem permitem que os pesquisadores determinassem um raio ótico único pertencente à cena. Se duas câmeras calibradas observam o mesmo ponto X da cena, como apresentado na subseção 3.2, é possível calcular as coordenadas 3D desse ponto ao encontrar a interseção dos dois raios ópticos. Esse é o princípio básico da **Visão Estereoscópica** que consiste em três passos simples:

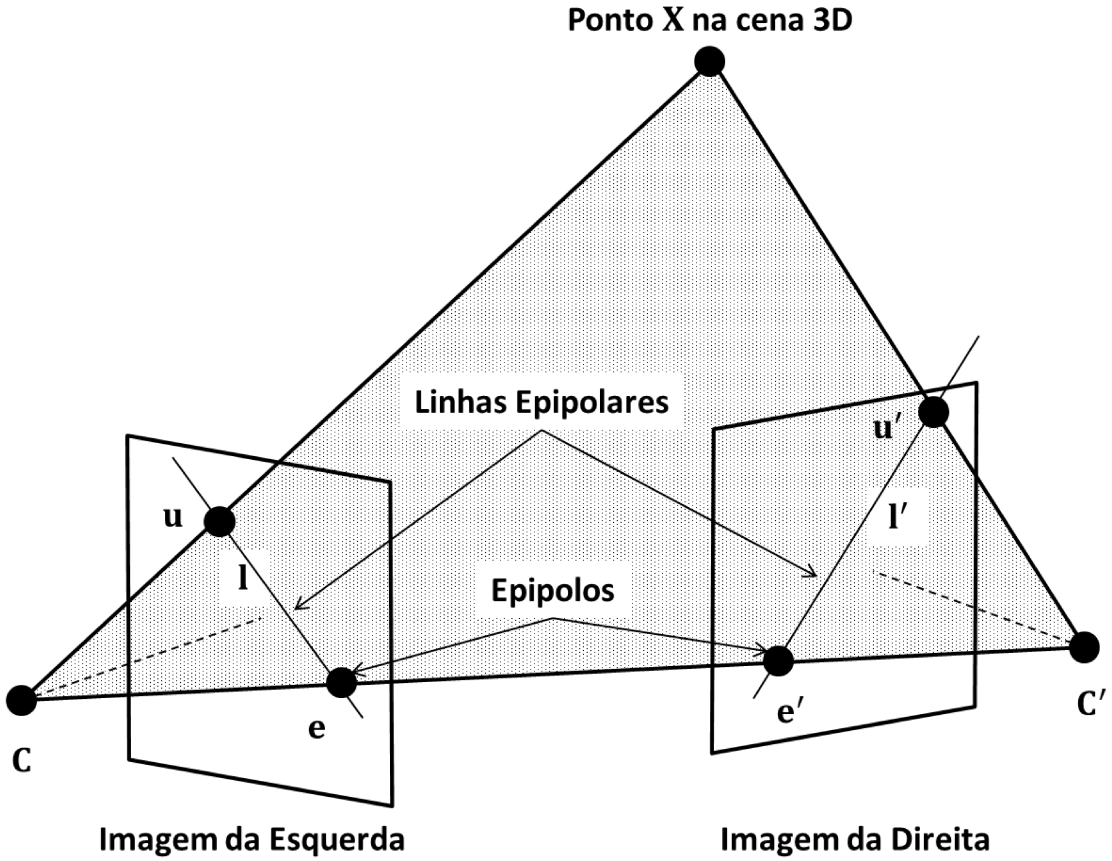
- Calibração da câmera;
- Estabelecer a correspondência dos pontos entre os pares de pontos fornecidos pelas imagens da direita e da esquerda (ou de cima e de baixo); e
- Reconstrução das coordenadas 3D dos pontos da cena.

Para facilitar a compreensão é utilizado a seguinte terminologia, as entidades relacionadas com a primeira imagem (calibração da câmera, pontos, rotação, etc.) é escrita sem apóstrofo, por exemplo, o ponto **u** da imagem. Para as entidades relacionadas com a segunda imagem será utilizado o apóstrofo para diferenciar da primeira imagem, por exemplo o ponto **u'** da imagem.

3.7.1 Geometria epipolar e a sua matriz fundamental

A Figura 3.2 ilustra a geometria de reconhecimento da cena com duas câmeras. Nessa figura a linha que conecta os centros ópticos $\mathbf{C}$ e $\mathbf{C}'$ é a **Linha Base**. Essa linha intercepta os planos da imagem nos pontos $\mathbf{e}$ e $\mathbf{e}'$ denominados de **Epipolos**. Alternativamente, pode-se dizer que um epipolo é a imagem do centro de projeção de uma câmera no plano de imagem da outra. Matematicamente, o epipolo pode ser descrito como $\mathbf{e} = M\mathbf{C}'$ e $\mathbf{e}' = M'\mathbf{C}$.

Figura 3.2 – Geometria do reconhecimento da cena com duas câmeras



Fonte: Autor

Qualquer ponto $\mathbf{X}$ da cena observado por duas câmeras e os pontos do centro óptico das duas câmeras ($\mathbf{C}$ e $\mathbf{C}'$) definem o **Plano Epipolar**. Esse plano intercepta os planos das imagens em algum ponto das linhas $\mathbf{l}$ e $\mathbf{l}'$ denominadas **Linhas Epipolares**. Alternativamente, uma linha epipolar é a projeção dos raios interceptados por uma câmera na outra (intercepção do plano da imagem com o plano epipolar). Todas as linhas epipolares passam pelos epipolos das imagens. A Figura 3.2 ilustra, de forma gráfica, os epipolos, as linhas epipolares e o plano epipolar para um sistema de duas câmeras.

Considere $\mathbf{u}$ e $\mathbf{u}'$ as projeções do ponto $\mathbf{X}$ da cena nas imagens geradas pelas câmeras da esquerda e da direita, respectivamente. O raio que liga $\mathbf{C}$ até $\mathbf{X}$ representa todos as possíveis posições de $\mathbf{X}$ para a imagem da esquerda e na imagem da direita ele é visto como a linha epipolar $\mathbf{l}'$. O ponto $\mathbf{u}'$ na imagem da direita que corresponde ao ponto $\mathbf{u}$ deve, então, estar na linha epipolar $\mathbf{l}'$ da imagem da direita. Isso resulta em $\mathbf{l}'^{\mathbf{T}} \cdot \mathbf{u}' = 0$. Essa situação é, evidentemente, simétrica, dessa forma pode-se

expressar $\mathbf{l}^{T} \cdot \mathbf{u} = 0$. O fato das posições das duas correspondências dos pontos e das imagem não serem arbitrárias é conhecido como **Restrição Epipolar**.

De acordo com a Equação (2.31), o raio da câmera da esquerda é considerado como a retroprojeção do ponto $\mathbf{u}$ da imagem que passa por $\mathbf{C}$ e pelo ponto da cena na forma $\mathbf{X} = M^{+} \cdot \mathbf{u}$. A linha epipolar **l'** é a projeção desse raio na segunda imagem, ou seja, ela passa pelos pontos $M' \cdot \mathbf{C} = \mathbf{e}'$ e $M' \cdot M^{+}\mathbf{u}$. Assim, pode-se relacionar os pontos e a linha epipolar pela Equação (3.11).

$$\mathbf{l}' = \mathbf{e}' \times (M' \cdot M^{+}\mathbf{u}) = S_{(\mathbf{e}')} \cdot M' \cdot M^{+}\mathbf{u} \qquad (3.11)$$

Na Equação (3.11) o produto vetorial foi substituído pelo produto matricial, como definido pela Equação (2.15). Dessa forma, pode-se mostrar que a linha epipolar $\mathbf{l}'$ é um mapeamento linear do ponto $\mathbf{u}$ correspondente. A matriz que representa esse mapeamento linear pode ser representada pela Equação (3.12).

$$F = S_{(\mathbf{e}')} \cdot M' \cdot M^{+} \qquad (3.12)$$

Dessa forma, pode-se simplificar a relação dos pontos com as linhas epipolares como mostrado na Equação (3.13).

$$\mathbf{l}' = F \cdot \mathbf{u} \qquad (3.13)$$

Se desejasse uma restrição em um determinado ponto em duas imagens, pode-se utilizar $\mathbf{l}'^{T} \cdot \mathbf{u}' = 0$ isso resulta na Equação (3.14).

$$\mathbf{u}'^{\mathrm{T}} \cdot F \cdot \mathbf{u} = 0 \qquad (3.14)$$

A Equação (3.14) é a representação na forma algébrica da **Restrição Epipolar**. A relação apresentada foi desenvolvida por Longuet-Higgins (LONGUET-HIGGINS, 1981). Ele foi o primeiro, dos estudiosos do campo da visão computacional, a descobri essa relação bilinear. A matriz *F* é chamada de **Matriz Fundamental** devido a fatos históricos. Porém, alguns autores preferem chamá-la de **Matriz Bifocal**.

A transposta da Equação (3.14) monstra que se as câmeras forem trocadas, então a matriz bifocal é substituída pela sua transposta.

Como M e M' são matrizes de ranking cheio e $S_{(\mathbf{e}')}$ possui ranking 2, pela Equação (3.12), F também é de ranking 2. Um mapeamento linear que mapeia pontos para linhas é chamado de uma **Correlação Projetiva**. Uma correlação projetiva é uma colinearização de um espaço projetivo em seu espaço dual, conduzindo os pontos para o hiperplano e preservando as incidências. No caso da geometria epipolar, a correlação projetiva mostrada na Equação (3.13) é singular. Em outras palavras, os pontos não-colineares são mapeados em linhas com uma interseção comum, pois $\mathbf{e}' \cdot$

$S_{(\mathbf{e}')} = 0$ e a Equação (3.12) requer que $\mathbf{e}' \cdot F = \mathbf{0}^{\mathrm{T}}$. Ao trocar as imagens, obtém-se a relação simétrica $F \cdot \mathbf{e} = 0$. Dessa forma, os epipolos são os vetores nulos da esquerda e da direita de F.

A matriz fundamental é um elemento muito importante para o estudo da geometria por múltiplas vistas, pois os seus valores capturam todas as informações que podem ser obtidas a respeito de um par de câmeras a partir das correspondências nas imagens.

3.7.1.1 Matriz fundamental a partir de matrizes de câmeras com forma restrita

A Equação (3.12) fornece condições para calcular *F* utilizando duas matrizes de câmera (M e M') arbitrárias. Entretanto, às vezes as matrizes de câmera apresentam uma forma restrita. Existem dois casos importantes em que essas restrições simplificam a Equação (3.12).

No primeiro caso as matrizes de câmera possuem a forma apresentada na Equação (3.15).

$$M = [\boldsymbol{I}|\mathbf{0}], M' = \left[\widetilde{M}'|\mathbf{e}'\right] \tag{3.15}$$

Esta forma da matriz da câmera pode ser justificada pelo que foi apresentado na subseção 3.3 quando foi mostrada a ambiguidade projetiva. Dessa forma a primeira matriz de câmera pode ser sempre escolhida como $M = [\boldsymbol{I}|\mathbf{0}]$. Assim, se o primeiro centro de projeção $\mathbf{C}$ satisfaz a condição de $M\mathbf{C} = 0$, ele está localizado na origem com $\mathbf{C} = [0,0,0,1]^{\mathrm{T}}$. Se a segunda matriz de câmera, M', satisfaz $M'\mathbf{C} = \mathbf{e}'$, então a última coluna de M' é, necessariamente, o segundo epipolo, como apresentado na Equação (3.15). Substituindo essas informações na Equação (3.15) e utilizando $M^{+} = [\boldsymbol{I}|\mathbf{0}]^{\mathrm{T}}$ a matriz fundamental pode ser escrita como mostrado na Equação (3.16).

$$F = S_{(\mathbf{e}')}\widetilde{M}' \tag{3.16}$$

No segundo caso as matrizes de câmera possuem a forma mostrada na Equação (3.17).

$$M = K[\boldsymbol{I}|\mathbf{0}], M' = K'[R| \quad -R\mathbf{t}] \tag{3.17}$$

A Equação (3.17) descreve as câmeras calibradas com os parâmetros intrínsecos da câmera nas matrizes de calibração K e K', além do movimento relativo da rotação R e da translação $\mathbf{t}$. Assim, tem-se a Equação (3.18).

$$M^{+} = \begin{bmatrix} K^{-1} \\ \mathbf{0}^{T} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \tag{3.18}$$

Juntando as informações mostradas, $F = S(M' \cdot \mathbf{C})M' \cdot M^{+} = S(-K' \cdot R \cdot \mathbf{t})K' \cdot R \cdot K^{-1}$, combinadas com $S(H\mathbf{u}) \simeq H^{-T}S(\mathbf{u})H^{-1}$, para todo $\mathbf{u}$ e H não singulares, obtém-se a Equação (3.19)

$$F = K'^{-\mathrm{T}}RS_{(\mathbf{t})}K^{-1} \tag{3.19}$$

3.7.2 Movimento relativo da câmera e a matriz essencial

Se a matriz da câmera pode ser expressa pela Equação (3.17) e se os parâmetros intrínsecos correspondem às matrizes K e K' são conhecidas, então pode-se compensar a transformação afim feita por K e K'. Na subseção 2.4.1 e na Figura 2.12 foi apresentado diversos sistemas de coordenadas para uma única câmera. Foi mostrado o sistema de coordenadas euclidiano escrito com o subscrito i e o ponto de medição $\mathbf{u}_i$ pertencente a esse sistema. O sistema de coordenadas afim foi escrito sem o subscrito. De acordo com essa convenção, pode-se expressar a transformação afim pelas Equações (3.20) e (3.21).

$$\mathbf{u} = K^{-1}\mathbf{u}_i \tag{3.20}$$

$$\mathbf{u}' = (K')^{-1}\mathbf{u}_i' \tag{3.21}$$

Utilizando a Equação (3.19) e a restrição epipolar, então a Equação (3.19) pode ser escrita em função de $\mathbf{u}_i$ e $\mathbf{u}_i'$ transformando-se na Equação (3.22).

$$\mathbf{u}_i'^{\mathrm{T}} \cdot E \cdot \mathbf{u}_i = 0 \tag{3.22}$$

A matriz E pode ser expressa pela Equação (3.23).

$$E = R \cdot S_{(\mathbf{t})} \tag{3.23}$$

E é conhecida como **Matriz Essencial**.

A restrição epipolar na forma $\mathbf{u}_i'^{\mathrm{T}} \cdot R \cdot S_{(\mathbf{t})} \cdot \mathbf{u}_i = 0$ possui um significado geométrico simples. Se for observado pelo sistema de coordenadas afim os vetores $\mathbf{u}_i$ e $\mathbf{u}_i'$ representam pontos 2D homogêneos na imagem. Se eles forem observados pelo sistema de coordenadas euclidiano esses vetores representam pontos 3D não-homogêneos. Estas duas representações são equivalentes, pois o problema refere-se a uma transformação linear entre dois sistemas de coordenadas. E a restrição epipolar apresenta que os vetores de três coordenadas $\mathbf{u}_i$, $R^{-1}\mathbf{u}_i'$ e $\mathbf{t}$ são coplanares. Três vetores $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ são coplanares se, e somente se, $\det(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \mathbf{a}^{\mathrm{T}}(\mathbf{b} \times \mathbf{c}) = 0$.

A matriz essencial possui ranking dois, isso significa que existe, exatamente, dois de seus valores singulares diferentes de zero. Diferente da matriz fundamental, a matriz essencial satisfaz um critério de restrição adicional de tal forma que esses dois valores singulares devem ser iguais. Isso ocorre porque os valores singulares de uma matriz são invariantes a uma transformação ortonormal da matriz e, dessa forma, a decomposição SVD pode ser expressa como $E = U \cdot D \cdot V^{\mathrm{T}}$ e tem-se a Equação (3.24).

$$D = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathrm{diag}[\sigma, \sigma, 0] \tag{3.24}$$

3.7.2.1 Decomposição da matriz essencial em rotação e translação

A matriz essencial E captura as informações a respeito do **movimento relativo** da segunda câmera em relação à primeira, descrito pela translação $\mathbf{t}$ e pela rotação R. Conhecido as matrizes de calibração das câmeras K e K'**,** o movimento relativo pode ser calculado pelos pontos correspondentes nas imagens seguindo os passos:

1. Estimar a matriz fundamental F pelos pontos correspondentes (subseção 3.7.4 deste capítulo);
2. Calcular $E = K'^{\mathrm{T}} \cdot F \cdot K$; e

3. Decompor E em $\mathbf{t}$ e R.

Como apresentado na subseção 3.2, opcionalmente, pode-se reconstruir os pontos 3D a partir dos pontos correspondentes da imagem pela triangulação.

Falta mostrar como pode-se decompor E em $\mathbf{t}$ e R. Se a matriz essencial E é determinada até uma escala desconhecida, como o que acontece no caso em que ela é estimada pelos pontos correspondentes entre as imagens. Como mostrado na Equação (3.23), a escala de $\mathbf{t}$ é desconhecida também. Isso significa que é possível reconstruir as câmeras e os pontos da cena até uma transformação de similaridade utilizando as Equações (3.25) e (3.26).

$$\bar{\mathbf{t}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.25}$$

$$\bar{R} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.26}$$

Assim, tem-se que $\bar{R}$ é uma matriz de rotação e que $\bar{R} \cdot S_{(\bar{\mathbf{t}})} = -\bar{R}^{\mathrm{T}} \cdot S_{(\bar{\mathbf{t}})} = \mathrm{diag}[1,1,0]$. Ao considerar $E \simeq U \cdot \mathrm{diag}[1,1,0]V^{\mathrm{T}}$ ser a decomposição SVD de E, tem-se que a translação pode ser calculada como mostrada na Equação (3.27).

$$S_{(\mathbf{t})} = V \cdot S_{(\bar{\mathbf{t}})} \cdot V^{\mathrm{T}} \tag{3.27}$$

A matriz de rotação não é obtida de forma única, dessa forma pode-se ter a Equação (3.28).

$$R = U \cdot \bar{R} \cdot V^{\mathrm{T}} \text{ ou } R = U \cdot \bar{R}^{\mathrm{T}} \cdot V^{\mathrm{T}} \tag{3.28}$$

Hartley (HARTLEY, 1992; HARTLEY, 1997) demonstrou que $R \cdot S_{(\mathbf{t})} \simeq U \cdot \mathrm{diag}[1,1,0]V^{\mathrm{T}} \simeq E$ e provou que não existe outra decomposição que pode ser encontrada.

A escala de ambiguidade de $\mathbf{t}$ também inclui o sinal de $\mathbf{t}$. Dessa forma, no todo tem-se quatro movimentos relativos qualitativamente diferentes, devido a duas ambiguidades da rotação e duas ambiguidades da translação.

3.7.3 Decomposição da matriz fundamental na matriz de câmera

Na subseção 3.3 foi proposto encontrar uma solução particular para o problema da reconstrução projetiva a partir de duas imagens, mostrado na Equação (3.5). Ou seja, encontrar a matriz da câmera e os pontos da cena que são projetados para formar os pontos da imagem resultante. Isto pode ser realizado ao estimar a matriz fundamental pelos pontos da imagem, decompô-la em duas matrizes de câmera e, então, calcular os pontos da cena usando a triangulação (subseção 3.2).

Nesta seção é apresentado como decompor F em duas matrizes de câmera M e M' de forma consistente com a imagem e a cena. Sabe-se, pela subseção 3.3, que devido à ambiguidade, projetiva a primeira matriz de câmera pode ser escolhida como $M = [I|\mathbf{0}]$ sem perda de generalidade. Agora, falta determinar a matriz M'.

Desmembrando a matriz S tem simetria de inclinação se ela satisfaz a condição de $S + S^{\mathrm{T}} = 0$. Pode-se dizer que qualquer matriz S que satisfaça a condição de $\mathbf{X}^{\mathrm{T}} \cdot S \cdot \mathbf{X} = 0$, para qualquer $\mathbf{X}$, possui simetria de inclinação. Para comprovar isto, é suficiente escrever o produto matricial na forma da Equação (3.29).

$$\mathbf{X}^{\mathrm{T}} \cdot S \cdot \mathbf{X} = \sum_i s_{ii} X_i^2 + \sum_{i<j} (s_{ij} + s_{ji}) X_i X_j = 0 \qquad (3.29)$$

Em que s_{ij} são as entradas de S. A Equação (3.29) é válida para qualquer $\mathbf{X}$ somente se todos os números s_{ii} e $s_{ij} + s_{ji}$ são zero.

Substituindo $\mathbf{u} = M\mathbf{X}$ e $\mathbf{u}' = M'\mathbf{X}$ em $\mathbf{u}'^{\mathrm{T}} \cdot F \cdot \mathbf{u} = 0$ resulta na Equação (3.30).

$$M'^{\mathrm{T}} \cdot F \cdot M = \begin{bmatrix} \bar{M}'^{\mathrm{T}} \\ \mathbf{b}'^{\mathrm{T}} \end{bmatrix} F[I|\mathbf{0}] = \begin{bmatrix} \bar{M}'^{\mathrm{T}} F & \mathbf{0} \\ \mathbf{b}'^{\mathrm{T}} F & 0 \end{bmatrix} \qquad (3.30)$$

Como a matriz mais à direita deve ter simetria de inclinação, $\bar{M}'^{\mathrm{T}} F$ deve ser simétrica em relação à sua inclinação. Assim, $\mathbf{b}'^{\mathrm{T}} F$ deve tender a zero. Finalmente, tem-se que $\mathbf{b}'$ torna-se o segundo epipolo, $\mathbf{e}'$. Isso foi mostrado na justificativa da Equação (3.15).

É fácil verificar que se $\bar{M}' = S \cdot F$, onde S é uma matriz arbitrária 3×3 e com simetria de inclinação, então $\bar{M}'^{\mathrm{T}} F$ também tem simetria de inclinação. Para verificar isso, é suficiente escrever $\bar{M}'^{\mathrm{T}} F = -F^{\mathrm{T}} \cdot S \cdot F$ e verificar que $(F^{\mathrm{T}} \cdot S \cdot F) + (F^{\mathrm{T}} \cdot S \cdot$

$F)^{\mathrm{T}} = 0$. Dessa forma, pode-se escolher $S = S_{(\mathbf{e}')}$ que fornecerá simplificações convenientes.

Resumindo, as matrizes de câmera consistentes com uma matriz fundamental F que pode ser escolhida como mostrado nas Equações (3.31) e (3.32).

$$M = [I|\mathbf{0}] \tag{3.31}$$

$$M' = \left[S_{(\mathbf{e}')} \cdot F \mid \mathbf{e}'\right] \tag{3.32}$$

As Equações (3.31) e (3.32) mostram que é possível verificar que, mesmo que a primeira câmera é fixada em $M = [I|\mathbf{0}]$, a segunda matriz M' não é determinada de forma única por F, pois existe a liberdade da escolha de S.

3.7.4 Estimando a matriz fundamental através de pontos

A geometria epipolar possui sete graus de liberdade (MOHR, 1993). Os epipolos $\mathbf{e}$, $\mathbf{e}'$ na imagem possuem, cada um, duas coordenadas, o que resulta em quatro graus de liberdade, e os outros três graus de liberdade surgem do mapeamento de qualquer uma das três linhas epipolares da primeira imagem na segunda. Alternativamente, pode-se notar que as nove componentes de F são conhecidas até uma determinada escala e, assim, temos uma nova restrição em que $\det(F) = 0$, isso conduz à mesma situação de $9 - 1 - 1 = 7$ parâmetros livres.

A correspondência de sete pontos nas imagens da direita e da esquerda permite calcular a matriz fundamental F utilizando um algoritmo não linear (FAUGERAS; LUONG; MAYBANK, 1992) conhecido como **O Algoritmo dos Sete Pontos**. Entretanto, se oito pontos estão disponíveis, a solução torna-se linear e o método passa a ser conhecido como **O Algoritmo dos Oito Pontos**. Diferente do algoritmo dos sete pontos, o algoritmo dos oito pontos pode ser estendido, diretamente, para a utilização de mais de oito pontos.

3.7.4.1 Algoritmo dos oito pontos

À primeira vista, o esboço do algoritmo dos oito pontos pode ser visto da seguinte forma. Considere $m > 8$ pares de pontos $(\mathbf{u}_i, \mathbf{u}_i')$ em coordenadas homogêneas, soluciona-se o sistema de equações da Equação (3.33).

$$\mathbf{u}_i'^{\mathrm{T}} \cdot F \cdot \mathbf{u}_i = 0, i = 1, \cdots, m \tag{3.33}$$

A solução desse problema é muito similar à estimação homográfica. De forma similar ao apresentado na subseção 2.3.3, pode-se utilizar a identidade $\mathbf{u}'^{\mathrm{T}} \cdot F \cdot \mathbf{u} = [\mathbf{u}' \circledast \mathbf{u}'^{\mathrm{T}}]\mathbf{f} = [u \cdot u' \quad u \cdot v' \quad u \cdot w' \quad v \cdot u' \quad v \cdot v' \quad v \cdot w' \quad w \cdot u' \quad w \cdot v' \quad w \cdot w']$, onde $\mathbf{f} = [f_{11}, f_{21}, \cdots, f_{23}, f_{33}]^{\mathrm{T}}$ e $\circledast$ é o produto matricial de Kronecker. Considerando todas as m correspondências obtêm-se a Equação (3.34).

$$\begin{bmatrix} \mathbf{u}_1^{\mathrm{T}} \circledast \mathbf{u}_1'^{\mathrm{T}} \\ \cdots \\ \mathbf{u}_m^{\mathrm{T}} \circledast \mathbf{u}_m'^{\mathrm{T}} \end{bmatrix} \mathbf{f} = W \cdot \mathbf{f} = 0 \tag{3.34}$$

Para as oito correspondências, com uma configuração não degenerativa, o sistema tem uma única solução (até uma determinada escala). Para mais correspondências pode-se solucionar o sistema usando SVD e pela minimização das distâncias algébricas. Para isso, os pontos da imagem devem estar pré-organizados, como descrito na subseção 2.3.3.

A matriz fundamental F, calculada pelo algoritmo dos oito pontos, é, de forma geral não singular, ou seja, não é uma matriz fundamental válida. Pode-se encontrar uma matriz F de ranking 2 que se aproxima de F em relação à norma de Frobenius seguindo as etapas:

1. Decompor $F = U \cdot D \cdot V^{\mathrm{T}}$ utilizando a SVD;
2. Fazer o menor valor singular na matriz diagonal D igualando a zero;
3. Utilizar a nova matriz diagonal $\overline{D}$ com apenas duas entradas diferentes de zero, fazer a decomposição inversa de $\overline{F} = U \cdot \overline{D} \cdot V^{\mathrm{T}}$.

3.7.4.2 Algoritmo dos sete pontos

Se apenas $m = 7$ pontos correspondentes estão disponíveis, a solução do sistema da Equação (3.33) é um subespaço linear de duas dimensões de $\mathcal{R}^9$, diferente do que ocorre para o caso de $m = 8$ em que a solução é um subespaço linear de uma única dimensão. Ou seja, na solução de $m = 7$ existem dois vetores $\mathbf{f}$ e $\mathbf{f}'$ que

satisfazem $W\mathbf{f} = W\mathbf{f}' = 0$. A decomposição SVD conduz a esses dois vetores mutualmente ortonormais.

O objetivo do algoritmo dos sete pontos é encontrar os pontos neste subespaço que satisfazem a restrição $\det(F) = 0$. Dessa forma a Equação (3.35) é usada para encontrar um escalar λ.

$$\det[\lambda F + (1-\lambda)F'] = 0 \qquad (3.35)$$

Em geral, a equação cúbica (3.35) possui, em geral, três soluções. Entretanto, até duas delas podem ser complexas. Assim, o algoritmo dos sete pontos pode apresentar uma, duas ou, até mesmo, três soluções diferentes para F.

Se seis ou sete pontos são relacionados por uma homografia, então existe uma infinidade de espaços que são soluções de F. Em outras palavras, esta é uma configuração degenerativa para calcular F (HARTLEY; ZISSERMAN, 2003).

3.7.4.3 Estimativa pela máxima verossimilhança para a estimação da matriz fundamental

A estimativa pela máxima verossimilhança utilizada neste trabalho é muito semelhante àquela usada para a homografia, entretanto, para esta etapa, é utilizado uma restrição ligeiramente diferente nas correspondências e, também, uma restrição adicional, $\det(F) = 0$.

Considere $[\hat{u}_i, \hat{v}_i]^{\mathrm{T}}$ e $[\hat{u}'_i, \hat{v}'_i]^{\mathrm{T}}$ os pontos da imagem em coordenadas não-homogêneas. Dessa forma, resolvendo o problema de otimização, tem-se a Equação (3.36).

$$\underset{F,u_i,v_i,u'_i,v'_i}{\mathrm{Min}} \sum_{i=1}^{m}[(u_i-\hat{u}_i)^2+(v_i-\hat{v}_i)^2+(u'_i-\hat{u}'_i)^2+(v'_i-\hat{v}'_i)^2],$$
$$i=1,\cdots,m,[u'_i,v'_i,1]F[u_i,v_i,1]^{\mathrm{T}}=0,\det(F)=0 \qquad (3.36)$$

Uma alternativa frequentemente utilizada é, primeiro, decompor F em matrizes de câmera, reconstruir os pontos da cena por triangulação (subseção 3.2) e, então utilizar o ajuste de conjunto (subseção 3.5). Não é nenhum obstáculo que a otimização seja realizada utilizando mais variáveis do que no problema de otimização mostrado na Equação (3.36).

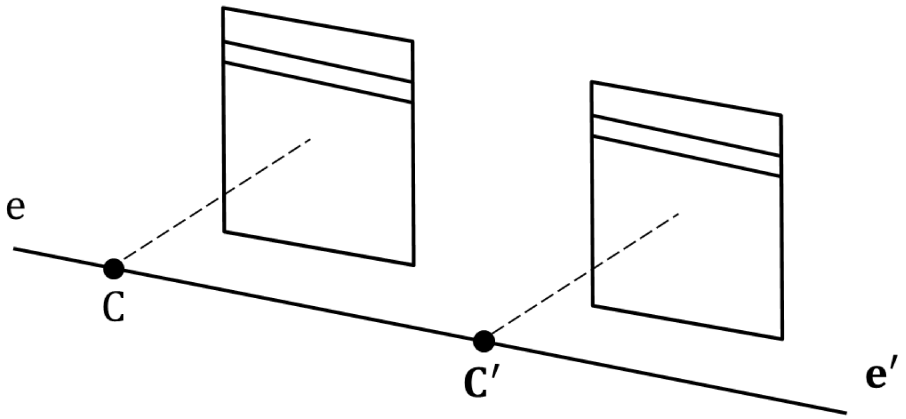
3.7.5 Configuração retificada de duas câmeras

A restrição epipolar reduz o número de dimensões do espaço de busca para uma correspondência simples entre u e u' reduzindo a varredura 2D na imagem para uma varredura 1D.

Um arranjo especial para o anel de câmera estéreo é conhecido como configuração retificada. Nessa configuração, os planos da imagem coincidem e a linha $\mathbf{CC}'$ é paralela a eles. Dessa forma, como pode ser observado na Figura 3.3, os epipolos são projetados no infinito e, também, a linha epipolar coincide com as linhas da imagem. Pode-se considerar, também, que os parâmetros da calibração intrínseca para ambas as câmeras são os mesmos. Para a configuração retificada os cálculos a serem realizados são mais simples. Essa configuração é utilizada frequentemente quando a correspondência estéril deve ser definida por um operador humano que necessita encontrar os pontos correspondentes de forma fácil (esta aproximação não-automática ainda é utilizada em fotogrametria e sensoriamento remoto). Este procedimento também é mais fácil para os computadores, pois é mais fácil rastrear as correspondências ao longo das linhas horizontais definidas do que em linhas não bem definidas.

A transformação geométrica que modifica uma configuração geral de câmera com linhas epipolares não paralelas para a configuração retificada é chamada de retificação da imagem. Considerando uma configuração retificada, pode-se recuperar a profundidade da cena ao verificar que os eixos ópticos são paralelos. Dessa forma obtém-se a noção de **disparidade**, que é frequentemente utilizada na literatura estereoscópica.

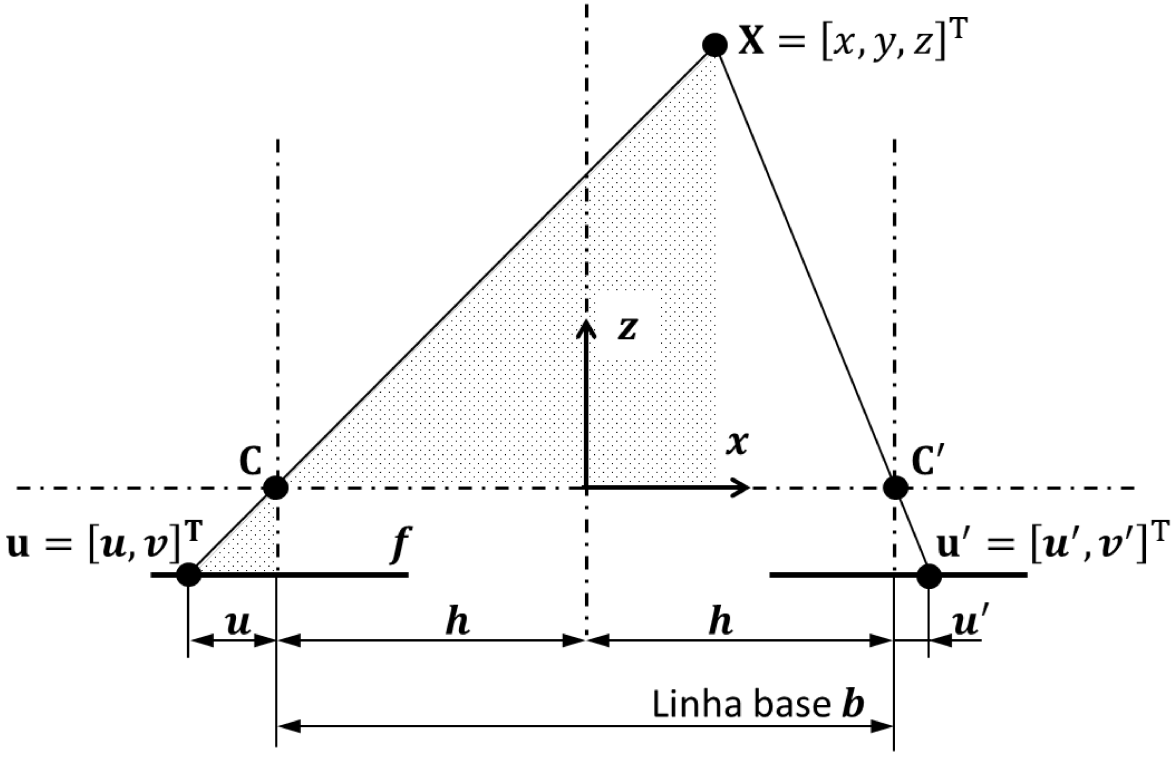
Figura 3.3 – Configuração retificada de duas câmeras onde as linhas epipolares são paralelas nas imagens e os epipolos localizados no infinito



Fonte: Autor

O diagrama simples na Figura 3.4 ilustra como proceder. Na Figura 3.4 tem-se uma visão panorâmica de duas câmeras com os eixos ópticos paralelos e separados por uma linha de base com distância $b = 2h$. Nas duas imagens fornecidas pelas câmeras um ponto $\mathbf{X}$, de coordenadas $(x, y, z)^{\mathrm{T}}$ na cena, é apresentado na imagem da esquerda como u e na da direita como u'. As coordenadas observadas na Figura 3.4 possuem o eixo z representando a distância a partir do eixo óptico das câmeras (o foco das câmeras localizados em $z = 0$) e o eixo x representando as distâncias horizontais. A coordenada y foi omitida para simplificação do desenvolvimento. A posição correspondente a $x = 0$ é considerada como estando no ponto médio entre as câmeras. Cada imagem da câmera possui um sistema de coordenadas, que para simplificação, possui sua origem no centro das imagens, isso implica em uma translação simples do sistema de coordenadas global. Ainda na Figura 3.4, os valores u, u', v, v' fornecem as coordenadas dos pontos em relação ao sistema de coordenadas local de cada imagem de cada câmera ($[u, v]^{\mathrm{T}}$ são coordenadas da câmera da esquerda e $[u', v']^{\mathrm{T}}$ são os da câmera da direita). Devido à simplificação possível pela configuração retificada as medidas são realizadas nas mesmas linhas (alturas), ou seja, $v = v'$.

Figura 3.4 – Geometria estereográfica elementar na configuração retificada



Fonte: Autor

Na Figura 3.4 a profundidade A profundidade $\boldsymbol{z}$ do ponto $\mathbf{X}$ na cena 3D pode ser calculado a partir da disparidade $\boldsymbol{d} = \boldsymbol{u}' - \boldsymbol{u}$. Valores de $\boldsymbol{u}$ e $\boldsymbol{u}'$ são medidos na mesma altura, ou seja, $\boldsymbol{v} = \boldsymbol{v}'$.

Ainda na Figura 3.4 observar-se notar a existência de uma disparidade d entre u e u'. Isto ocorre devido à diferença entre as posições das câmeras, ou seja, $d = u - u'$, $d < 0$. Dessa forma, pode-se utilizar a geometria elementar para tentar deduzir a profundidade z da coordenada $\mathbf{X}$.

De acordo com a Figura 3.4 a reta que passa por $\mathbf{u}$ e $\mathbf{C}$ ou $\mathbf{C}$ e $\mathbf{X}$ são hipotenusas similares de triângulos retângulos. Temos também que as medidas h e a distância focal f são valores positivos, z é uma coordenada positiva e x, u e u' são coordenadas que podem ser positivas ou negativas. Assim, pode-se descrever essas relações como mostrado nas Equações (3.37) e (3.38).

$$\frac{u}{f} = -\frac{h+x}{z} \tag{3.37}$$

$$\frac{u'}{f} = \frac{h-x}{z} \tag{3.38}$$

Eliminando a variável x nas Equações (3.37) e (3.38), resultando em $z(u' - u) = 2h \cdot f$ e, com isso, obtém-se a Equação (3.39).

$$Z = \frac{2h \cdot f}{u' - u} = \frac{b \cdot f}{u' - u} = \frac{b \cdot f}{d} \tag{3.39}$$

É possível verificar na Equação (3.39) que $d = u' - u$ é a disparidade detectada na observação de $\mathbf{X}$. Se $(u' - u) \to 0$, então $z \to \infty$. A disparidade nula ($d = 0$) indica que o ponto está (efetivamente) a uma distância infinita do observado. Pontos 3D distantes possuem uma disparidade pequena. O erro relativo na profundidade z é grande para pequenas disparidades e uma grande linha base reduz o erro relativo em z.

As outras duas coordenadas 3D do ponto $\mathbf{X}$ podem ser calculadas como mostrado nas Equações (3.40) e (3.41).

$$x = \frac{-b(u+u')}{2d} \tag{3.40}$$

$$y = \frac{b \cdot v}{d} \tag{3.41}$$

3.7.6 Cálculo da retificação

Foi apresentado que a geometria estereoscópica necessita que os pontos correspondentes possam ser buscados em um espaço 1D ao longo da linha epipolar. Também foi mencionado que um par de câmeras na configuração retificada facilita a busca pelas correspondências estereoscópicas. Dessa forma, é possível aplicar um caso especial de transformação geométrica, exceto nos casos degenerativos, chamada **retificação da imagem** para imagens capturadas por um anel estereoscópico em que as câmeras possuam eixos ópticos não paralelos. O resultado dessa transformação é um conjunto de imagens com as linhas epipolares paralelas.

Os valores para a câmera da esquerda são identificados com o subscrito $_E$ e os valores da câmera da direita pelo subscrito $_D$. O sobrescrito * é utilizado para descrever os valores após a retificação. O procedimento de retificação consiste em dois passos, nos quais as matrizes K_E e K_D são as matrizes de calibração intrínseca da câmera da esquerda e da direita, respectivamente. Os passos do procedimento de retificação são:

1. Encontrar um par de homografias retificadoras K_E e K_D para as imagens da esquerda e da direita, respectivamente. Esse par deve ser tal que as linhas epipolares são equivalentes e, também, paralelas às linhas da imagem; e
2. Distorcer controladamente (rotacionar e redimensionar) as imagens e modificar as matrizes de projeção das câmeras. As imagens devem ser rotacionadas utilizando as homografias K_E e K_D e mais as matrizes de projeção das câmeras modificas como $M_E^* = H_E \cdot M_E$ e $M_D^* = H_D \cdot M_D$.

As câmeras são retificadas pela homografia, como mostrado nas Equações (3.42) e (3.43).

$$M_E^* = H_E \cdot M_E = H_E \cdot K_E \cdot R_E \cdot [I \mid -\mathbf{C}_E] \tag{3.42}$$

$$M_D^* = H_D \cdot M_D = H_D \cdot K_D \cdot R_D \cdot [I \mid -\mathbf{C}_D\,] \tag{3.43}$$

Considerando $\mathbf{e}_E$ e $\mathbf{e}_D$ os epipolos das imagens da esquerda e da direita. De forma análoga, tem-se $\mathbf{l}_E$ e $\mathbf{l}_D$ como as linhas epipolares e $\mathbf{u}_E$ e $\mathbf{u}_D$ as projeções de um ponto da cena nos planos da imagem. Seja F^* a matriz fundamental correspondente às imagens retificadas com $\lambda \neq 0$. A condição necessária para a retificação que torna as linhas epipolares coincidentes com as linhas em ambas as imagens é mostrado nas Equações (3.44) e (3.45).

$$\mathbf{l}_D^* = \mathbf{e}_D^* \otimes \mathbf{u}_D^* = \lambda F^* \mathbf{u}_E^* \tag{3.44}$$

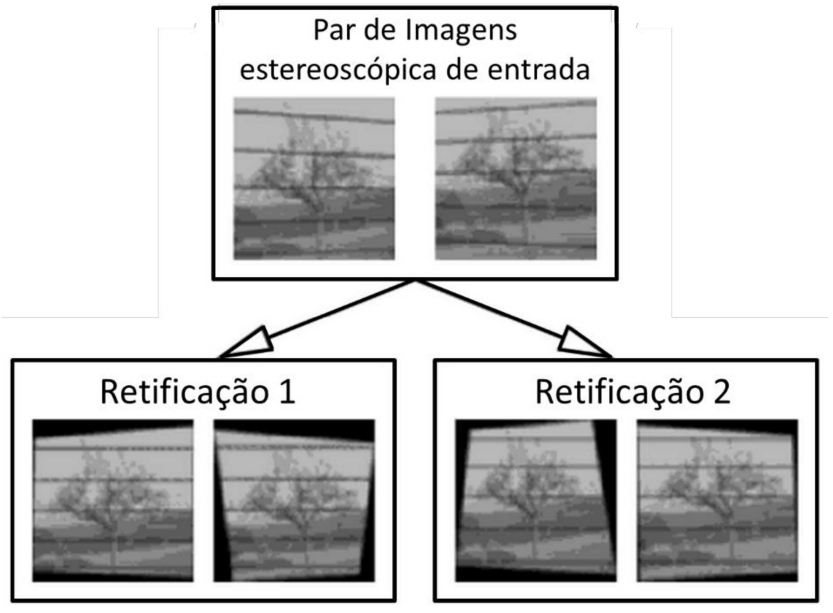
$$M_D^* = H_D \cdot M_D = H_D \cdot K_D \cdot R_D \cdot [I \mid -\mathbf{C}_D\,] \tag{3.45}$$

Onde:

$$F^* \simeq \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \tag{3.46}$$

A homografia retificadora não é única. Dois casos de retificação são apresentados na Figura 3.5. O problema que deve ser solucionado agora é escolher qual das várias retificações encontradas é a melhor.

Figura 3.5 – Dois casos possíveis de retificação devido a ambiguidade do processo



Fonte: Autor

3.7.6.1 Algoritmo de retificação da imagem

A retificação das imagens pode ser realizada conforme dos seguintes passos:

1. Os epipolos de ambas as imagens são transladados para o infinito. Seja $\mathbf{e}_E = [e_1, e_2, 1]^{\mathrm{T}}$ o epipolo da imagem da esquerda e $e_1^2 + e_2^2 \neq 0$. Como mostrado na Equação (3.47) esse epipolo é mapeado em $\mathbf{e}^* \simeq [1,0,0]^{\mathrm{T}}$ como uma rotação do epipolo $\mathbf{e}_E$ em relação ao eixo u e a projeção

$$\widehat{H}_E \simeq \begin{bmatrix} e_1 & e_2 & 0 \\ -e_2 & e_1 & 0 \\ -e_1 & -e_2 & e_1^2 + e_2^2 \end{bmatrix} \tag{3.47}$$

2. As linhas epipolares são unificadas para ter um par elementar de homografias retificadoras. Uma vez que $\mathbf{e}_D^* = [1,0,0]^{\mathrm{T}}$ e ambos os espaços são nulos (esquerdo e direito) de $\hat{F}$, a matriz fundamental modificada pode ser escrita como mostrado na Equação (3.48):

$$\hat{F} \simeq \begin{bmatrix} 0 & 0 & 0 \\ 0 & \alpha & \beta \\ 0 & \gamma & \delta \end{bmatrix} \tag{3.48}$$

Como mostrado nas Equações (3.49) e (3.50) as homografias retificadoras elementares $\widehat{H}_E$ e $\widehat{H}_D$ são escolhidas de forma a fazer $\alpha = \delta = 0$ e $\beta = -\gamma$; e

$$\overline{H}_E = H_S \widehat{H}_E \tag{3.49}$$

$$\overline{H}_D = \widehat{H}_D \tag{3.50}$$

Onde:

$$H_S \simeq \begin{bmatrix} \alpha\delta - \beta\gamma & 0 & 0 \\ 0 & -\gamma & -\delta \\ 0 & \alpha & \beta \end{bmatrix} \tag{3.51}$$

Então,

$$F^* = \left(\widehat{H}_D\right)^{-\mathrm{T}} F \left(H_S \widehat{H}_E\right)^{-1} \tag{3.52}$$

3. Um par de homografias ótimas é escolhido do conjunto que preserva a matriz fundamental F^*.

Considere $\overline{H}_E$ e $\overline{H}_D$ as homografias retificadoras elementais (ou algum outro tipo de homografia retificadora). As homografias H_E e H_D também são homografias retificadoras, uma vez que obedecem a equação $H_D F^* H_E^{\mathrm{T}} = \lambda F^*$, onde $\lambda \neq 0$. Isso assegura que as imagens são mantidas retificadas.

As Equações (3.53) e (3.54) mostram que as estruturas internas de H_E e H_D permitem entender o significado dos parâmetros livres nas classes das homografias retificadoras.

$$H_E = \begin{bmatrix} l_1 & l_2 & l_3 \\ 0 & s & u_0 \\ 0 & q & 1 \end{bmatrix} \tag{3.53}$$

$$H_E = \begin{bmatrix} r_1 & r_2 & r_3 \\ 0 & s & u_0 \\ 0 & q & 1 \end{bmatrix} \tag{3.54}$$

Onde:

- $s \neq 0$ e é uma escala vertical comum;
- u_0 é um deslocamento vertical comum;
- l_1 e r_1 são inclinações da esquerda e da direita;
- l_2 e r_2 são escalas horizontais esquerda e direita;
- l_3 e r_3 são deslocamentos da esquerda e da direita; e
- q é a distorção perspectiva comum.

O terceiro passo é necessário, pois a homografia elementar pode conduzir a distorções severas nas imagens.

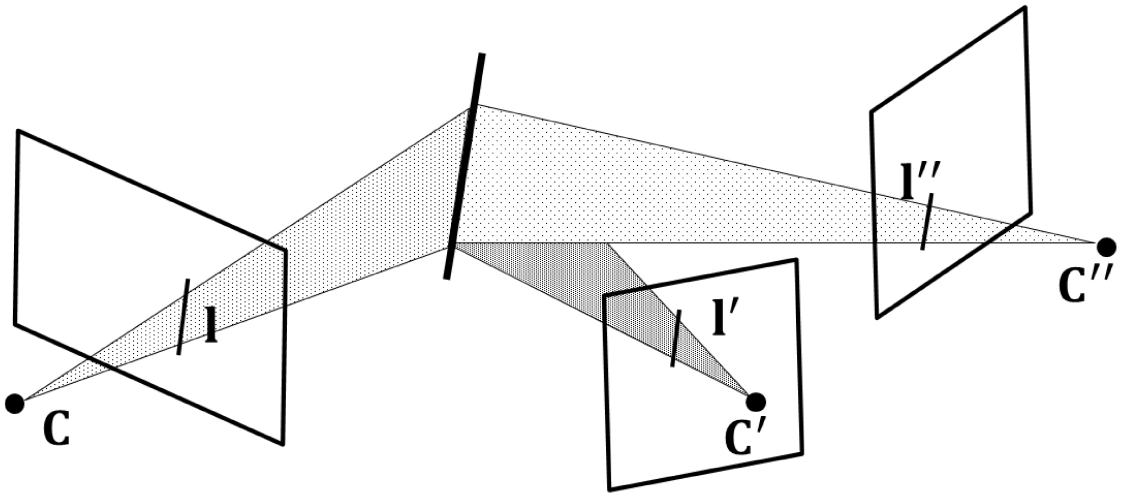
O algoritmo difere pelo modo como os parâmetros livres são selecionados. Uma aproximação que minimiza a distorção residual na imagem é apresentada em (LOOP; ZHANG, 1999; GLUCKMAN; NAYAR, 2001). Outra aproximação a ser considerado é o quanto os dados subjacentes se modificam ao utilizar a análise espectral e minimizar as perdas de informações na imagem (MATOUSEK; SARA; HLAVAC, 2004).

3.8 Utilizando Três Câmeras e o Tensor Trifocal

A subseção 3.7 combinou os pontos entre duas vistas que existem na geometria epipolar. Foi apresentado na subseção 3.4 que também é possível utilizar três e quatro vistas para encontrar a profundidade dos pontos da cena. Nesta seção é apresentado uma maneira de analisar três vistas que permita encontrar a distância dos pontos da cena até o observador, retornando, assim, a informação da profundidade desses pontos. Esta técnica consiste em encontrar um conjunto de funções trilineares capazes de descreverem as coordenadas da imagem.

A seguir é apresentado como encontrar o tensor trifocal baseado nos trabalhos de (HARTLEY; ZISSERMAN, 2003). As restrições encontras nas três vistas recebem as suas formas simples a partir das equações que calculam a linha $\mathbf{l}$ na primeira vista a partir de uma linha $\mathbf{l}'$ da segunda vista e, também, de uma linha $\mathbf{l}''$ da terceira vista. O significado geométrico dessa construção é simples: Retroprojetar as linhas $\mathbf{l}'$ e $\mathbf{l}''$ no plano da cena encontrando uma linha da cena comum a esses planos e, então, projetar essa linha na primeira vista. A Figura 3.6 ilustra um exemplo desse processo.

Figura 3.6 –Correlação das restrições ao longo das três vistas



Fonte: Autor

Na Figura 3.6 As câmeras possuem os seus centros em $\mathbf{C}$, $\mathbf{C}'$ e $\mathbf{C}''$ e, também, seus próprios planos de imagem. Uma linha em 3D é projetada como as linhas $\mathbf{l}$, $\mathbf{l}'$ e $\mathbf{l}''$

Considere as três vistas geradas por câmeras que apresentam as matrizes de câmera M, M' e M''. Devido à ambiguidade projetiva descrita na subseção 3.3, pode-se escolher $M = [I|\mathbf{0}]$ sem perda de generalidade. Assim, usando os resultados da Equação (3.15), pode-se desenvolver as Equações (3.55) a (3.57).

$$M = [I|\mathbf{0}] \tag{3.55}$$

$$M' = [\bar{M}'|\mathbf{e}'] \tag{3.56}$$

$$M'' = [\bar{M}''|\mathbf{e}''] \tag{3.57}$$

Nas Equações (3.56) e (3.57) os epipolos $\mathbf{e}'$ e $\mathbf{e}''$ são a projeção do centro da primeira câmera, $\mathbf{C} = [0,0,0,1]^{\mathrm{T}}$, na segunda e terceira câmera, respectivamente.

De forma a satisfazer essas restrições, os planos da cena, são representados pelas Equações (3.58) a (3.60).

$$\mathbf{a} = M^{\mathrm{T}}\mathbf{l} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.58}$$

$$\mathbf{a}' = M'^{\mathrm{T}}\mathbf{l}' = \begin{bmatrix} \bar{M}'^{\mathrm{T}}\mathbf{l}' \\ \mathbf{e}'^{\mathrm{T}}\mathbf{l}' \end{bmatrix} \tag{3.59}$$

$$\mathbf{a}'' = M''^{\mathrm{T}}\mathbf{l}'' = \begin{bmatrix} \bar{M}''^{\mathrm{T}}\mathbf{l}'' \\ \mathbf{e}''^{\mathrm{T}}\mathbf{l}'' \end{bmatrix} \tag{3.60}$$

Os planos representados pelas Equações (3.58) a (3.60) são retroprojetados a partir das linhas da imagem (como também indicado pela Equação (2.32)) possuem uma linha da cena em comum. Isto ocorre somente se os vetores representados pelas Equações (3.58), (3.59) e (3.60) são linearmente dependentes. Isto é, $\mathbf{a} = \lambda'\mathbf{a}' + \lambda''\mathbf{a}''$ para algum escalar λ' e λ''. Aplicando isto à quarta coordenada dos vetores nas Equações (3.58), (3.59) e (3.60) conduz a $\lambda'\mathbf{e}'\mathbf{l}' = -\lambda''\mathbf{e}''^{\mathrm{T}}\mathbf{l}''$. Substituindo nas três primeiras coordenadas dos vetores nessas equações obtém-se a Equação (3.61).

$$\mathbf{l} \simeq (\mathbf{e}''^{\mathrm{T}}\mathbf{l}'')\bar{M}'^{\mathrm{T}}\mathbf{l}' - (\mathbf{e}'^{\mathrm{T}}\mathbf{l}')M''^{\mathrm{T}}\mathbf{l}'' = (\mathbf{l}''^{\mathrm{T}}\mathbf{e}'')M'^{\mathrm{T}}\mathbf{l}' - (\mathbf{l}'^{\mathrm{T}}\mathbf{e}')\bar{M}''^{\mathrm{T}}\mathbf{l}'' \tag{3.61}$$

Reorganizando a Equação (3.61) e utilizando $\mathbf{l} \simeq [\mathbf{l}'^{\mathrm{T}}T_1\mathbf{l}'', \quad \mathbf{l}'^{\mathrm{T}}T_2\mathbf{l}'', \quad \mathbf{l}'^{\mathrm{T}}T_3\mathbf{l}'']^{\mathrm{T}}$obtém-se a equação (3.62).

$$T_i = \mathbf{m}_i'\mathbf{e}''^{\mathrm{T}} - \mathbf{m}_i''\mathbf{e}'^{\mathrm{T}} \tag{3.62}$$

Tem-se, também, $\bar{M}' = [\mathbf{m}_1' \quad \mathbf{m}_2' \quad \mathbf{m}_3']$ e $\bar{M}'' = [\mathbf{m}_1'' \quad \mathbf{m}_2'' \quad \mathbf{m}_3'']$. As três matrizes 3×3 de T_i podem ser vistas como parcelas do **Tensor Trifocal** que possui dimensão de $3 \times 3 \times 3$.

A Equação (3.62) é bilinear nas coordenadas das linhas da imagem e descreve como calcular a linha da imagem na primeira vista considerando que são conhecidas as linhas nas outras duas vistas. Na subseção 3.4 foi mostrado que existe uma única função trilinear abrangendo o ponto $\mathbf{u}$ na primeira imagem, a linha $\mathbf{l}'$ e a linha $\mathbf{l}''$ que desaparecem se existe um ponto da cena projetado nelas. Dessa forma, a partir da relação $\mathbf{l}^{\mathrm{T}}\mathbf{u} = 0$ pode-se escrever a Equação (3.63).

$$[\mathbf{l}'^{\mathrm{T}}T_1\mathbf{l}'', \quad \mathbf{l}'^{\mathrm{T}}T_2\mathbf{l}'', \quad \mathbf{l}'^{\mathrm{T}}T_3\mathbf{l}'']\mathbf{u} = 0 \tag{3.63}$$

As nove restrições de correspondência entre os três pontos $\mathbf{u}$, $\mathbf{u}'$e $\mathbf{u}''$ na primeira, segunda e terceira vista, respectivamente, podem ser obtidos ao substituir qualquer linha da matriz $S_{(\mathbf{u}')}$ por $\mathbf{l}'$ e qualquer linha de $S_{(\mathbf{u}'')}$ por $\mathbf{l}''$.

O tensor trifocal $\{T_1 \quad T_2 \quad T_3\}$ possui $3^3 = 27$ parâmetros, porém ele é definido até uma determinada escala. Isso conduz a 26 parâmetros. Entretanto, esses parâmetros satisfazem oito relações não-lineares, reduzindo o trabalho para apenas 18 parâmetros livres.

Devido às múltiplas correspondências nas três vistas, o tensor trifocal pode ser estimado ao resolver os sistemas apresentados ou na Equação (3.61) ou na (3.63). Geralmente esses sistemas são sobredeterminados e são lineares em relação às componentes do tensor. Neste ponto, o pré-condicionamento apresentado na subseção 2.3.3 é essencial.

Se o tensor trifocal é conhecido, então as matrizes de projeção correspondentes a cada câmera individual podem ser calculadas a partir desse tensor. O tensor trifocal expressa a relação entre as imagens e é independente de uma transformação de projeção 3D particular. Isto implica que as matrizes de projeção correspondentes às câmeras podem ser calculadas até uma ambiguidade projetiva.

O algoritmo para decompor o tensor trifocal em três matrizes de projeção pode ser encontrado em (HARTLEY; ZISSERMAN, 2003).

3.8.1 Algoritmos de correspondências estereoscópicas

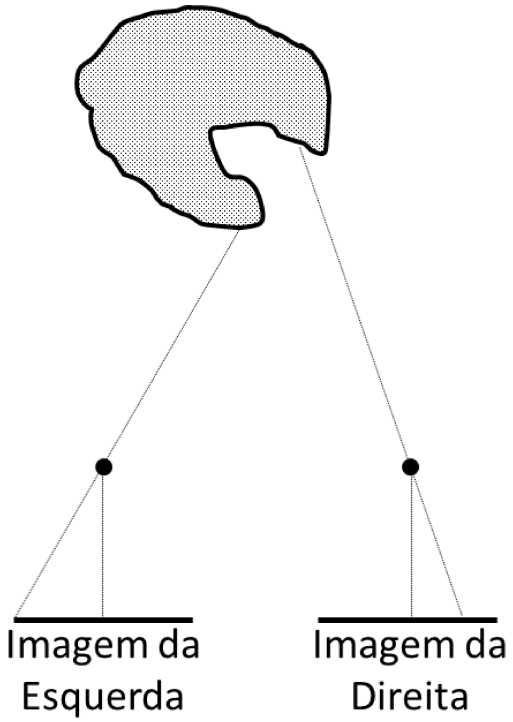
Foi apresentado na subseção 3.7.1 que é possível conhecer muito sobre a geometria 3D da cena se são conhecidos quais os pontos da primeira imagem correspondem à pontos da segunda imagem. A solução deste **problema de correspondência de pontos** é um passo importante em qualquer problema de fotogrametria, visão estereoscópica ou análise por movimentação. Este trabalho descreve como o mesmo ponto pode ser encontrado em duas imagens se a mesma cena é observada por duas vistas diferentes. Neste caso, é assumido que as duas imagens se sobrepõem e, assim, os pontos correspondentes são analisados nessa área em que ocorreu a sobreposição.

Na análise da imagem, alguns métodos são baseados na suposição de que as imagens possuem um espaço vetorial linear, como, por exemplo, autoimagens ou interpolação linear de imagens (WERNER; HERSCH; HLAVÁC, 1995; ULLMAN; BASRI, 1991). Esta suposição linear não é válida para imagens em geral (BEYMER; POGGIO, 1996), mas alguns autores negligenciam este fato. A estrutura de um espaço vetorial assume que a i-ésima componente de um vetor deve referenciar à i-ésima componente do outro vetor. Com isto, pode-se assumir que os problemas de correspondência foram resolvidos.

Métodos de solução automática do problema de correspondência de pontos é um tópico de constante pesquisa na visão computacional. Então, uma conclusão pessimista pode resultar na não existência de uma solução que possa unificar todos os casos. A causa disso, é que o problema de correspondência de pontos conduz uma ambiguidade inerente. Um exemplo disso é uma cena que contém um objeto plano, branco e sem textura. A imagem deste objeto apresenta uma grande região de brilho uniforme e, quando os pontos correspondentes buscados na imagem da esquerda e da direita de um objeto plano, não existe uma característica que poderia distingui-los. Como apresentado na Figura 3.7, outra dificuldade inevitável na busca dos pontos correspondentes é o problema da **auto oclusão** que ocorre em imagens de objetos não convexos, em que alguns pontos são visíveis pela câmera da esquerda e não são

visíveis pela câmera da direita ou são visíveis pela câmera da direita e não são visíveis pela câmera da esquerda.

Figura 3.7 – Auto oclusão impedindo a busca por pontos correspondentes



Fonte: Autor

Felizmente, a intensidade uniforme e a auto oclusão são raros ou incomuns na maioria das cenas de interesse práticos. Estabelecer correspondências entre as projeções dos mesmos pontos em diferentes vistas é baseado em encontrar características nas imagens que são similares em ambas as vistas e, então, é calculada a similaridade local.

Na maioria dos casos, a ambiguidade inerente do problema de correspondência de pontos pode ser reduzida utilizando diversas restrições. Algumas dessas restrições são devido à geometria do processo de captura das imagens; algumas propriedades fotométricas da cena; e devido a algumas propriedades dominantes dos objetos no mundo real/natural. Desta forma, diversas pesquisas têm sido desenvolvidas com o intuito de obter algoritmos para identificar correspondências estereoscópicas. A seguir é apresentado algumas aproximações de restrições que permitem, de forma concisa, encontrar os pontos correspondentes. Nem todas as restrições são utilizadas em todos os processos de correspondência de pontos. A lista a seguir apresenta as restrições utilizadas mais comumente que permitem a ter uma

visão mais aprofundada do problema de correspondência de pontos (KLETTE; KOSCHAN; SCHLÜNS, 1996).

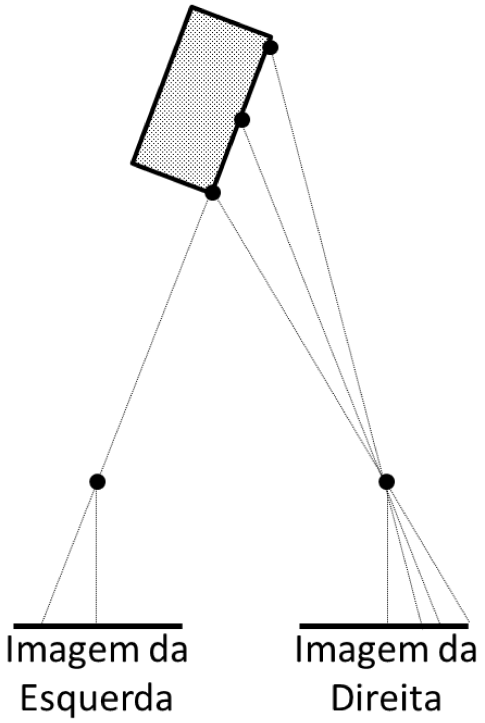
O primeiro grupo de restrições depende muito da geometria e da fotometria do processo de captura de captura da imagem.

- **Restrição epipolar:** os pontos correspondentes só podem estar localizados na linha epipolar na segunda imagem. Isso reduz um problema potencialmente 2D em um espaço 1D. A restrição epipolar foi explicada em detalhes na subseção 3.7;

Restrição de singularidade: na maioria dos casos, nessa restrição um pixel na primeira imagem pode corresponder a no máximo um pixel na segunda imagem. A exceção surge quando dois ou mais pontos estão localizados em um dos raios vindos da cena na primeira câmera e eles podem ser vistos como pontos separados na segunda imagem. Este caso surge de forma similar ao problema de auto oclusão e é mostrado na

- Figura 3.8;

Figura 3.8 – Exceção da restrição por singularidade



Fonte: Autor

- **Restrição de simetria:** se as imagens da direita e da esquerda podem ser trocadas sem modificar a imagem, então o mesmo conjunto de pares de pontos correspondentes devem ser obtidos;
- **Restrição da compatibilidade fotométrica:** nessa restrição ocorre uma diferença muito pequena de intensidade quando um ponto da cena é registrado por duas imagens. Raramente é encontrada a mesma intensidade nos pontos devido ao ângulo mútuo entre a fonte de luz, a normal da superfície e as posições de observação. Mas estes fatores conduzem a uma pequena variação de intensidade e as vistas não variam muito. Na prática, esta restrição é muito natural nas condições de captura da imagem. A vantagem é que a intensidade na imagem da esquerda pode ser transformada em intensidades da imagem da direita usando transformações muito simples; e
- **Restrição de similaridade geométrica:** as características geométricas dos objetos da cena não se alteram muito quando observados na primeira e, depois, na segunda imagem, como por exemplo os comprimentos, as orientações, as regiões e os contornos.

O segundo grupo de restrições explora algumas propriedades dos objetos típicos encontrados nas cenas.

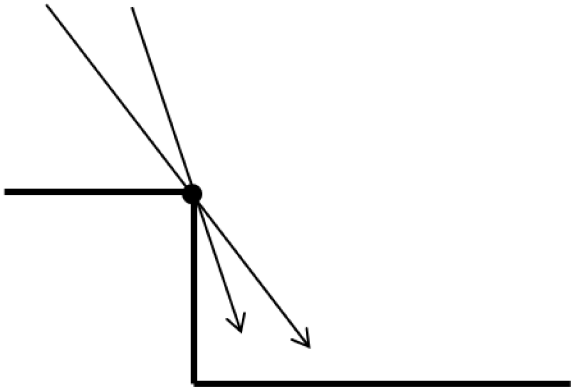
- **Restrição da diferença de rugosidades:** Esta restrição leva em consideração que as variações na imagem se modificam lentamente quase em todos os locais da imagem. Ao assumirmos dois pontos $\mathbf{p}$ e $\mathbf{q}$ que sejam próximos um do outro. Se a projeção de $\mathbf{p}$ na imagem da esquerda é $\mathbf{p}_E$ e na imagem da direita é $\mathbf{p}_D$ e para $\mathbf{q}$ temos $\mathbf{q}_E$ e $\mathbf{q}_D$. Se assumirmos que foi encontrado a correspondência entre $\mathbf{p}_E$ e $\mathbf{p}_D$, então a quantidade fornecida pela Equação (3.64), que fornece a diferença absoluta entre os pontos, dever ser pequena;

$$\left|\left|\mathbf{p}_E - \mathbf{p}_D\right| - \left|\mathbf{q}_E - \mathbf{q}_D\right|\right| \tag{3.64}$$

- **Restrição de Compatibilidade de Características:** Aqui a restrição é na origem física dos pontos coincidentes. Os pontos só podem ser coincidentes se eles apresentam a mesma origem física como, por exemplo, a descontinuidade na superfície dos objetos; o contorno de uma sombra criada por um objeto; bordas com oclusão; e bordas

especulares. É possível verificar que as arestas em uma imagem causadas por efeitos especulares ou auto oclusão não podem ser utilizadas para solucionar os problemas de correspondência, pois elas se modificam com a alteração do ponto de vista. Por outro lado, como mostra a Figura 3.9, é possível identificar auto oclusões causadas por descontinuidades abruptas da superfície.

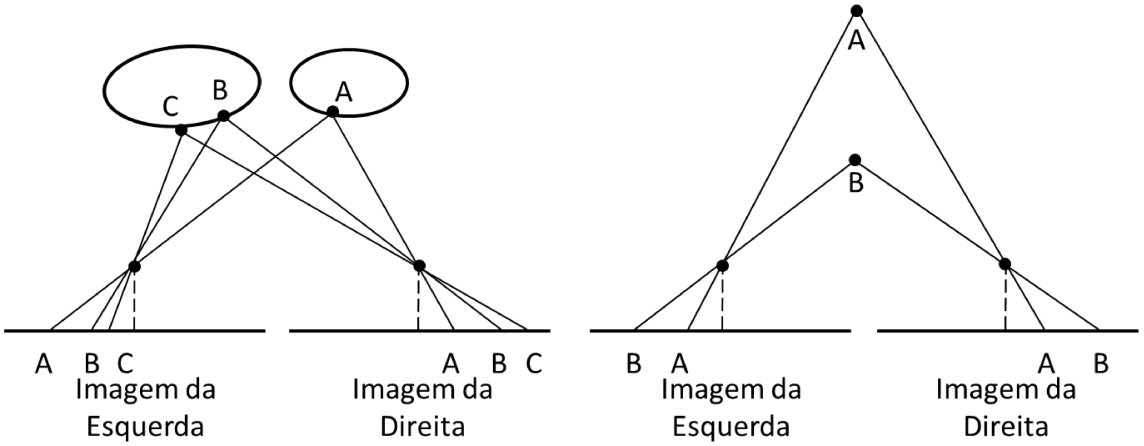
Figura 3.9 – Auto oclusão devido a uma descontinuidade abrupta da superfície



Fonte: Autor

- **Desigualdade no Intervalo de Pesquisa:** Esta restrição é colocada no comprimento da busca nos métodos artificiais que buscam as correspondências.
- **Diferença Limite do Gradiente:** Esta restrição tem origem nos experimentos psicofísicos em que ele demonstrou que o sistema de visão humano só consegue juntar imagens estereoscópicas se a diferença na alteração do gradiente dos pixels correspondentes for pequena e dentro de um limite. Esta restrição é uma versão mais fraca da Restrição da Diferença de Rugosidades.
- **Restrição de Ordenação:** Esta restrição apresenta que para superfícies de profundidade similares a correspondência dos pontos característicos normalmente são encontrados na linha epipolar na mesma ordem, como é possível verificar na Figura 3.10A. Se houver um objeto fino muito mais próximo à câmera do que o fundo da cena, a ordem que que eles aparecem na imagem da esquerda pode ser diferente da ordem em que eles aparecem na da direita, como apresentado na Figura 3.10B. A restrição de ordenação é violada raramente na prática.

Figura 3.10 – Demonstração da restrição de ordenação



(A) Pontos correspondentes são encontrados na mesma ordem na linha epipolar; (B) Mudança de posição dos objetos na linha epipolar devido a existência de um objeto fino muito próximo à câmera.
Fonte: Autor

Todas estas restrições foram utilizadas em um ou mais algoritmos existentes de correspondência estereoscópica. A seguir será apresentado um axioma destes algoritmos. Do ponto de vista histórico os algoritmos de correspondência estereoscópica foram e ainda são divididos em dois grandes paradigmas:

1. De baixo nível; baseado em correlações; e metodologia bottom-up;
2. De alto nível; baseado em características; e metodologia top-down.

Inicialmente acreditava-se que características de alto nível, tais como cantos e seguimentos de linhas retas, deveriam ser identificados automaticamente e, então, relacionados. Este era um desenvolvimento natural da fotogrametria, em eram utilizados pontos de características que eram identificados por um operador humano desde o início do Século XX.

Experimentos psicológicos com **Estereogramas de Pontos Aleatórios** realizado por Julesz (1960) levou a uma nova visão do processo. O experimento de Julesz demonstrou que os humanos não precisam criar característica monoculares antes que ocorra a percepção de profundidade binocular. Um estereograma de pontos aleatórios é criado da seguinte forma: a imagem da esquerda é completamente aleatória e a imagem da direita é criada, a partir da imagem da esquerda, de uma forma consistente em que parte dela é deslocada de acordo com uma diferença capaz de criar o efeito estereoscópico desejado. O observador deve observar o estereograma de pontos aleatórios a uma distância de aproximadamente 20 centímetros. Tal “Estereograma de Pontos Aleatórios” foi amplamente divulgado sobre o nome de “Imagens 3D” em muitas revistas populares.

Desenvolvimento mais recente nesta área utilizam a combinação dos métodos de correspondência de baixo nível e de alto nível (TANAKA; KAK, 1990).

3.8.1.1 Correlação baseada em combinação em blocos

As correlações baseadas em algoritmos de correspondência usam a suposição que os pixels na correspondência possuem intensidades semelhante (relembrando da restrição da compatibilidade fotométrica). A intensidade de um pixel individual não fornece informações suficientes. Como existem muitos candidatos em potencial com intensidades similares e, assim, as intensidades de diversos pixels vizinhos devem ser consideradas. Normalmente uma janela 5×5 ou 7×7 ou 3×9 deve ser utilizada. Estes métodos são conhecidos como **Estereoscopia baseada em Área**. Quanto maior for a janela de busca, maior é a discriminação.

Pode-se ilustrar esta aproximação com um algoritmo simples chamado de **Combinação de blocos** (KLETTE; KOSCHAN; SCHLÜNS, 1996). Assumindo a configuração estereoscópica canônica com os eixos ópticos das duas câmeras paralelos, a ideia básica do algoritmo é que todos os pixels na janela (chamada de bloco) possuam todos a mesma disparidade. Isto significa que uma, e apenas uma, disparidade é calculada para cada bloco. Uma das imagens, digamos a da esquerda, é dividida em blocos e uma busca por correspondência, na imagem da direita, é conduzida para cada um destes blocos na imagem da direita. A medida de similaridade entre os blocos pode ser, por exemplo, o valor RMS do erro de intensidade e a disparidade é aceita para a posição onde o valor RMS do erro é o menor. A modificação máxima da posição é limitada pela restrição do limite de disparidade. A busca do menor valor RMS do erro pode apresentar mais de um mínimo e, neste caso, uma restrição adicional é utilizada para ajudar a resolver o problema da ambiguidade.

O resultado não obedece às restrições de simetria, de ordenação e de limite do gradiente, pois o resultado não é um casamento direto de um-para-um.

Outra aproximação relevante é a de Nishihara (1984). Ele observou que na tentativa de um algoritmo fazer a correlação de pixels individuais (com por exemplo o casamento por cruzamento por zero (MARR; POGGIO, 1979)) ele está destinado a ter um péssimo desempenho, pois os ruídos nas imagens irão tornar a tarefa de localização das características impossível. Uma observação secundária é de que tais

mecanismos de correlação necessitam de um processamento muito pesado e demorado para se obter as correspondências. Nishihara verificou que os sinais (e as magnitudes) da resposta de um detector de bordas é uma propriedade muito mais estável para o processo de busca do que a borda em si ou a localização da característica. Tendo isto em vista, Nishihara deixou a disposição um algoritmo que explora, simultaneamente, a resposta de um detector de bordas e um ataque de correspondência por espaço e escala.

A aproximação é correlacionar grande manchas em uma grande escala e, então, refinar a qualidade da correlação ao reduzir a escala, utilizando as informações grosseiras para inicializar a correlação da escala mais fina. Um a resposta de borda é gerada em cada pixel de ambas as imagens na escala maior e, então, uma grande área da imagem da esquerda (representada pelo, por exemplo, seu pixel central) é correlacionada com uma grande área da imagem da direita. Isto pode ser feito de forma rápida e eficiente utilizando-se do fato que a função correlação apresenta picos muito agudos nos pontos de casamento e, então, apenas um pequeno número de teste é suficiente para encontrar o máximo de uma correlação. Este casamento das áreas grosseiras pode, então, ser refinado até chegar a uma resolução desejada, de várias formas interativas, utilizando-se do conhecimento obtido da escala grosseira como dica para a disparidade correta em uma determinada posição. Portanto, a qualquer momento do algoritmo, as superfícies nas vistas são modeladas como prismas retangulares de alturas diferentes; as áreas da superfície destes prismas retangulares podem ser reduzidas através de um algoritmo de refinamento de escala. Para tarefas tais como evitar obstáculos é possível que apenas as informações da escala mais grosseira sejam suficientes e, assim, o algoritmo terá um ganho na eficiência.

Qualquer algoritmo de correlação estereoscópica pode ser potencializado ao utilizar padrões de pontos de luzes aleatoriamente na cena, para fornecer padrões de correlação mesmo em áreas da cena que possuam texturas uniforme. O sistema resultante tem sido demonstrado em sistemas de movimentação de robôs e em manipuladores de objetos, sendo estas técnicas implementadas de forma robusta em tempo real.

3.8.1.2 Correspondentes estereográficas baseada em características

Métodos de correspondência baseados em características normalmente salientam pontos ou conjuntos de pontos que se destacam e, também, são fáceis de serem encontrados. Caracteristicamente, estes são pixels em de bordas, linhas, cantos, etc., e as correspondências são procurados de acordo com propriedades de tais características como, por exemplo, orientação ao longo das bordas, ou o comprimento de seguimentos de linhas. As vantagens dos métodos baseados em características sobre as correlações baseadas em intensidade são:

- Métodos baseados em características são menos ambíguos, pois o número de candidatos em potencial para as correspondências é menor;
- A correspondência resultante é menos dependente da variação fotométrica das imagens;
- As disparidades podem ser calculadas com elevada precisão, as características podem ser buscadas na imagem a precisão subpixel.

Será apresentado um exemplo de um método de correspondência baseado em características – o Algoritmo PMF (subseção 3.8.1.3), nomeado com as iniciais de seus inventores (POLLARD; MAYHEW; FRISBY, 1985). Ele trabalha assumindo que um conjunto de pontos característicos (como um detector de borda, por exemplo) foi identificado na imagem por um operador. O resultado do algoritmo é uma correspondência entre os pares de tais pontos. De forma a realizar esta tarefa três restrições são aplicadas: a restrição epipolar; a restrição de singularidade; e diferença limite do gradiente.

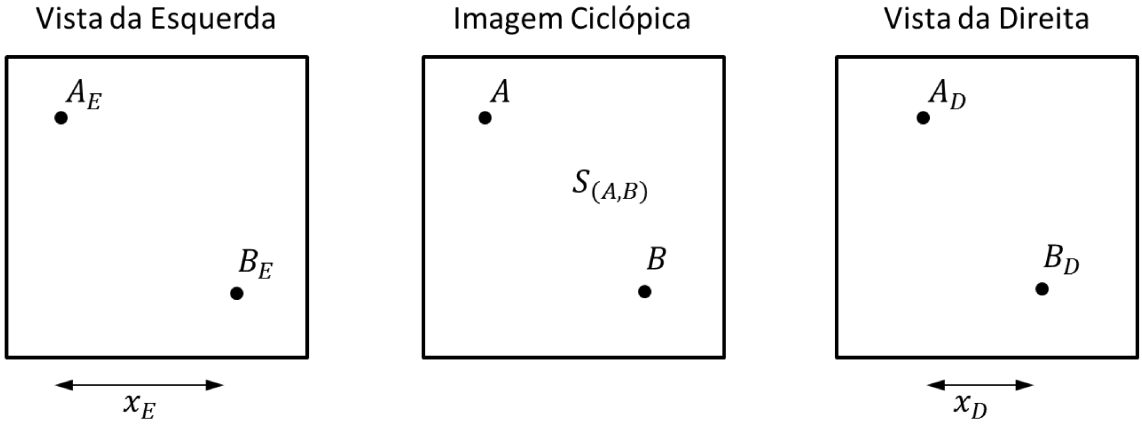
As duas primeiras restrições não são peculiares a este algoritmo, elas também foram utilizadas por Marr e Poggio (1979), por exemplo, mas, entretanto, a terceira restrição que estipula um limite para a variação do gradiente é a novidade deste algoritmo. A **Diferença de Gradiente** mede a divergência de um par de pontos relacionados.

Suponhamos que um ponto A ou B em 3D apareça com $A_E = (a_{xE}, a_y)$ ou $B_E = (b_{xE}, b_y)$ na imagem da esquerda e $A_D = (a_{xD}, a_y)$ ou $B_D = (b_{xD}, b_y)$ na imagem da direita, como mostrados na Figura 3.11. Foi utilizado a restrição epipolar, assim, temos que as coordenadas y devem ser iguais. A imagem **Ciclópica** é definida como aquela dada pela média das coordenadas, ou seja:

$$A_C = \left(\frac{a_{xE}+a_{xD}}{2}, a_y\right) \tag{3.65}$$

$$B_C = \left(\frac{b_{xE}+b_{xD}}{2}, b_y\right) \tag{3.66}$$

Figura 3.11 – Definição do gradiente de disparidade



Fonte: Autor

e a **Separação Ciclópica** S delas é dada pela distância que separa os pontos nestas imagens

$$\begin{aligned} S_{(A,B)} &= \sqrt{\left[\left(\frac{a_{xE}+a_{xD}}{2}\right) - \left(\frac{b_{xE}+b_{xD}}{2}\right)\right]^2 + \left(a_y - b_y\right)^2} \\ &= \sqrt{\frac{1}{4}[(a_{xE} - b_{xE}) + (a_{xD} - b_{xD})]^2 + \left(a_y - b_y\right)^2} \\ &= \sqrt{\frac{1}{4}(x_E + x_D)^2 + \left(a_y - b_y\right)^2} \end{aligned} \tag{3.67}$$

a dispersão entre as correspondência de A com B é dada por

$$\begin{aligned} D_{(A,B)} &= (a_{xE} - a_{xD}) - (b_{xE} - b_{xD}) \\ &= (a_{xE} - b_{xE}) - (a_{xD} - b_{xD}) \\ &= x_E - x_D \end{aligned} \tag{3.68}$$

o gradiente da diferença dos pares correspondentes é dado pela razão da dispersão da separação ciclópica:

$$\Gamma_{(A,B)} = \frac{D_{(A,B)}}{S_{(A,B)}} = \frac{x_E - x_D}{\sqrt{\frac{1}{4}(x_E + x_D)^2 + \left(a_y - b_y\right)^2}} \tag{3.69}$$

Dada estas definições, a restrição a ser explorada é que, na prática, o gradiente da diferença Γ pode ser considerado como sendo limitado. Em testes realizados ele dificilmente excede a 1 (um). Isso significa que dispersões muito pequenas não são aceitáveis se os pontos correspondentes são extremamente próximos uns dos outros na cena 3D. Isto se apresenta como uma observação

intuitivamente razoável e, também, é apoiada em uma grande gama de evidências físicas (POLLARD; MAYHEW; FRISBY, 1985). Uma solução para o problema de correspondência é, então, extraída pelo processo de relaxação, no qual todas as correspondências possíveis são pontuadas de acordo com que se eles são suportados por outras correspondências possíveis que não violam a diferença limite do gradiente estipulada. Correlações com pontuação elevada são consideradas como corretas, o que permite uma evidência firme para extrair correspondências subsequentes.

3.8.1.3 Algoritmo de correspondência estereográfica PMF

Ao utilizar correspondências estereográficas deve-se seguir os passos:

1. Extrair as características para correlacionar as imagens da esquerda e da direita. Estas características podem ser, por exemplo, pixels de contornos;
2. Para cada característica na imagem da esquerda deve-se considerar uma correlação possível na imagem da direita. Isto pode ser definido de forma apropriada pela linha epipolar;
3. Para cada uma das correlações realizadas deve-se incrementar a sua probabilidade, pontuando-as de acordo com o número de outras correlações possíveis encontradas que não violem a diferença limite do gradiente estipulada;
4. Qualquer correlação que possuam a maior pontuação para os pixels que as compõem são agora tratadas como corretas. Devido à restrição de singularidade, estes pixels devem ser retirados de todas as outras considerações.
5. Retornar ao passo 2 e recalcular as pontuações, levando em consideração a correlação encontrada;
6. Termina quando todos os pontos correlacionados forem encontrados.

Observa-se que aqui a restrição epipolar é utilizada no passo 2 para limitar em uma única dimensão as possibilidades de correlacionar os pixels e, também, a restrição de singularidade utilizada no passo 4 para garantir que um pixel em particular não seja utilizado mais que uma vez nos cálculos de um gradiente.

O mecanismo de pontuação deve levar em consideração o fato que, no mais remoto que seja, possam existir duas correlações possíveis. Caso isto aconteça as

correlações devem, também, satisfazer a restrição da diferença limite do gradiente. Estes requisitos são atendidos da seguinte forma:

- Considerar apenas correlações que estão “próximas” a aquela a ser pontuada. Na prática é tipicamente adequado considerar somente aquelas dentro de um círculo de raio igual à 7 pixels e centrada nos pixels da correlação (embora o tamanho do raio depende da precisão da geometria em que o algoritmo está lidando, tamanho do objeto);
- Ponderando o valor da pontuação levando em consideração a sua distância a partir da correlação que está sendo avaliada. Assim, os pares mais distantes, que são mais prováveis de satisfazerem o limite por acaso devem possuir uma pontuação menor.

O algoritmo PMF tem demonstrado que funciona relativamente bem. Ele também é atrativo, pois ele se comporta bem com implementações paralelas e pode ser extremamente rápido quando utilizado em um hardware devidamente escolhido. Ele apresenta a desvantagem (assim como vários outros algoritmos) de que os seguimentos de linhas horizontais são difíceis de serem correlacionados. Muitas vezes estes seguimentos de linhas se confundem nas varreduras adjacentes e, devido a geometria paralela das câmeras, qualquer ponto em uma linha pode ser correlacionado com qualquer ponto da linha correspondente na outra imagem.

Desde a criação do algoritmo PMF diversos outros algoritmos de várias complexidades foram propostos. Dois algoritmos eficientes e fáceis de serem implementados utilizam-se tanto de técnicas de otimização, denominada de programação dinâmica (GIMEL’FARB, 1999), ou da correlação estável de confiança (ŠÁRA, 2002). São mantidos em http://vision.middlebury.edu/stereo/eval/ (site visitado em 12/12/2014) uma grande lista de vários algoritmos para correlação estereoscópica.

3.9 Considerações Finais

Este capítulo apresentou a técnica utilizada para analisar as imagens e como desenvolver o modelo em 3D utilizando representações 2D de cenas.

O próximo capítulo mostra a bancada experimental construída nesse trabalho.

Capítulo 4

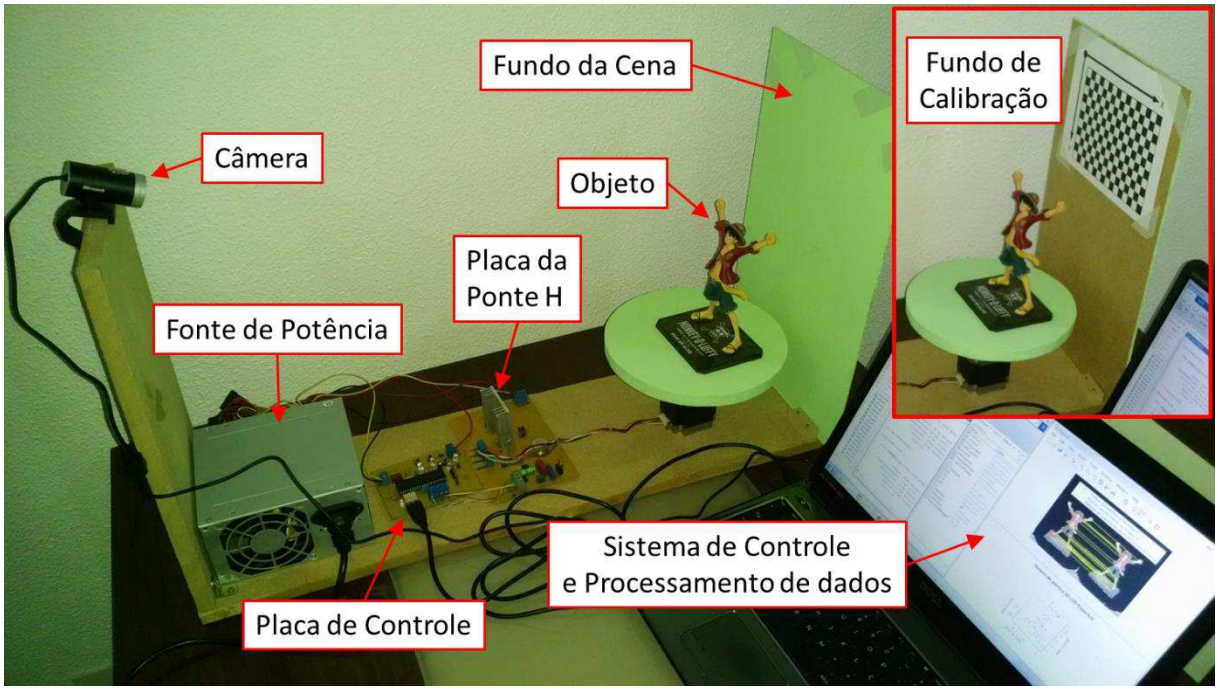
4 BANCADA DE LABORATÓRIO

Neste capítulo será mostrado a bancada de experimentos construída. O objetivo desta bancada é permitir desenvolver o modelo em 3D utilizando representações 2D de objetos, tendo como conhecimento *a priori* as posições de câmera.

4.1 Introdução

No Capítulo 2 foi apresentado a teoria geométrica por trás das projeções que ocorrem ao converter o mundo 3D em uma imagem 2D pela câmera. Na tentativa de implementar esses cálculos geométricos foi elaborado uma bancada experimento que consiste em uma mesa rotatória, uma câmera fixa e um fundo elaborado de forma a permitir a obtenção dos parâmetros internos da câmera. A Figura 4.1 apresenta uma visão geral da bancada de experimento.

Figura 4.1 – Visão geral da bancada de Laboratório

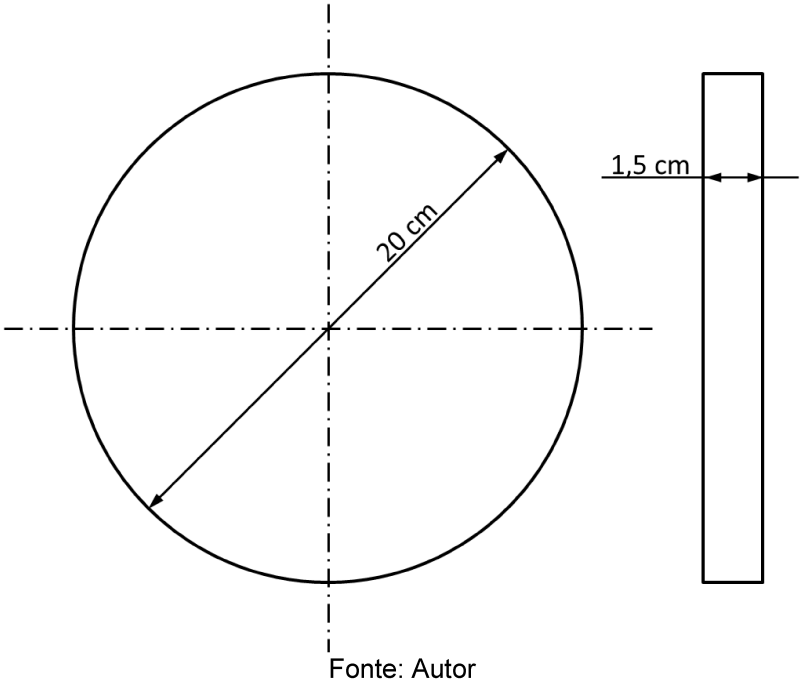


Fonte: Autor

4.2 Mesa Giratória

A mesa giratória consiste de um apoio circular feito de compensado com 20 cm de diâmetro e 1,5 cm de espessura, conforme apresentado na Figura 4.2.

Figura 4.2 – Dimensões do apoio circular



Fonte: Autor

Para girar o apoio circular é utilizado o motor de passo NEMA 23, apresentado na Figura 4.3. A vantagem de utilizar o motor de passo NEMA 23 é que possível controlar, com grande exatidão, o ângulo de giro do apoio circular e, dessa forma, ter um bom controle da posição do objeto que será digitalizado, pois o ângulo de passo desse motor é de 1,8°/passo. Utilizando técnicas e um sistema de controle simples é possível melhorar a resolução desse motor, dividindo o passo do motor ao meio por cada pulso de controle, dessa forma temos uma resolução de 0,9°/pulso.

Figura 4.3 – Motor de passo NEMA 23 utilizado na Bancada de Laboratório

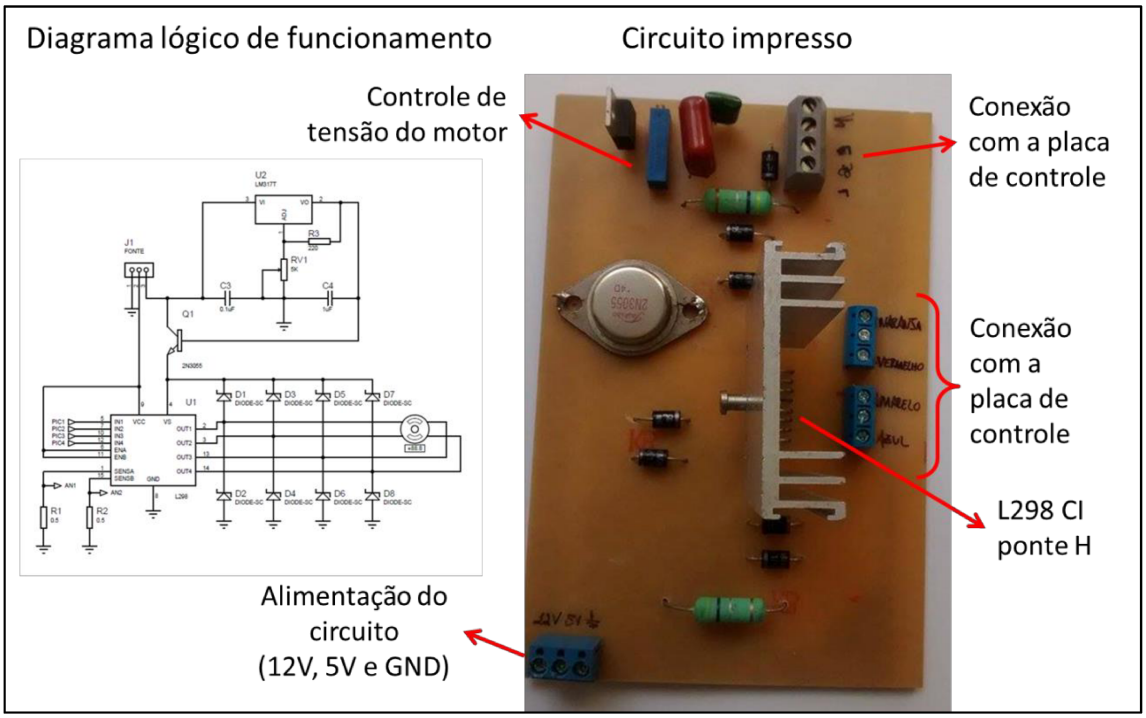
Fonte: Autor

Essa grande resolução será importante para avaliar quanto de sobreposição as imagens devem ter para que seja possível restaurar a profundidade dos pontos da cena.

Para fazer o controle do motor de passo e obter a resolução de 0,9°/pulso, foi utilizado uma ponte H (CI L298) controlada pelo microprocessador PIC18F4550 (fazendo a lógica do acionamento da ponte H) de forma a controlar a corrente nos enrolamentos do motor e, assim, conseguir que o movimento do motor seja de meio passo. A Figura 4.4 apresenta o diagrama lógico da conexão do motor com a ponte H, além de alguns acessórios para o controle da tensão e corrente no motor.

Para controlar o motor de passo foi criada uma placa contendo um micro controlador PIC18F4550, denominada de placa de controle. A função principal da placa de controle é permitir conectar um PC ao motor. A maioria dos microcontroladores possuem um hardware que interpreta o USART (Universal Synchronous Asynchronous Receiver Transmitter) que permite comunicação de forma serial com um PC utilizando um simples conversor lógico como o CI MAX232. Entretanto, os atuais PC não possuem mais a porta de comunicação serial. A saída encontrada para conectar o PC ao motor foi a utilização da porta USB (Universal Serial Bus), presente nos atuais PC. Para isso teve-se que recorrer a um micro controlador que aceitasse a comunicação nos padrões da USB, dessa forma o PIC18F4550 foi escolhido.

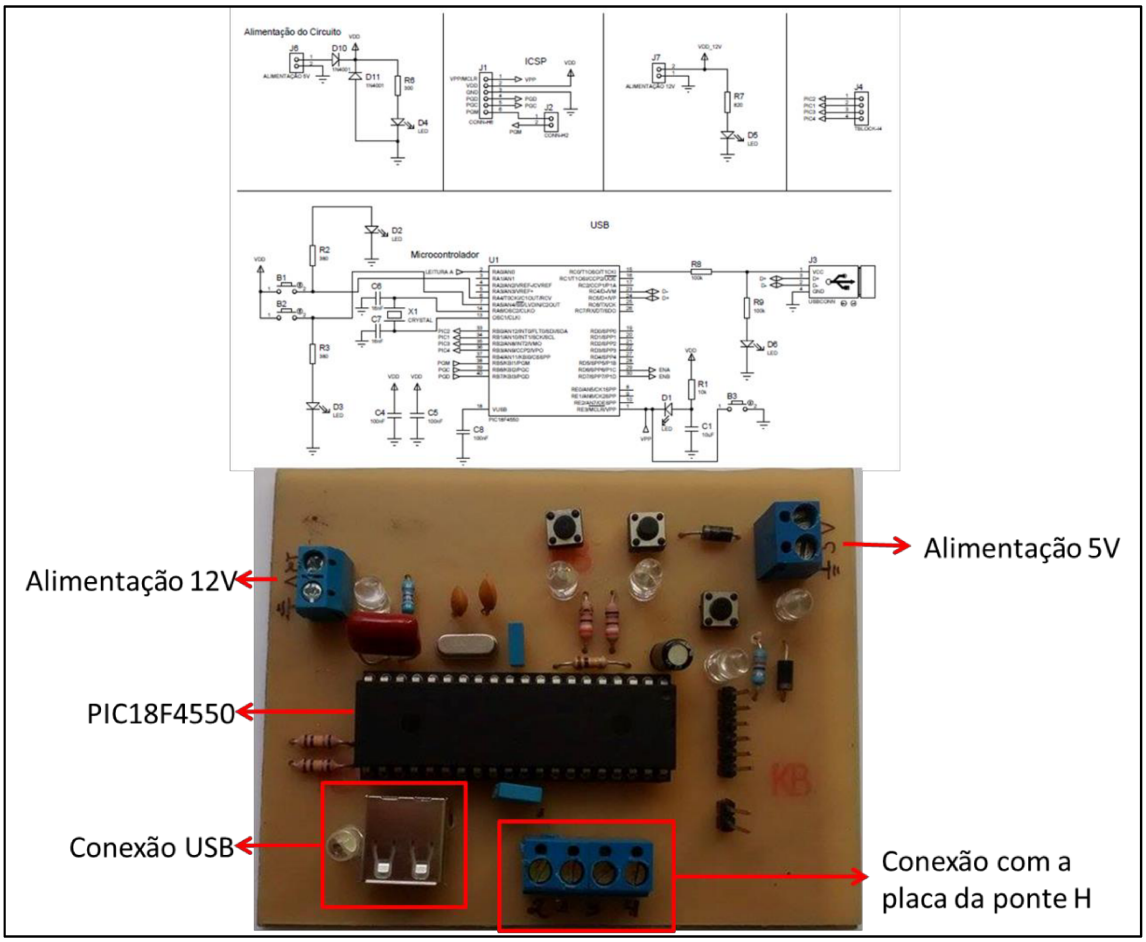
Figura 4.4 – Circuito lógico da ligação da ponte H com o motor de passo



Fonte: Autor

Além de permitir a conexão com o PC, o PIC18F4550 é o responsável pela lógica de controle do motor de passo, fazendo que o motor avance meio passo a cada solicitação do PC e, retirando do PC, responsabilidade sobre o controle e acionamento da ponte H. Dessa forma o PC fica apenas com a escolha do sentido de giro e do número de passos. Assim, para que o motor gire o PC precisa apenas de enviar à placa de controle duas mensagens simples: uma para que o motor avance um passo no sentido horário e a outra para que o motor avance um passo no sentido anti-horário. A quantidade de movimentos que o motor deverá fazer em um sentido é controlada pelo PC repetindo a mensagem. O esquema elétrico da placa de controle é apresentado na Figura 4.5.

Figura 4.5 – Esquema elétrico da placa de controle



Fonte: Autor

Na tentativa de simplificar o projeto elétrico as placas (da ponte H e a do controle) foram alimentadas por uma fonte de PC convencional. A escolha foi feita por ela já fornecer os 5V para alimentação da parte lógica dos circuitos e 12V para o motor com potência suficiente para todo o processo.

4.2.1 Software do micro controlador da placa de controle

A princípio, tentou-se desenvolver um software para o micro controlador de forma que, ao conectar a placa ao PC, a placa de controle seria reconhecida como um dispositivo HID (Human Interface Device). Os dispositivos HID são aqueles reconhecidos pelos sistemas operacionais como ferramentas de interação com o PC, como mouses, teclados, joysticks, etc. A grande vantagem que o sistema teria caso ele pudesse ser reconhecido pelo PC como um dispositivo HID se deve ao fato que o sistema operacional utilizaria um driver genérico para se comunicar com o sistema de

controle e, dessa forma, não seria necessário instalar algum driver quando fosse utilizar o equipamento.

Para o processamento das imagens optou-se em utilizar o MATLAB®, pois esse software já teria incluso nele os mecanismos de captura da imagem por webcam e um toolbox de processamento de imagens, permitindo focar os trabalhos apenas nos algoritmos de processamento. Entretanto, o MATLAB® não oferece suporte a dispositivos USB (MATHWORKS SUPPORT TEAM, 2012). Na tentativa de contornar esse problema o código criado para o PIC18F4550 utiliza o USB CDC (USB communications Device Class). Ao utilizar o USB CDC o sistema operacional reconhece o dispositivo USB como sendo uma porta serial e o MATLAB® é capaz de utilizar portas seriais do PC. A desvantagem encontrada nesse caso é a necessidade de instalação de um driver para que o sistema operacional identifique o USB CDC.

Com todos esses mecanismos montados como apresentado foi possível desenvolver um software em MATLAB quer permite escolher quanto o motor deve girar entre as fotos e, também, adquirir as imagens de forma automática. O que permite avaliar o índice de sobreposição das imagens.

4.3 Câmera

Para adquirir as imagens a bancada de laboratório possui uma Microsoft® LifeCam Cinema. Essa câmera foi escolhida devido a sua alta resolução, de até 5 megapixel (2880X1620 pixels). Dessa forma a câmera consegue capturar muitos detalhes da cena. Os detalhes são importantes na hora de detectar pontos característicos da cena. Outras características da câmera são sistema de estabilização da imagem e zoom automático. A câmera é apresentada na Figura 4.6

Figura 4.6 – Câmera utilizada na bancada de teste

Fonte: Site do Fabricante (Microsoft®)

Entretanto o fabricante não oferece os parâmetros de calibração da câmera. Como foi demonstrado no Capítulo 2, a transformação projetiva da câmera pode ser definida como:

$$\mathrm{u} = [\mathrm{P}] \cdot \mathrm{X} \tag{4.1}$$

Em que u é a coordenada do ponto na imagem em coordenadas homogênea ($\mathrm{u} = [\mathrm{u}, \mathrm{v}, 1]^T$), X é a coordenada do ponto na cena ($\mathrm{X} = [\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1]^T$) e a matriz que mapeia X em u é a matriz $[\mathrm{P}]$ denominada matriz da câmera.

A matriz da câmera pode ser dividida em duas em matrizes que traduzem as características da câmera. Essas características são classificadas como intrínsecas e extrínsecas. Dessa forma temos $[\mathrm{P}]$ como:

$$[\mathrm{P}] = [\mathrm{H}] \cdot [\mathrm{E}] \tag{4.2}$$

A matriz $[\mathrm{E}]$ carrega as características extrínsecas da câmera tais como a rotação da câmera no espaço e a distância do centro de coordenadas. Já a matriz $[\mathrm{H}]$ carrega as características intrínsecas da câmera, em outras palavras, as características internas dela como a distância focal e as distorções causadas pela curvatura da lente. Como o fabricante não informa a calibração da câmera (matriz $[\mathrm{H}]$) e a câmera possui foco automático, torna-se necessário criar um mecanismo que permita o cálculo da matriz $[\mathrm{H}]$, evitando erros n a matriz $[\mathrm{P}]$ e, consequentemente problemas na restauração da coordenada Z em X.

4.4 Tratamento da Cor do Fundo da Cena

Devido às características do fundo como sombras e texturas podem levar o algoritmo a detectar pontos que não sejam do objeto em estudo. Ao girar o objeto com a mesa giratória esses falsos pontos podem ser encontrados em mais de uma fotografia, levando a erros de posicionamento da projeção da câmera e dos pontos no espaço.

Para resolver esse problema, tanto a base da mesa giratória quanto o fundo foram pintados na cor verde limão, de forma a obter uma estrutura lisa e de cor uniforme, evitando, assim, os possíveis falsos erros. Essa técnica de destacar o fundo com uma determinada cor é chamada de “Chroma Key”.

A importância em utilizar a técnica de Chroma Key é que ela facilita a segmentação da imagem, removendo uma determinada cor da imagem para revelar outra imagem escondida (VIDAL, 2012), que neste caso é nenhuma. O verde limão é comumente utilizado em sistemas de Chroma Key por possibilitar a utilização de menos luz por ser claro e refletir luzes fracas, reduzindo as sombras que podem interferir na segmentação da imagem (PRODUCCINE, 2013).

Dessa forma, através da técnica de Chroma Key é possível fazer a segmentação da imagem, criando uma máscara e, então, remover o fundo da imagem e, também, vários erros de análise por ruídos causados pelo fundo da cena. Com o resultado do Chroma Key obtém-se apenas a imagem do objeto. A Figura 4.7 apresenta o resultado do Chroma Key utilizado na bancada de laboratório. Ainda na Figura 4.7 podemos verificar que na imagem da direita tem-se dificuldade em separar o fundo do objeto, pois algumas partes dele se assemelham com a cor do fundo, enquanto que com o fundo verde não temos tanta semelhança do fundo com o objeto.

Figura 4.7 – Efeito do Chroma Key na bancada de laboratório

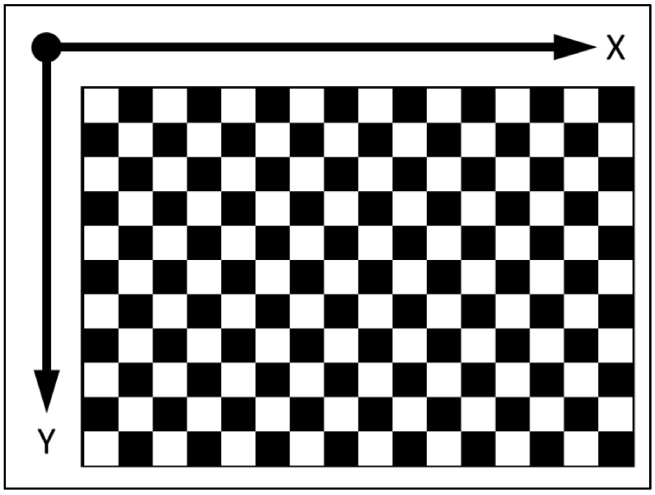
Fonte: Autor

4.5 Tratamento do Fundo da Cena para Calibrar a Câmera

Como foi apresentado na subseção 4.3 desse capítulo a matriz de projeção pode ser expressa pela Equação (4.2). O parâmetro $[E]$ pode ser obtido conhecendo-se o movimento e a posição da câmera (neste caso o giro do motor da mesa). Mas o parâmetro $[H]$ é interno à câmera e o fabricante não fornece esses valores. Outro problema encontrado é a posição do plano focal, já que a câmera utilizada possui sistema de foco automático, que também não é fornecido ao capturar a imagem.

Com o intuito de recuperar o parâmetro $[H]$ foi adicionado ao fundo da cena um padrão no formato de um tabuleiro de xadrez, como mostrado na Figura 4.8. Nesse tabuleiro, cada quadrado possui lados medindo 1 cm. Conhecendo as dimensões dos pontos desse tabuleiro é possível recuperar o parâmetro $[H]$ da câmera em cada fotografia.

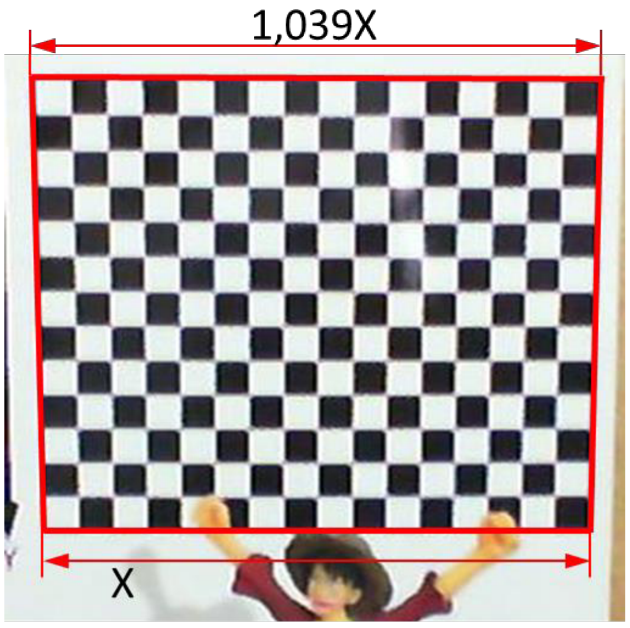
Figura 4.8 – Tabuleiro utilizado para recuperar o parâmetro [H] da câmera



Fonte: Autor

Na Figura 4.9 é apresenta a distorção do tabuleiro criada pela câmera. Ainda na Figura 4.9 é possível verificar uma variação que a linha superior do tabuleiro possui 1,039 vezes a quantidade de pixels da linha inferior do tabuleiro.

Figura 4.9 – Distorção nas dimensões do tabuleiro devido às propriedades intrínsecas da câmera



Fonte: Autor

Dessa forma, com o auxílio do tabuleiro sobre a imagem do objeto é possível estimar os parâmetros intrínsecos da câmera (os parâmetros que causa as distorções na imagem), pois é conhecida a geometria do tabuleiro e o seu formato possibilita identifica-lo facilmente na cena.

4.6 Considerações Finais

Este capítulo apresentou a bancada experimental construída neste trabalho.

O próximo capítulo apresenta como utilizar as metodologias matemáticas apresentadas nos Capítulos 2 e 3 deste trabalho utilizando a bancada experimental apresentada neste capítulo.

Capítulo 5

5 PROCEDIMENTO EXPERIMENTAL E RESULTADOS OBTIDOS

Este capítulo apresenta como utilizar as metodologias matemáticas já apresentadas e a bancada experimental para a geração de modelos em 3D de representações 2D de objetos.

5.1 Introdução

No Capítulo 2 foi apresentada a parte teórica para o desenvolvimento do trabalho. O Capítulo 4 descreve o funcionamento da bancada de laboratório. Neste capítulo é descrito como utilizar as metodologias matemáticas apresentadas no Capítulo 2, na bancada de laboratório apresentada no Capítulo 4 e, além disso, são apresentados os resultados obtidos. Para isso, este capítulo tem o objetivo de:

- Apresentar os conceitos e o método de segmentação utilizado para diferenciar o objeto do fundo da cena;
- Explicar o procedimento utilizado para a calibração da câmera, utilizando elementos da cena;
- Descrever a cinemática do movimento da câmera em relação ao objeto;
- Apresentar como obter os pontos característicos da cena/objeto;
- Demonstrar como relacionar os pontos característicos, já determinados, em diferentes cenas;
- E explicar como utilizar os pontos característicos, a calibração da cena e a cinemática da câmera para um modelo 3D do objeto da cena.

5.2 Segmentação da Imagem

A segmentação da imagem é o processo de dividir a imagem em múltiplas partes. Nesse trabalho a segmentação foi utilizada para remover o fundo da cena e o apoio circular, permitindo uma definição clara do objeto a ser reconstruído em 3D.

Existem diversas formas de realizar a segmentação da imagem. Aqui serão avaliadas as seguintes técnicas:

- Método de Otsu;
- Baseada em cor utilizando o agrupamento K-means
- Transformada watershed
- Detecção de borda e morfologia
- Por limiar no sistema de cores cieLAB

A segmentação é importante para os testes na bancada de laboratório, pois ao girar o objeto, o fundo da cena não se modifica. Isso poderá interferir na etapa seguinte de extração dos pontos característicos, pois poderá identificar pontos fora do objeto e pertencente ao fundo da cena. Caso isso ocorra, o algoritmo de triangulação irá utilizar esses pontos e, com isso, o resultado será, na melhor das hipóteses, um cilindro com o objeto de interesse no seu interior.

5.2.1 Segmentação pelo método de Otsu

O método de Otsu é um método de segmentação não paramétrico e não supervisionado para a seleção automática dos limiares dos níveis de cinza para a segmentação de imagens. Para isso o método de Otsu utiliza apenas os momentos acumulativos de ordem zero e de primeira ordem do histograma da escala de cinza, dessa forma o algoritmo consegue uma seleção ótima de limiar através de um critério de discriminação para maximizar a separação das classes resultantes em escala de cinza (OTSU, 1979). A Figura 5.1 apresenta o resultado de segmentação pelo método de Otsu. Ainda na Figura 5.1 é possível verificar que o resultado não foi satisfatório, pois partes do boneco foram confundidas com o fundo da cena.

Figura 5.1 – Resultado da segmentação pelo método de Otsu

A imagem da esquerda apresenta a imagem original e a da direita a máscara de remoção do fundo
Fonte: Autor

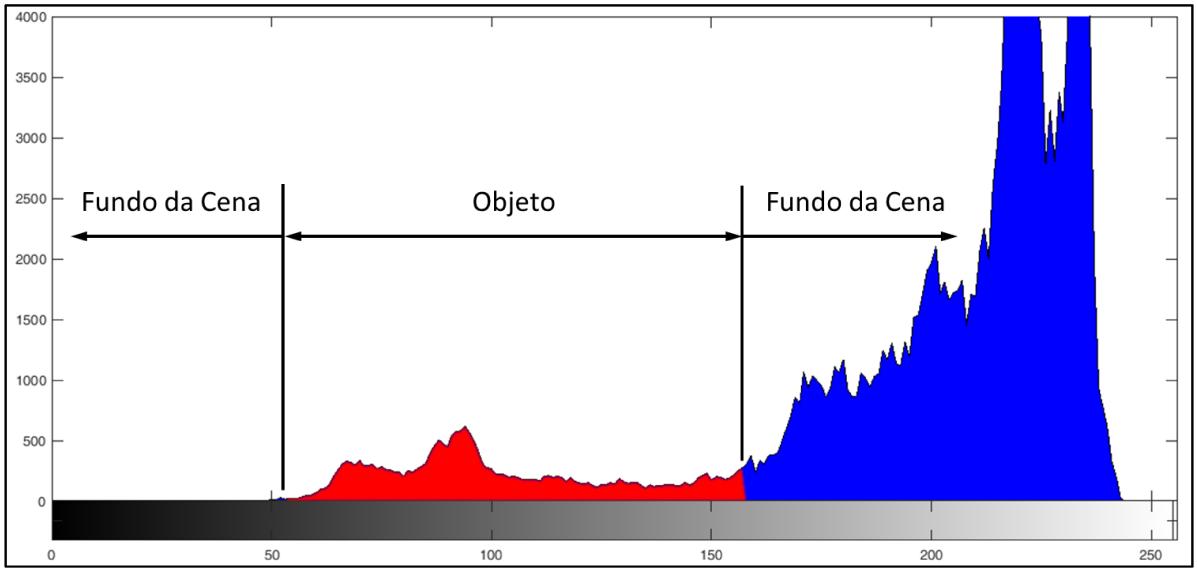
O método de Otsu utiliza o histograma da escala de cinza da imagem para identificar o limiar do que é o objeto e o que é o fundo da cena. Entretanto, algumas cores do boneco possuem nível de cinza muito próximo do nível de cinza do fundo, levando ao algoritmo remover parte do objeto. A Figura 5.2 faz uma comparação da imagem colorida com a imagem em escala de cinza. Ainda na Figura 5.2 é possível verificar uma dificuldade em distinguirmos onde termina o braço do boneco e começa o fundo da cena. A Figura 5.3 apresenta o efeito do algoritmo de Otsu no histograma da imagem.

Como apresentado nas Figura 5.1, Figura 5.2 e Figura 5.3, o método Otsu para a segmentação não é eficiente para ser utilizado na bancada de laboratório. No método Otsu o histograma da imagem é tratado como uma função densidade de probabilidade discreta, sendo a segmentação feita após o algoritmo escolher valores de limiares dentro do histograma de forma a maximizar a variância das classes.

Figura 5.2 – Comparação da imagem colorida com a imagem em escala de cinza em relação ao fundo

Fonte: Autor

Figura 5.3 – Efeito do algoritmo de Otsu no histograma da imagem



Fonte: Autor

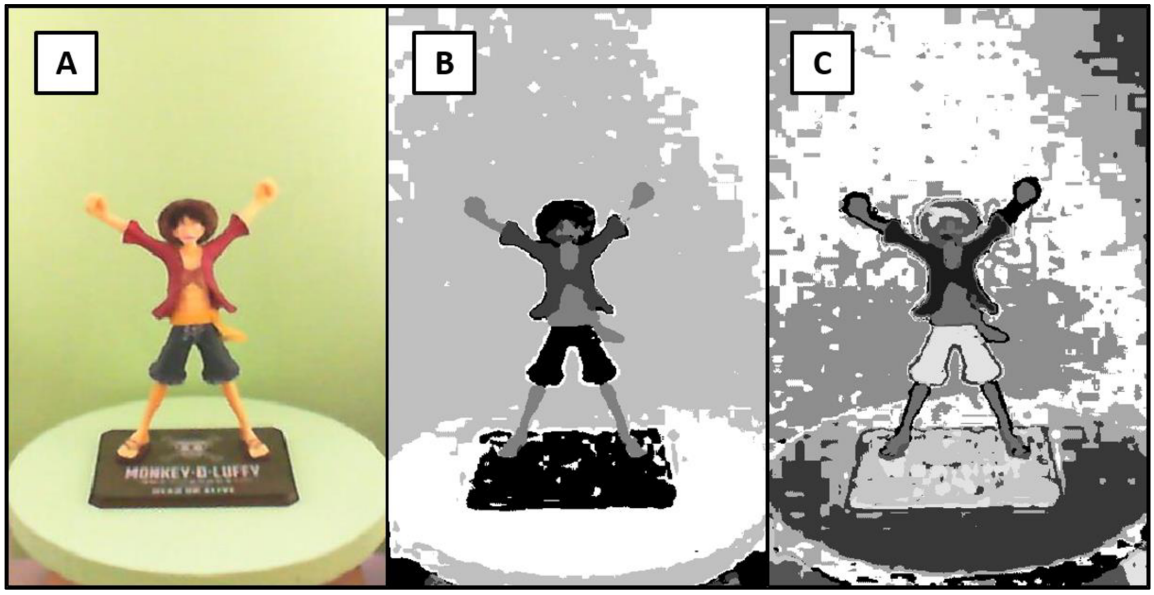
5.2.2 Segmentação baseada em cor utilizando o agrupamento K-means

O agrupamento k-means é um método de agrupamentos automáticos de dados segundo um grau de semelhança. Ele tem como objetivo particionar n observações dentre k grupos onde cada observação pertence ao grupo mais próximo

da média. Isso resulta em uma divisão do espaço de dados em um Diagrama de Voronoi.

Por meio do agrupamento k-means é possível separar diversas partes da imagem dependendo das suas características (SHI e MALIK, 2000). Aqui nesse trabalho foi utilizado para identificar o fundo da cena tentando identificar o grupo correspondente à cor verde limão. A Figura 5.4 apresenta a classificação da imagem utilizando-se 5 e 10 agrupamentos (5 e 10 clusters).

Figura 5.4 – Comparação da classificação de grupos da imagem aplicando a técnica de k-means

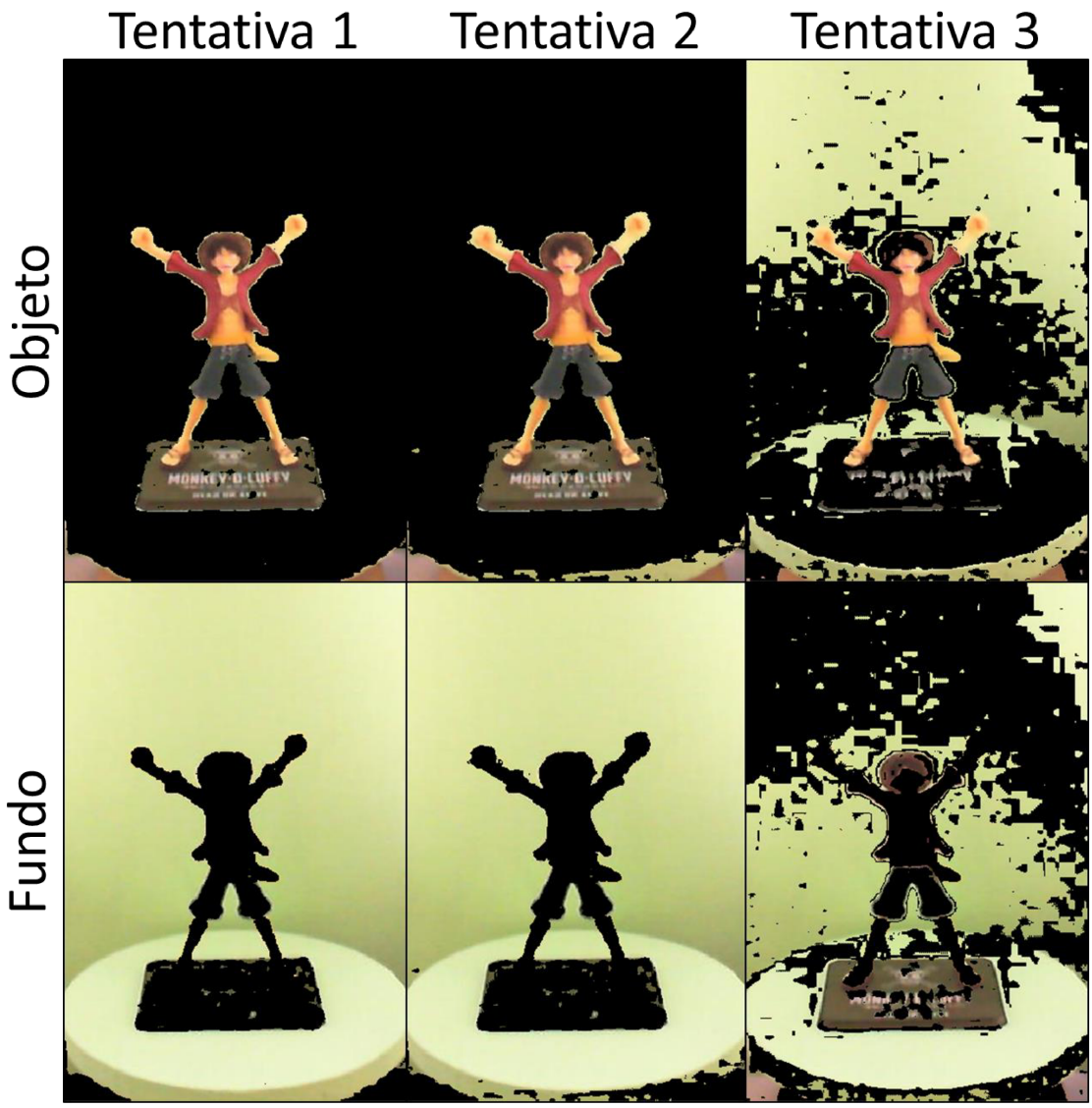


A) Imagem original, B) Resultado da classificação utilizando 5 agrupamentos e C) Resultado da classificação quando se utilizou 10 agrupamentos.
Fonte: Autor

Como pode-se ver na Figura 5.4, quanto maior for o número de agrupamentos utilizados maior é a chance de identificação do que é o objeto e o que é o fundo. Isso pode ser verificado ao notar que o braço do boneco não foi devidamente separado do fundo da cena ao utilizar 5 agrupamentos. Entretanto, ao aumentar o número de agrupamentos torna-se mais difícil fazer a classificação automática dos agrupamentos correspondentes ao fundo e ao objeto, pois a classificação por k-means utiliza um processo aleatório para tentar minimizar as influências e maximizar os resultados (ARTHUR e VASSILVITSKII, 2007). Como o processo possui uma chance estatística de classificação é possível encontrar mais agrupamentos correspondentes ao objeto em uma tentativa do que em outra. Dessa forma, ao aumentar o número de agrupamentos possíveis deve-se, também, aumentar a complexidade do

reconhecimento da cor de cada agrupamento. A Figura 5.5 apresenta diferentes resultados obtidos ao utilizar um algoritmo simples de classificação após o k-means, utilizando 10 agrupamentos para a segmentação. Apesar de alguns resultados da Figura 5.5 serem muito semelhantes ainda é possível detectar algumas diferenças.

Figura 5.5 – Possíveis variações do algoritmo de k-means usando 10 agrupamentos e um mecanismo simples de classificação da cor dos agrupamentos



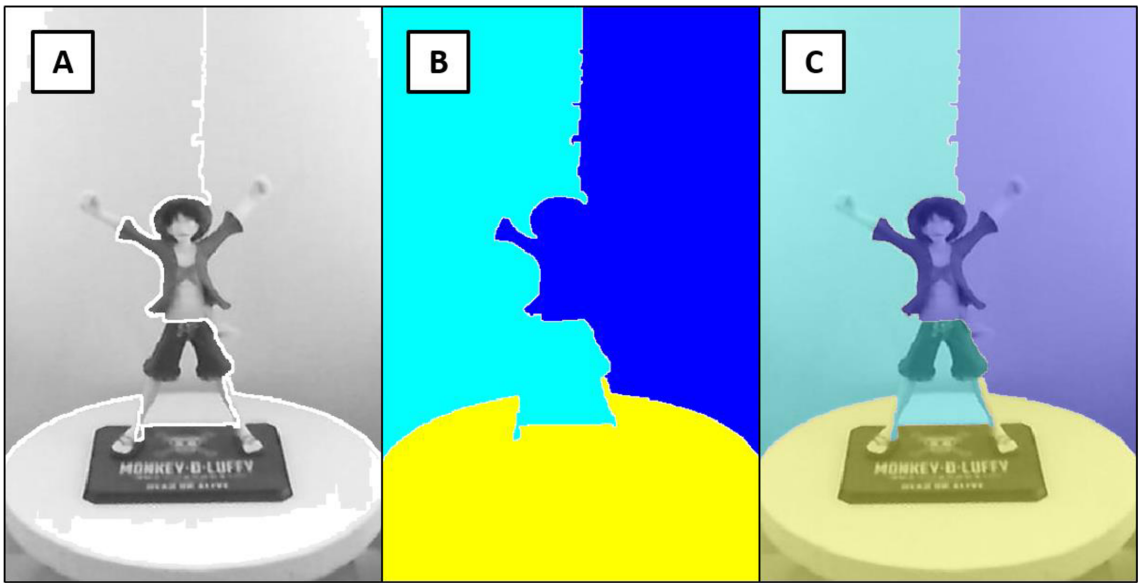
Fonte: Autor

5.2.3 Segmentação pela transformada watershed

De acordo com Roerdink e Meijster (2000) a transformada watershed pode ser descrita como uma técnica que consiste em convertermos os contornos das figuras em barreiras e, então, (hipoteticamente) enchermos as diversas regiões da imagem

com um fluido. Toda a região coberta com o mesmo fluido será considerada um objeto da imagem. Assim é possível separar diversas áreas ao observarmos os contornos existentes na imagem. A Figura 5.6 apresenta o resultado da transformada de watershed para a separação do objeto e do fundo da cena.

Figura 5.6 – Resultado da transformada de watershed para a separação do fundo e do objeto



A) Resultado da segmentação pela transformada de watershed em escala de cinza, B) Regiões encontradas pela transformada de watershed e C) Resultado da segmentação pela transformada de watershed com as regiões indicadas por cor.
Fonte: Autor

O resultado da transformada de watershed não foi satisfatório como pode ser visto na Figura 5.6, pois não conseguiu separar o objeto do fundo da cena. Isso se deve ao gradiente suave que existe entre algumas partes do boneco e o fundo da cena o que impede a obtenção do contorno nessa região. Além disso, existem diversos contornos abertos na imagem do gradiente, como pode ser visto na Figura 5.6. Na tentativa de fechar os contornos abertos foi utilizado alguns algoritmos de suavização, porem isso não foi capaz de melhorar o resultado apresentado na Figura 5.6

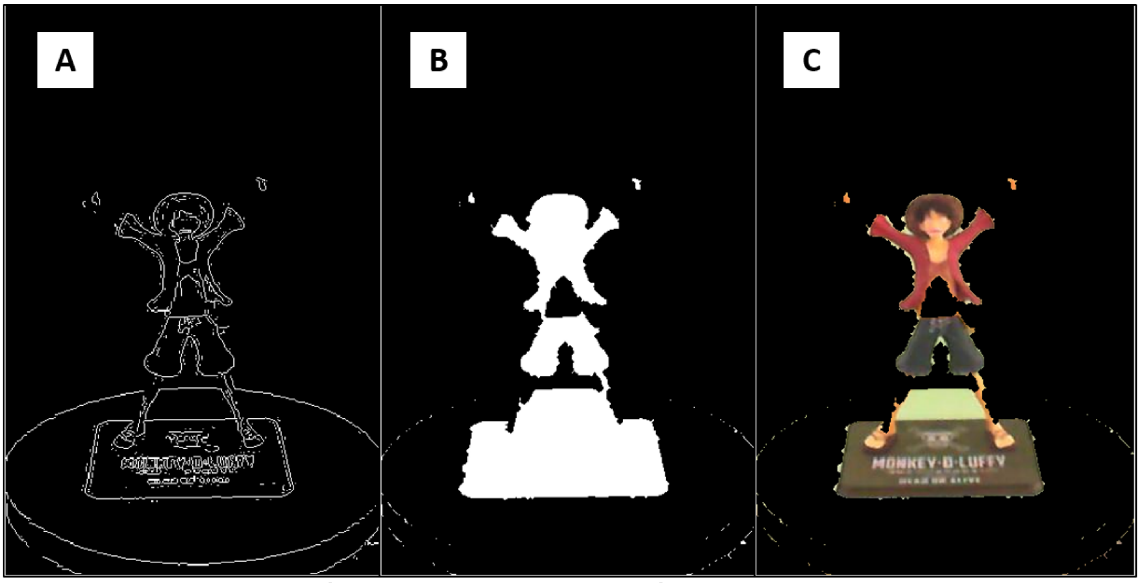
5.2.4 Segmentação pela detecção de borda e morfologia

De acordo com Caixia et al. (2014) os contornos de um objeto é uma característica importante de uma imagem e podem ser utilizados para a segmentação

de uma imagem. Dessa forma foi utilizado um processo morfológico para tentar remover o formato do objeto e diferenciá-lo do fundo da cena.

Para isso foi utilizado o operador de Sobel para encontrar os contornos do objeto. Em seguida foi feito um tratamento na imagem para preencher as áreas obtida após o operador de Sobel e, com isso, criar uma máscara capaz de extrair o objeto do fundo da cena. Os resultados obtidos são apresentados na Figura 5.7.

Figura 5.7 – Resultado da transformada da segmentação por identificação do contorno do objeto



A) resultado do operador de Sobel; B) máscara criada e C) resultado após a aplicação da máscara.
Fonte: Autor

O resultado apresentado na Figura 5.7 descartam a utilização dessa técnica para a remoção do fundo, quando o objeto de interesse possui aberturas, pois como é possível verificar parte do fundo da cena é visível entre as pernas do boneco, uma vez que o operador de Sobel vai identificar essa região como sendo uma área fechada, levando a resultados insatisfatórios. Além do mais, o pequeno gradiente entre algumas partes do boneco e o fundo (como os braços) fazem com que esse método remova algumas partes do objeto.

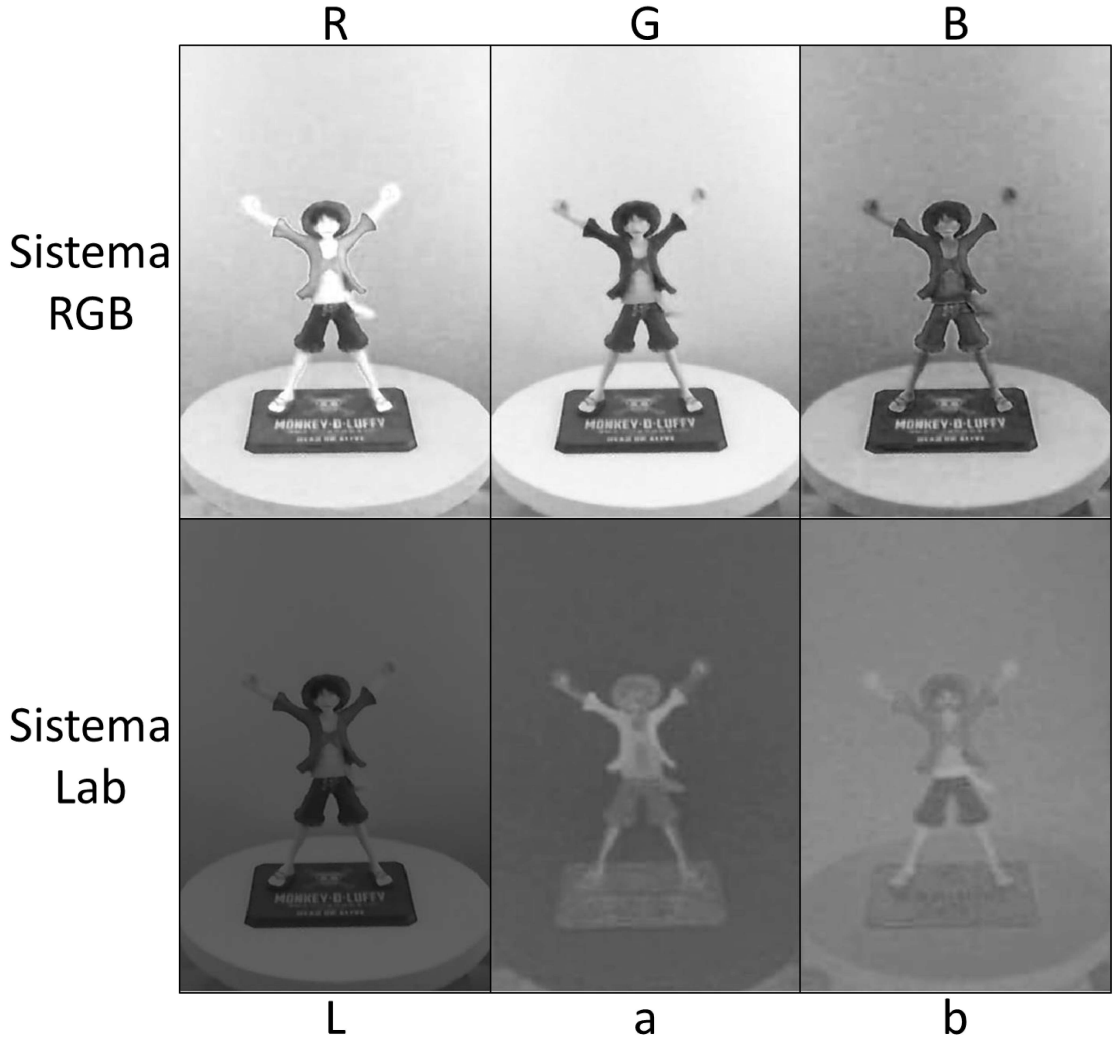
5.2.5 Segmentação por limiar no sistema de cores cieLAB

Em um sistema de espaço de cor RGB as cores são formadas pela soma de suas componentes vermelha, verde e azul. Esse espaço de cores não é interessante

para escolher um limiar e fazer a filtragem das cores da cena, pois partes do objeto podem conter a componente de dentro do limiar, como mostrado na Figura 5.8.

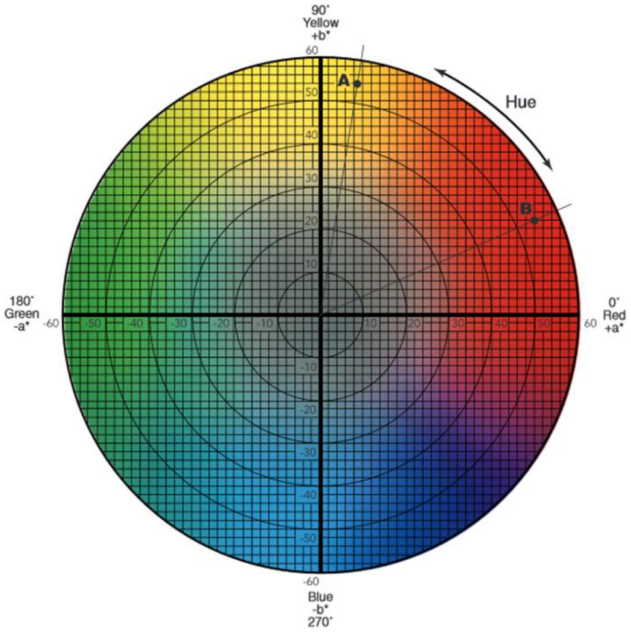
Já no sistema de espaço de cores cieLAB (também conhecido por apenas Lab) as cores são separadas nas componentes L, a e b. A componente L corresponde ao brilho da imagem enquanto as componentes a e b correspondem aos eixos do diagrama de matiz apresentado na Figura 5.9. Dessa forma fica muito mais fácil separarmos as cores no sistema Lab, pois é possível atacar diretamente uma faixa de cores próximas (nesse caso o verde limão) já que é possível especificar uma faixa de valores nos eixos a e b (BANSAL; AGGARWAL, 2011).

Figura 5.8 – Componente das cores da cena no sistema RGB e Lab.



Fonte: Autor

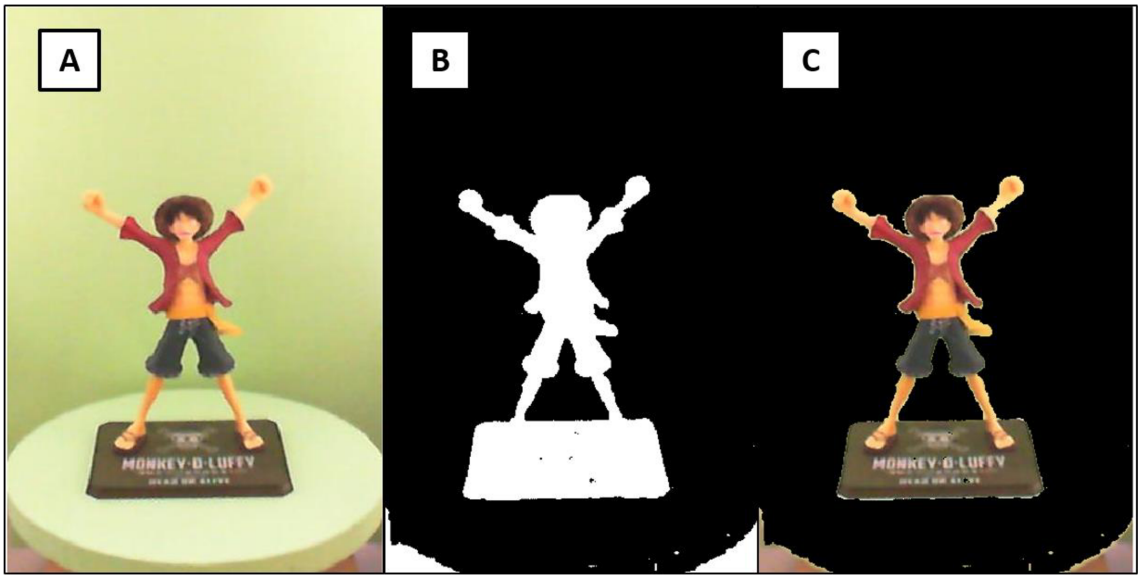
Figura 5.9 – Diagrama de matiz do sistema Lab, apresentando os eixos a e b.



Fonte: Imagem do Google

Aproveitando as vantagens do sistema de cores cieLAB é possível criar um algoritmo simples que utiliza limiares para remover as cores indesejadas da cena. O resultado obtido está apresentado na Figura 5.10.

Figura 5.10 – Resultado da segmentação por limiar no sistema de cores cieLAB



A) Imagem original; B) máscara criada e C) resultado da aplicação da máscara.
Fonte: Autor

5.2.6 Resultado dos processos de segmentação de imagens

Como foi mostrado nas seções anteriores, os métodos de segmentação apresentados que permitem, de forma satisfatória, a segmentação da imagem para separação do fundo da cena e do objeto são as segmentações utilizando o agrupamento K-means e a segmentação por limiar utilizando o sistema de cores cieLAB.

Apesar dos dois métodos supracitados permitirem a segmentação de forma satisfatória eles apresentam vantagens diferentes. Para começar, a segmentação por limiar utilizando o sistema de cores cieLAB necessita de um algoritmo mais simples e, então, execução mais rápida. Entretanto, caso o objeto possua áreas com cores semelhantes às do fundo da cena essas áreas serão removidas por essa técnica. Já a técnica de segmentação utilizando o agrupamento k-means necessita de um algoritmo mais elaborado para o seu funcionamento correto, exigindo mais tempo de processamento, mas pode não remover áreas do objeto que apresentem a mesma cor do fundo.

Por questões de simplicidade e agilidade, quando utilizado a bancada de laboratório será utilizada a segmentação pelo limiar utilizando o sistema de cores cieLAB.

5.3 Calibração da Câmera

Como descrito no Capítulo 4, não se conhece as distorções causadas pela lente da câmera. Também não é possível fixar a distância focal, pois a câmera utilizada possui um sistema de foco automático e, dessa forma, cada imagem pode possuir uma distância focal diferente.

Na tentativa de contornar os problemas causados pela câmera, foi adicionado na parte de traz do fundo da cena um padrão no formato de um tabuleiro de xadrez. Esse tabuleiro permitirá realizar a calibração da câmera. O padrão foi fixado na parte de traz da cena para não interferir no processo de segmentação.

Para a calibração deve-se virar o fundo da cena, revelando o tabuleiro. Em seguida deve-se seguir os passos da calibração e, por fim, virar o fundo da cena para que a face verde esteja voltada para a câmera. Teste com diversos objetos indicaram que a variação da distância focal pode ser desprezada ao inclinar o fundo da cena

atrás do objeto. Dessa forma, a calibração obtida não se perderá ao girar o fundo da cena.

5.3.1 Detecção do tabuleiro na cena

O objetivo da calibração de câmeras é obter os parâmetros intrínsecos da câmera. A maioria dos processos de calibração de câmeras consiste na detecção de um objeto com geometrias conhecidas na cena. A forma mais comum de fazer isso é utilizar um padrão quadriculado (preto e branco para aumentar o contraste), comumente denominado de tabuleiro (PLACHT et al., 2014).

Como o tabuleiro possui uma geometria já definida, as suas distorções na imagem são causadas pelas imperfeições da lente (ou conjunto óptico). Em termos técnicos, pode-se dizer que as deformações na imagem são causadas pelas propriedades intrínsecas da câmera.

O tabuleiro apresenta algumas características que facilitam a sua detecção na cena. O algoritmo ROCHADE (PLACHT et al., 2014) aproveita dessas características seguindo os passos:

1. Cálculo do Gradiente, para o reconhecimento das possíveis linhas do tabuleiro;
2. Limiar Local, para separar as linhas do resto da imagem;
3. Dilatação Condicional, para fechar as possíveis linhas que podem não estar contínuas;
4. Cálculo da Linha Central;
5. Extrair os Pontos de Intersecção;
6. Combinar os Pontos de Intersecção;
7. e Verificação da forma do Tabuleiro.

Os resultados da implementação do algoritmo ROCHADE é apresentado na Figura 5.11 para diversas posições do tabuleiro, demonstrando o reconhecimento com sucesso.

Figura 5.11 – Resultado do algoritmo ROCHADE reconhecendo o tabuleiro em diversas orientações na cena.

Fonte: Autor

5.3.2 Calibração da câmera

A calibração da câmera é a técnica que tem o objetivo de determinar um conjunto de parâmetros da câmera que descrevem o mapeamento dos pontos em 3D para o plano 2D. Vários métodos de calibração podem ser encontrados nas literaturas (ABDEL-AZIZ e KARARA, 1971; MELEN, 1994; SLAMA, 1980; e TSAI, 1987). Neste trabalho utilizou-se o método de quatro passos para a calibração da câmera, descrito em Heikkilä e Silvén (1997). Este método foi escolhido pois, diferente da maioria dos métodos, este se preocupa com os processos durante a calibração, como: controle dos pontos de extração de informação da imagem, modelo de ajuste, correção da imagem e, também, dos erros envolvidos em cada estágio.

Os quatros passos para a calibração podem ser descritos como: Estimação dos parâmetros lineares; estimação dos parâmetros não-lineares; ajustes caso os pontos de controle que possuam projeções maiores do que o tamanho de um pixel; e, por fim, a correção da imagem.

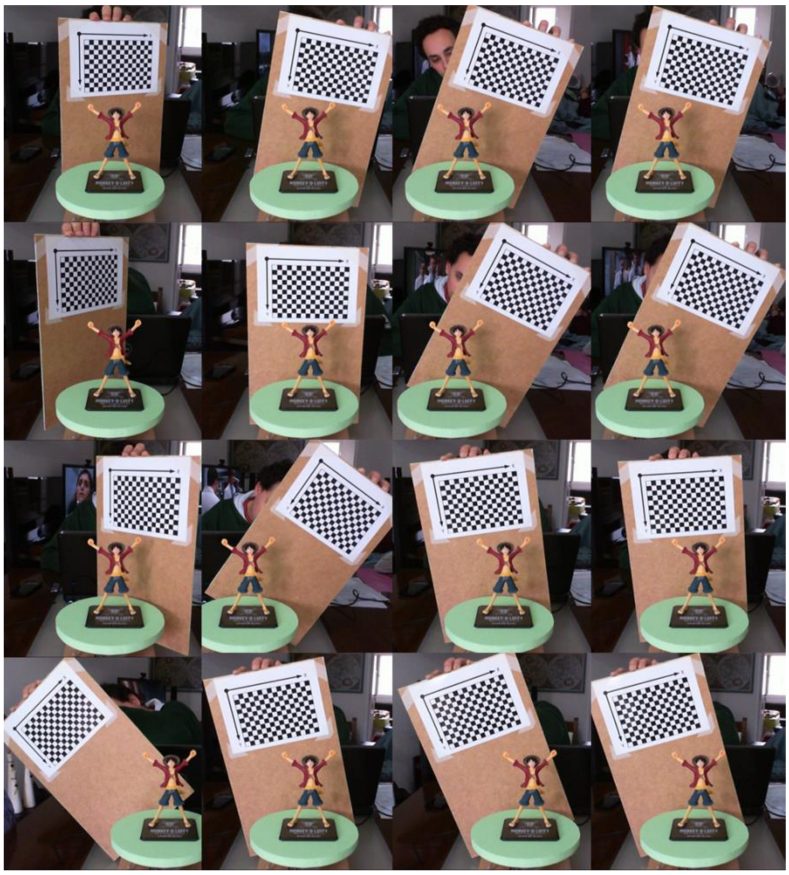
O método da transformação linear direta (DLT) é a técnica utilizada para a estimação dos parâmetros lineares e não-lineares. Como a matriz de característica intrínseca da câmera, representada pela Equação (2.26), apresenta cinco parâmetros a DLT leva a um sistema linear com $2n \cdot 6$ incógnitas, sendo n o número de amostras (imagens do tabuleiro). Dessa forma, só será possível obter uma solução única para a matriz de característica intrínsecas da câmera se forem utilizados três ou mais imagens do tabuleiro para a calibração. Para uma calibração adequada, é aconselhada a utilização de mais de três amostras, sendo possível melhorar a precisão do resultado pela da estimação da máxima verossimilhança (ZHANG, 2000). Entretanto, deve-se tomar cuidado para que as posições do tabuleiro não sejam muito próximas ou similares, pois isso impede do sistema linear a convergir em um valor.

Dessa forma, pode-se resumir o procedimento de calibração para a bancada de laboratório da seguinte forma:

1. Coloca-se o objeto sobre mesa giratória;
2. Coloca-se o fundo da cena com o tabuleiro voltado para a câmera;
3. Tira-se várias fotografias com o tabuleiro em posições diferentes;
4. Calcula-se a estimativa dos parâmetros intrínsecos através do método de quatro passos descrito em Heikkilä e Silvén (1997).

A Figura 5.12 apresenta a utilização dessa metodologia para a calibração da câmera em que foi utilizado 15 amostras. A Figura 5.12 apresenta apenas a área das amostras em que se visualiza o tabuleiro, omitindo o restante da imagem na sua apresentação.

Figura 5.12 – Parte das imagens utilizadas para a calibração da câmera com a Bancada de Laboratório.

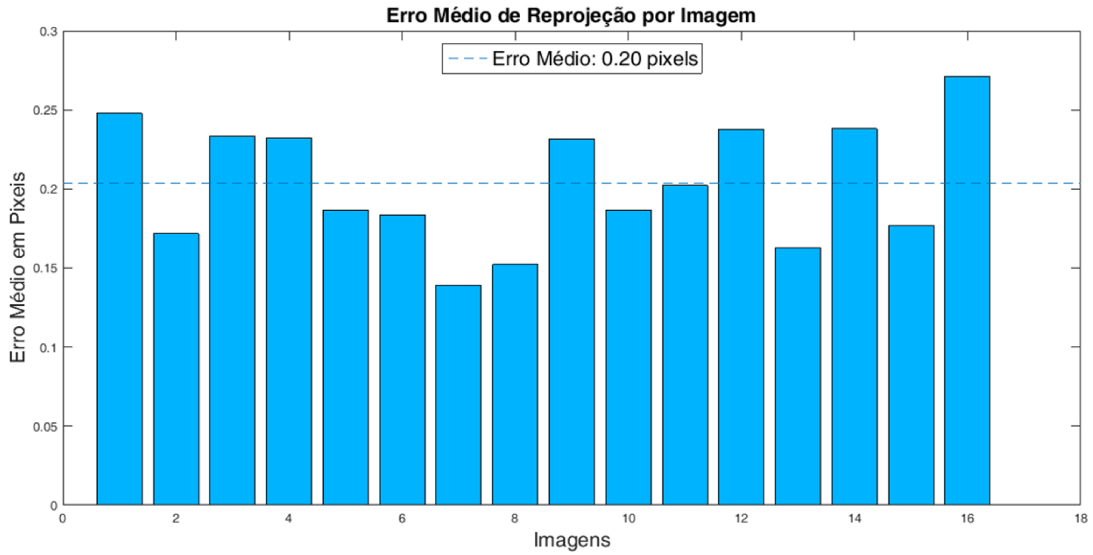


Fonte: Autor

O procedimento de calibração descrito acima levou à resultados satisfatórios, com erro médio de 0,20 pixel. A Figura 5.13 apresenta o erro de reprojeção para cada uma das imagens utilizadas na calibração.

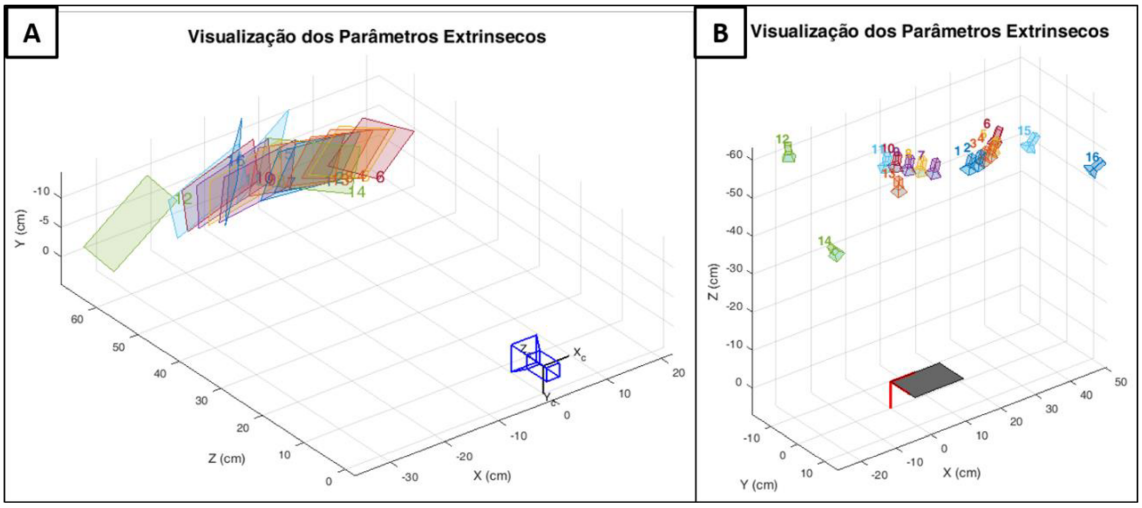
Para verificar se ocorreram erros durante o processo de calibração calculou-se, também, as reprojeções de cada câmera, sendo o resultado apresentado na Figura 5.14. Pode-se verificar que o algoritmo não gerou resultados inconsistentes, pois, de acordo com a Figura 5.14, não foram encontradas reprojeções de câmeras absurdas, como, por exemplo, câmeras de ambos os lados do tabuleiro.

Figura 5.13 – Erro médio de reprojeção para cada imagem utilizada na calibração.



Fonte: Autor

Figura 5.14 – Resultado da reprojeção das câmeras com o resultado da calibração



A) representação a câmera parada e a visualização das várias posições do tabuleiro. B) Utilizando o tabuleiro como referência e encontrou-se as coordenadas de cada câmera.
Fonte: Autor

Para finalizar, os parâmetros intrínsecos encontrados, durante este teste, foram:

$$[\mathrm{H}] = \begin{bmatrix} 940{,}1833 & 0.0 & 651.7856 \\ 0.0 & 940{,}7888 & 317.7112 \\ 0.0 & 0.0 & 1{,}0 \end{bmatrix} \qquad (5.1)$$

Também foram calculadas, durante a calibração, as distâncias da câmera aos centros dos tabuleiros, levando a uma distância média de 60 cm, coerente com a

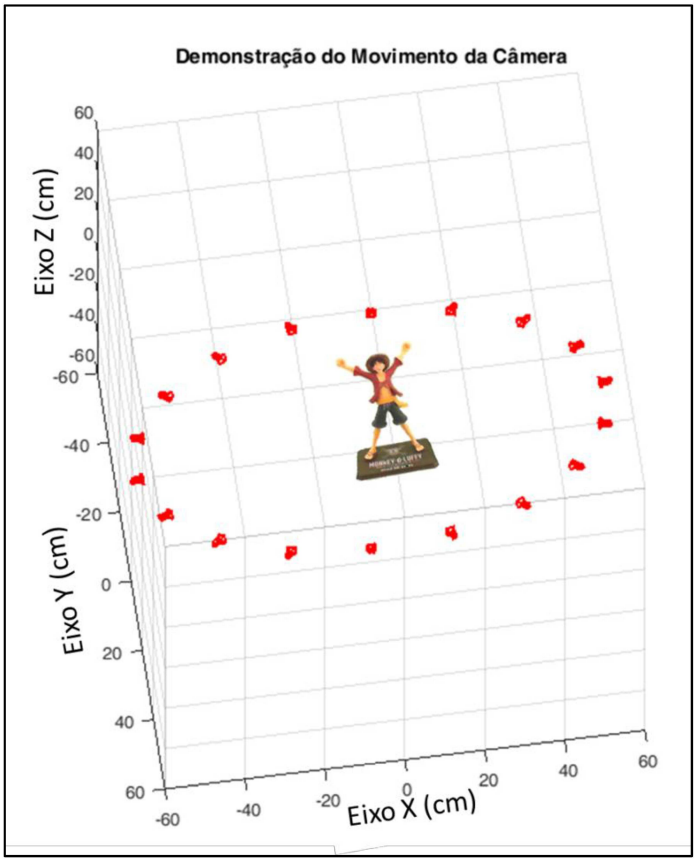
distância da bancada de laboratório. Dessa forma, pode-se dizer que os resultados obtidos na calibração são satisfatórios.

5.4 Movimento da Câmera e Característica Extrínseca

Conhecendo os parâmetros intrínsecos da câmera precisa-se, agora, conhecer os parâmetros extrínsecos, ou seja, os pontos de captura das imagens das vistas do objeto (ou posição das câmeras).

A bancada foi projetada na forma mais simples, a câmera fica parada enquanto o objeto é racionado sobre a mesa giratória. Entretanto, o equacionamento necessário para descrever o objeto girando torna essa aproximação difícil. Para facilitar a abordagem matemática foi utilizada a inversão de mecanismo. Com a inversão de mecanismo o centro de coordenadas global foi definido como o centro da base giratória e, por essa técnica, a câmera é visualizada girando ao redor do objeto, como apresentado na Figura 5.15.

Figura 5.15 – Representação gráfica da inversão de movimento apresentando a câmera girando ao redor do objeto.



Fonte: Autor

Definiu-se, também, que o eixo vertical do objeto é o eixo Y do sistema de coordenadas global e, dessa forma, o movimento da câmera é de girar ao redor do eixo Y sobre o plano ZX (ou plano de azimute).

Assim, a rotação da câmera pode ser descrita como sendo (PAUL, 1981):

$$\mathrm{R} = \begin{bmatrix} \cos(\theta) & 0 & \mathrm{sen}(\theta) \\ 0 & 1 & 0 \\ -\mathrm{sen}(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{5.2}$$

em que θ representa o ângulo de giro dado pela câmera ao redor do objeto (ou a mesa giratória) para obter a imagem.

Além da rotação, a câmera está, aproximadamente, a 60 cm de distância do objeto (de acordo com a calibração). Se considerarmos a posição inicial da câmera sobre o eixo X pode-se descrever essa posição através do vetor de translação (PAUL, 1981)

$$\mathrm{T} = \begin{bmatrix} 60 \\ 0 \\ 0 \end{bmatrix} \tag{5.3}$$

Por fim, todas as posições das câmeras conhecidas através do produto vetorial da Equação (5.3) com a Equação (5.2)

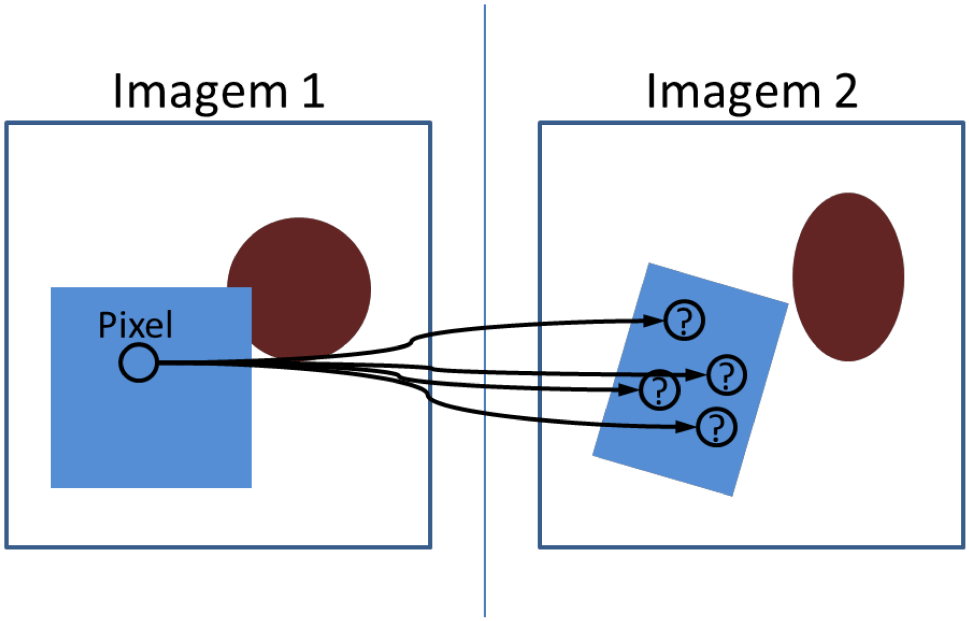
$$\mathrm{P} = [\mathrm{T}] \times [\mathrm{R}] \tag{5.4}$$

A Equação (5.4) define todas as equações de posições das câmeras em função do ângulo de giro do motor e, dessa forma, as características extrínsecas.

5.5 Detecção dos Pontos Característicos de uma Imagem

Após conhecer as caraterísticas intrínsecas e extrínsecas da câmera basta relacionar os pontos de uma imagem na outra. Entretanto, devido à grande quantidade de pixel em uma imagem, torna-se inviável tentar correlacionar todos os pontos de uma imagem com os pontos da outra imagem, pois pixels de valores semelhantes (cores) podem não serem localizados corretamente na outra imagem. A Figura 5.16 apresenta a dificuldade na relação direta pixel a pixel.

Figura 5.16 – Demonstração da incerteza da posição do Pixel, marcado na Imagem 1, na Imagem 2



Fonte: Autor

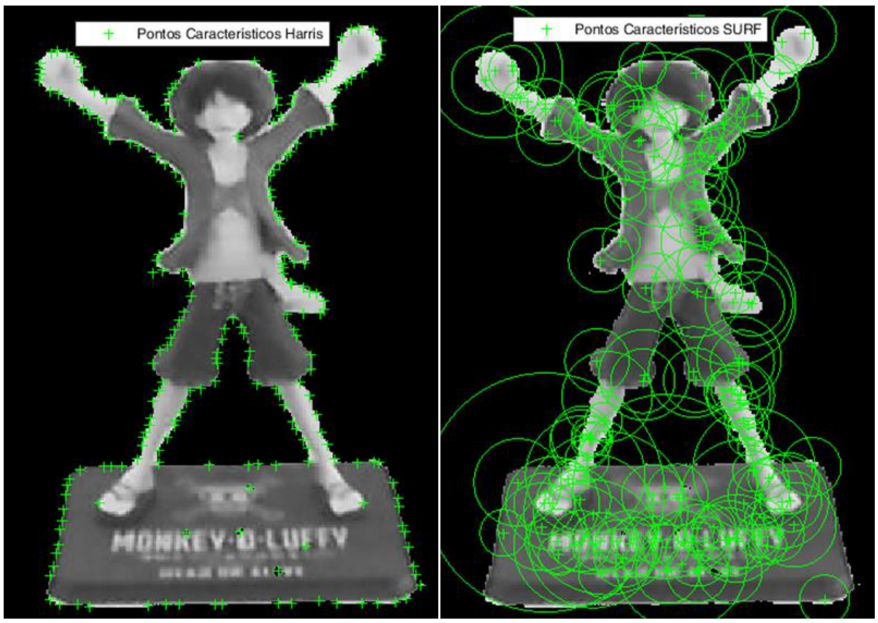
Para contornar essa dificuldade, será usado a abordagem por *features* ou pontos característicos. A vantagem dos pontos característicos sobre os pixels se deve ao fato dos pontos característicos representarem o centro de um padrão ou de uma estrutura distinta encontrada na imagem. Dessa forma, reduzem as possibilidades de possíveis posições na segunda imagem e, também, a quantidade de pontos que devem ser analisados. Para essa tarefa existem vários detectores de pontos característicos, a saber, FAST (ROSTEN e DRUMMOND, 2006); Harris (HARRIS e STEPHENS, 1988); SURF (BAY, TUYTELAARS e GOOL, 2006); BRISK (LEUTENEGGER, CHLI e SIEGWART, 2011); e MSER (MATAS et al., 2002).

No atual trabalho será utilizado apenas os detectores de Harris e SURF. A vantagem do SURF é a capacidade do reconhecimento de regiões independente da escala e rotação, sendo adequado aos testes feitos com a bancada de laboratório. Já o detector de Harris apresenta uma boa capacidade de identificação de características em estruturas que apresentem quinas e arestas, sendo preferencial para estruturas de engenharia civil (XIAO e QUAN, 2009), tornando-o preferencial à segunda parte desse trabalho.

A Figura 5.17 apresenta a comparação entre os algoritmos de Harris e de SURF. Pode-se verificar na Figura 5.17 que o algoritmo de Harris conseguiu pegar apenas pontos no contorno no objeto em teste, enquanto o algoritmo SURF conseguiu detectar pontos característicos no interior do objeto. Além de ter conseguido detectar

pontos no interior do objeto, o algoritmo SURF também carrega informações como escala, métrica, sinal do Laplaciano e orientação da característica. Todas essas informações foram traduzidas nos círculos da Figura 5.17 e serão utilizadas para o casamento entre os pontos característicos de uma imagem na outra.

Figura 5.17 – Diferentes pontos característicos identificados pelo algoritmo Harris e o de SURF



Fonte: Autor

Apesar do algoritmo de Harris apresentar-se inferior ao algoritmo SURF nos testes com a bancada de laboratório, de acordo com Xiao e Quan (2009), ele torna-se mais adequado para análises de estruturas de geometria bem definida, como as da engenharia civil. Dessa forma o algoritmo de Harris não será aproveitado na próxima etapa do trabalho.

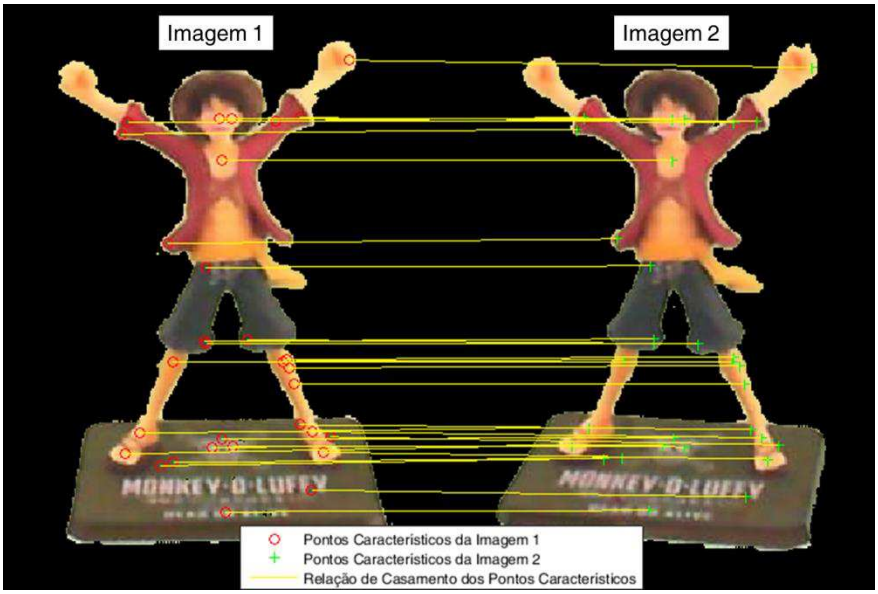
5.5.1 Casamento de pontos característicos em duas imagens

Após a identificação dos pontos característicos em duas imagens distintas, resta fazer o casamento desses pontos para identificar os seus pares. Dessa forma será possível fazer a triangulação desses pontos e obter, assim, a distância do ponto à câmera, ou seja, será retornada a coordenada perdida durante o processo de captura da cena 3D em uma imagem 2D.

O casamento dos pontos é feito através de testes de compatibilidade dos pontos nas duas imagens, sendo esse teste de compatibilidade feito utilizando o

RANSAC como descrito no Capítulo 2. A Figura 5.18 apresenta o resultado do casamento de pontos. É importante verificar, também, que a Figura 5.18 apresenta menos pontos característicos do que a Figura 5.17. Isso ocorre porque, apesar de terem sido encontrados mais pontos característicos (Figura 5.17), o algoritmo não conseguiu fazer o casamento de todos eles, seja por oclusão ou o erro de casamento superou o limiar estabelecido.

Figura 5.18 – Casamento de pontos característicos da Imagem 1 com os da Imagem 2

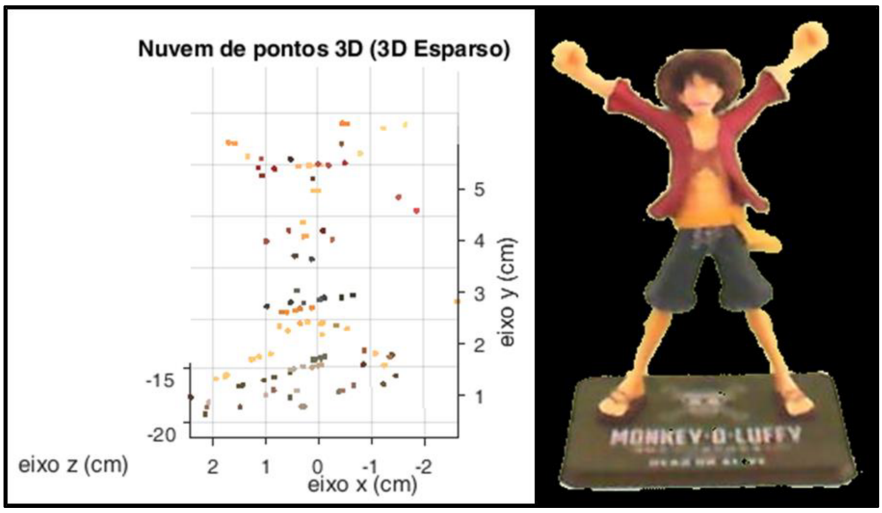


Fonte: Autor

5.6 Triangulação dos Pontos Característicos e o 3D Esparso

Conhecido os casamentos dos pontos característicos e as matrizes das câmeras (características intrínsecas e extrínsecas) é possível fazer a triangulação desses pontos casados, conforme demonstrado no Capítulo 2, e, dessa forma, encontrar a distância deles às câmeras. Conhecido a distância dos pontos e as posições da câmera torna-se possível o posicionamento desses pontos no espaço e, assim, criar uma nuvem de pontos denominada de 3D Esparso. A Figura 5.19 apresenta a nuvem de pontos 3D e a comparação com uma imagem do objeto utilizando 6 imagens.

Figura 5.19 – 3D Esparso e a comparação do objeto.



Fonte: Autor

Apesar da Figura 5.19 apresentar um resultado promissor ela também apresenta a necessidade de ajustes na posição dos pontos, pois alguns pontos são apresentados distante de sua posição original. Este resultado foi obtido pelo casamento de todas as imagens com todas as outras cinco.

5.7 Considerações Finais

Neste capítulo foi apresentado como utilizar a bancada experimental e as metodologias matemáticas apresentadas nos Capítulos 2 e 3 para a reconstrução 3D de um objeto utilizando apenas representações 2D deste objeto.

A bancada experimental deste capítulo permite ter um conhecimento *a priori* das posições da câmera. No próximo capítulo é mostrado como obter as posições de câmera tendo como referência apenas as representações 2D da cena.

Capítulo 6

6 RECONSTRUÇÃO 3D DE CENAS ATRAVÉS DE CÂMERAS COM MOVIMENTOS NÃO DEFINIDOS

No Capítulo 4 foi apresentado a bancada de testes para trabalhar com reconstrução 3D de modelos através de fotografias. Esta bancada permite ter um conhecimento *a priori* das posições de câmera e, assim, simplifica o problema de reconstrução 3D por conhecer as posições da câmera.

Neste capítulo será demonstrado realizar a reconstrução 3D de cenas quando não se tem o conhecimento *a priori* das posições de câmera.

6.1 Introdução

Até o momento o problema da reconstrução 3D havia sido simplificado devido ao fato de que as posições da câmera são conhecidas ou o seu movimento é bem definido, e, assim, pode-se aproveitar as equações da robótica para encontrar essas posições. Com isso pode-se, facilmente, realizar a triangulação dos pontos característicos para obter a nuvem de pontos no espaço. Assim, o algoritmo de reconstrução 3D deve ser capaz de realizar as seguintes tarefas:

1. Obter os parâmetros intrínsecos da câmera (calibração da câmera);
2. Obter os parâmetros extrínsecos de cada posição de câmera (ou conhecido a priori ou calculado);
3. Obtenção dos pontos característicos da imagem;
4. Casamento dos pontos característicos;
5. Triangulação para a obtenção do 3D.

Entretanto, quando não se conhece as posições da câmera e o movimento dela não é definido (não é possível descrever, matematicamente, as posições da câmera em função do movimento dela) o algoritmo acima não pode ser utilizado, pois falta informações para referente à terceira etapa dele.

Porém, ao considerar que todas as imagens pertencem à mesma cena pode-se tentar calcular a posição de uma câmera em relação a uma outra ao relacionar

pontos comuns das duas imagens e, assim, obter os parâmetros extrínsecos de cada posição de câmera. Conhecido os parâmetros extrínsecos de cada posição de câmera pode-se, então, ser feito o processo de triangulação para obter a nuvem de pontos 3D da cena.

6.2 Metodologia para Obter as Posições da Câmera Através da Análise das Imagens

Ao considerarmos um conjunto de pontos $\{[u_i, u_i']\}_{i=1}^{m}$, em que u_i corresponde aos pontos característicos em uma imagem e u_i' aos pontos característicos em outra imagem da mesma cena, haverá, então, uma transformação homogênea H que consiga relacionar os pontos u_i nos pontos u_i', como apresentado pela Equação (2.12).

Ao fazer o casamento dos pontos do conjunto $\{[u_i, u_i']\}_{i=1}^{m}$ verifica-se que alguns casamentos de pontos tendem a seguir o efeito de uma determinada transformação homogênea. Isso permite alguns algoritmos para calcular a matriz fundamental F. Os valores da matriz fundamental capturam todas as informações que podem ser obtidas sobre um par de câmeras a partir das correspondências nas imagens. A subseção 3.7.4 apresenta diversos algoritmos que permitem estimar a matriz fundamental através da análise dos casamentos dos pontos do conjunto $\{[u_i, u_i']\}_{i=1}^{m}$. Para este trabalho, optou-se em utilizar o algoritmo de estimação da matriz fundamental pela máxima verossimilhança, utilizando-se do algoritmo de RANSAC para determinar quais os pontos são adequados (inliers) e quais os pontos são discrepantes (outliers).

Ao considerar a transformação homogênea que relaciona as duas imagens como sendo uma transformação afim pode-se relacionar os pontos u_i e u_i' com as matrizes de calibração de cada câmera K e K' como apresentado pelas Equações (3.20) e (3.21). Aplicando a restrição epipolar, as Equações (3.20) e (3.21) podem ser escritas na forma da Equação (3.22). Na Equação (3.22) surge, então, a matriz essencial E.

A matriz essencial E captura as informações a respeito do movimento relativo da segunda câmera em relação à primeira, sendo essa relação descrita pela translação $\mathbf{t}$ e rotação R da segunda câmera em função da primeira. Ou seja, através

da matriz essencial podemos descrever as matrizes de câmeras como sendo $M = K[\boldsymbol{I}|\mathbf{0}]$ (matriz que representa a primeira câmera) e $M' = K'[R| \quad -R\mathbf{t}]$ (matriz que representa a segunda câmera).

Assim, é possível decompor a matriz essencial em translação e rotação de acordo com os seguintes passos:

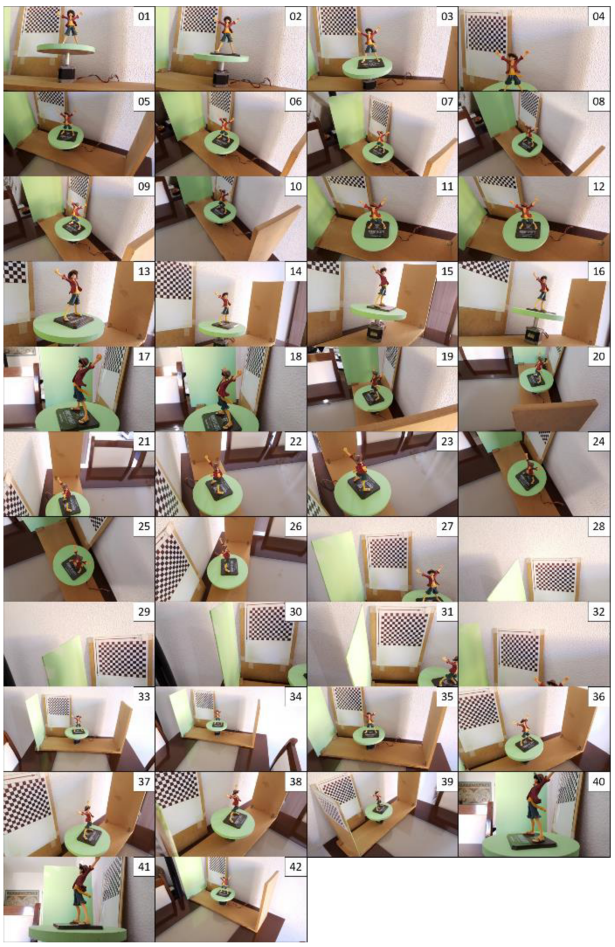
1. Estimar a matriz fundamental F pelos pontos correspondentes (subseção 3.7.4);
2. Calcular $E = K'^{\mathrm{T}} \cdot F \cdot K$;
3. Decompor E em $\mathbf{t}$ e R.

Entretanto, como apresentado na subseção 3.7.2.1, ao calcular a matriz essencial através de pontos correspondentes de duas imagens ela terá uma escala desconhecida devido as incertezas sobre as posições de câmera. Com isso, a cena será construída mantendo a proporção dos elementos contidos na cena, porém ela não terá o tamanho real. Pode-se dizer, então, que a sena será reconstruída até uma transformação de similaridade.

6.3 Resultado da Metodologia para Obter as Posições da Câmera Através da Análise das Imagens

Para a avaliação da metodologia de reconhecimento das posições de câmeras através das análises das imagens, criou-se um banco de imagens, contendo 42 imagens, da bancada de testes apresentada na subseção 2.1 e na Figura 4.1. Esse banco de imagens está apresentado na Figura 6.1. Ao observar a Figura 6.1 verifica-se que foi adicionado à cena um padrão quadriculado em que se conhece o tamanho e a forma deste padrão. Através desse padrão foi possível obter os parâmetros intrínsecos da câmera utilizada.

Figura 6.1 – Banco de imagens do teste de detectar as posições da câmera através da análise de imagens



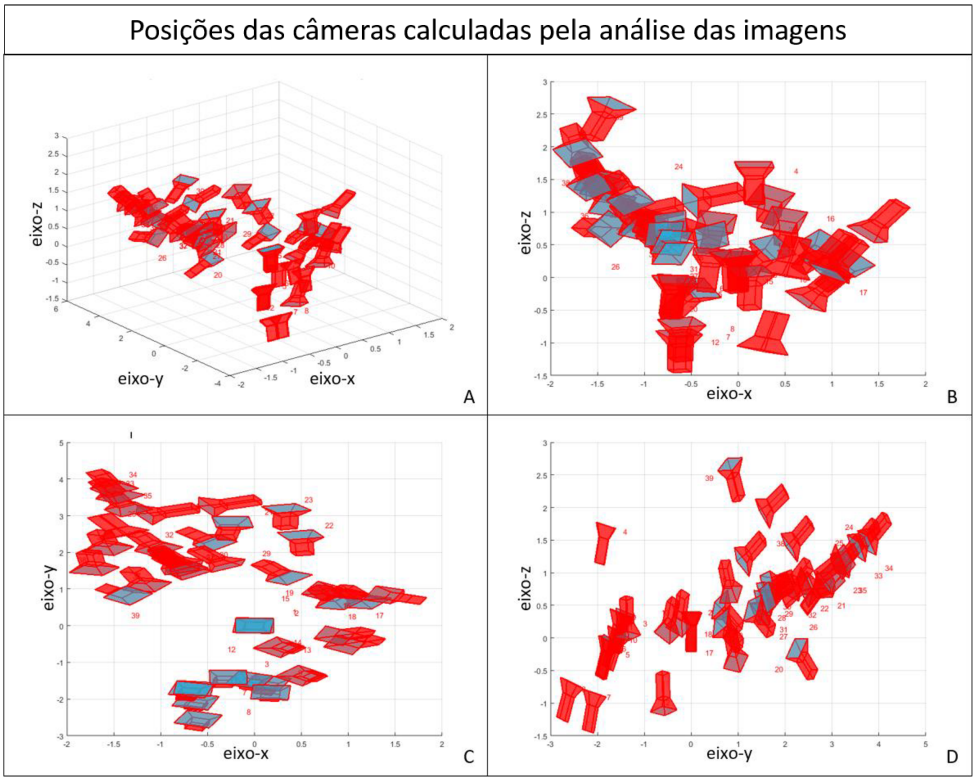
Fonte: Autor

Ao aplicar a metodologia descrita no banco de imagens apresentado na Figura 6.1 obteve-se 39 posições de câmera que estão apresentadas na Figura 6.2. Antes de localizar as posições de câmeras ordenou-se as imagens de forma a encontrar os pares com a maior quantidade de pontos homólogos para o cálculo da matriz fundamental. Para simplificar o processo de ordenamento, cada imagem foi utilizada em apenas dois pares. Por fim, as imagens foram nomeadas com a sequência encontrada, como apresentado na Figura 6.1, e o resultado do processo de ordenamento está resumido na Tabela 6.1.

A diferença entre o número de imagens utilizadas, 42 imagens, e o número de posições de câmera encontradas, 39 câmeras, se deve ao fato que a sequência de imagens utilizada, apesar de ter maximizado a possibilidade do cálculo da matriz fundamental, não foi o suficiente para calcular todas as posições de câmera, pois alguns pares de imagens não apresentaram uma quantidade suficiente de pontos

homólogos para o cálculo da matriz fundamental ou os pontos obtidos não estavam distribuídos de forma a permitir calcular a matriz fundamental, como no par das imagens 39 e 40 em que não foi possível calcular a posição da câmera 40 tendo a posição da câmera 39 como referência.

Figura 6.2 – Representação das posições da câmera calculadas pela análise das imagens da bancada de testes



A) vista em perspectiva das posições; B) vista do plano X-Z; C) Vista do plano X-Y; D) Vista do plano Y-Z
Fonte: Autor

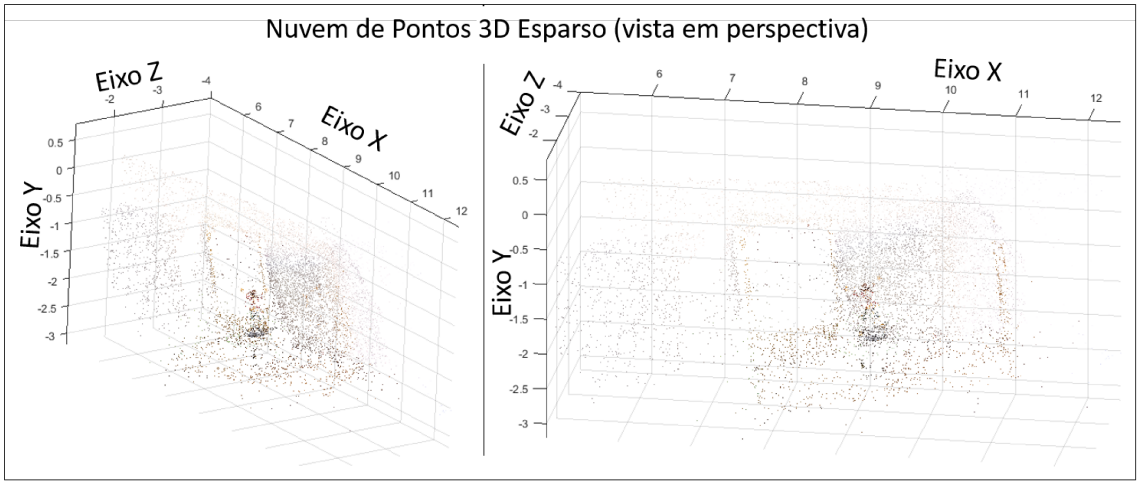
Obtido diversos pontos característicos, encontrado os pontos homólogos nos pares de imagem e, por fim, conhecido as posições das câmeras realizou-se o processo de triangulação, utilizando os pontos obtidos e as posições de câmeras calculadas. Como resultado obteve-se a nuvem de pontos 3D apresentada na Figura 6.3. Uma visão mais clara da nuvem de pontos é fornecida pela Figura 6.4 que apresenta a nuvem de pontos detalhando o plano X-Y; pela Figura 6.5 na qual é possível visualizar a nuvem de pontos pelo plano X-Z; e pela Figura 6.6 na qual é possível visualizar na nuvem de pontos tendo como referência o plano Y-Z.

Tabela 6.1 – Relação de casamento de pontos característicos para o cálculo das posições da câmera

Par de imagem utilizado	Quantidade de pontos homólogos encontrados	Consegue-se calcular a posição da câmera	Par de imagem utilizado	Quantidade de pontos homólogos encontrados	Consegue-se calcular a posição da câmera
01->02	2110	Sim	22->23	210	Sim
02->03	570	Sim	23->24	25	Sim
03->04	289	Sim	24->25	198	Sim
04->05	153	Sim	25->26	23	Sim
05->06	216	Sim	26->27	17	Sim
06->07	494	Sim	27->28	194	Sim
07->08	309	Sim	28->29	194	Sim
08->09	151	Sim	29->30	113	Sim
09->10	114	Sim	30->31	144	Sim
10->11	19	Sim	31->32	29	Sim
11->12	968	Sim	32->33	28	Sim
12->13	22	Sim	33->34	63	Sim
13->14	279	Sim	34->35	75	Sim
14->15	142	Sim	35->36	33	Sim
15->16	102	Sim	36->37	133	Sim
16->17	28	Sim	37->38	406	Sim
17->18	590	Sim	38->39	26	Sim
18->19	87	Sim	39->40	10	Não
19->20	71	Sim	40->41	4	Não
20->21	14	Sim	41->42	4	Não
21->22	56	Sim	Total de Pontos utilizados		8.713

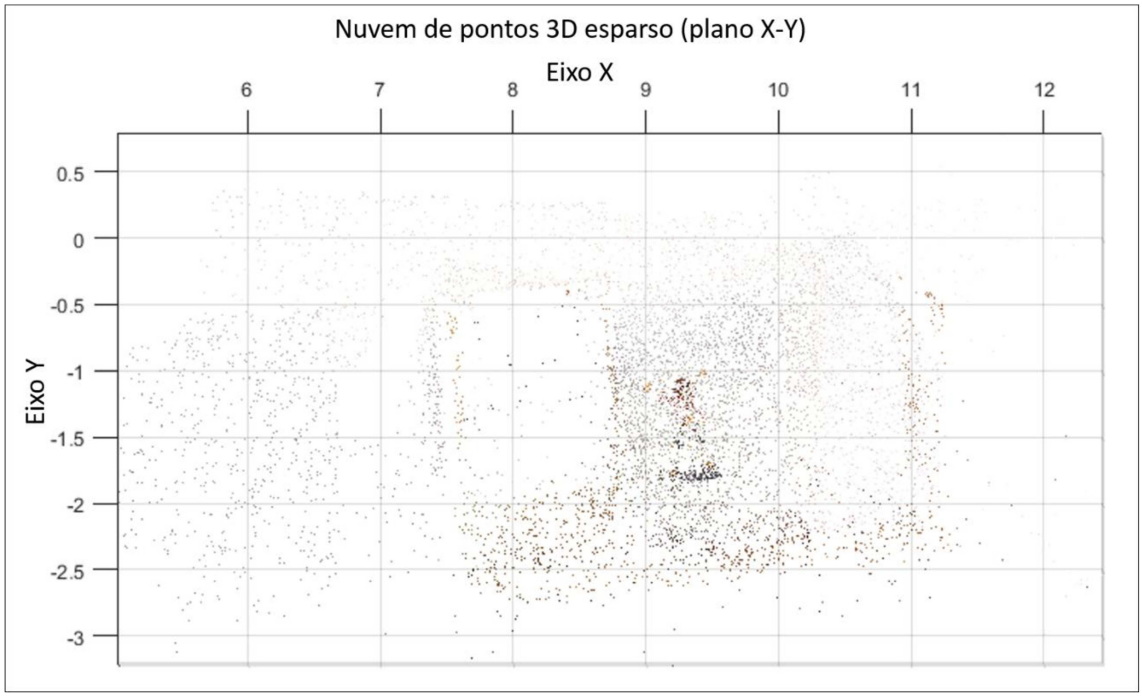
Fonte: Autor

Figura 6.3 – Nuvem de pontos 3D esparso representando a bancada de testes com 8.713 pontos



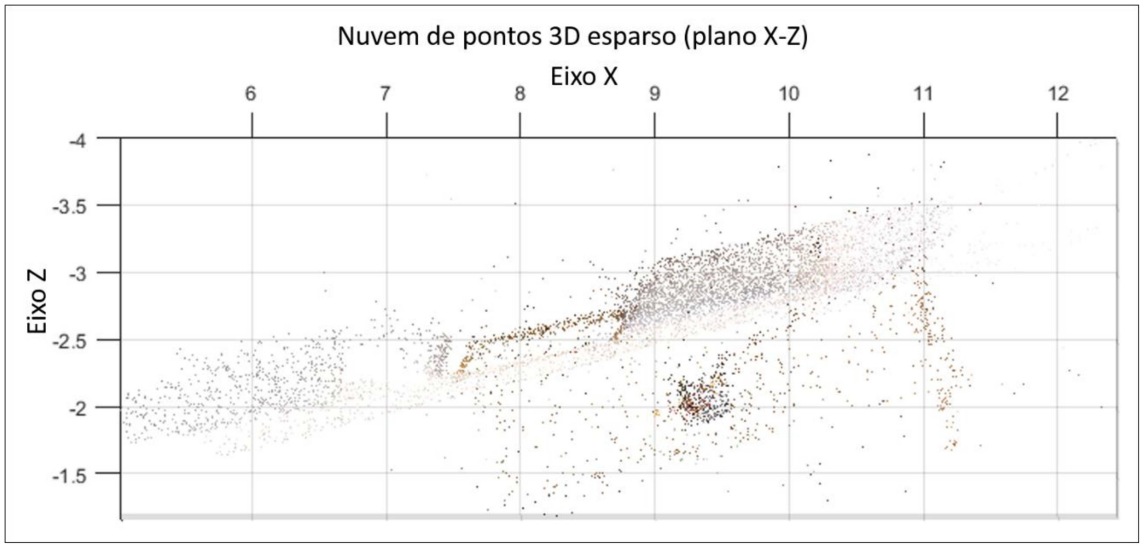
Fonte: Autor

Figura 6.4 – Nuvem de pontos 3D esparso representando a bancada de testes por uma vista no plano X-Y



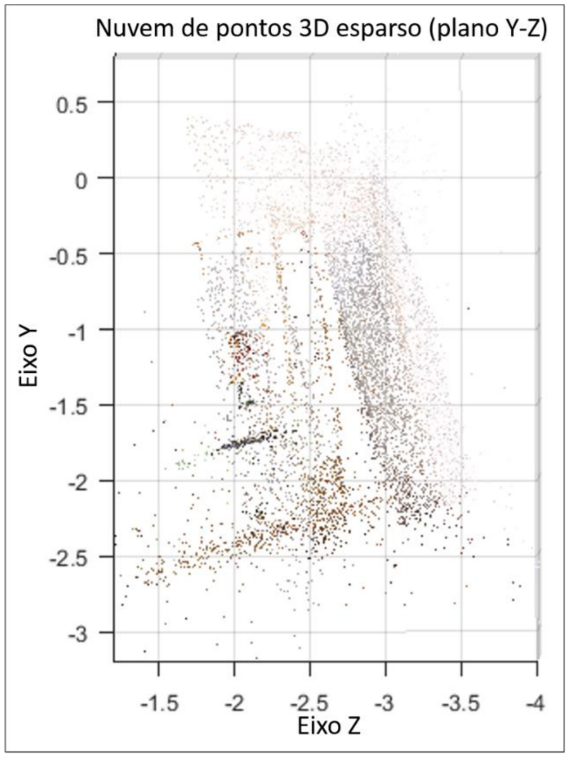
Fonte: Autor

Figura 6.5 – Nuvem de pontos 3D esparso representando a bancada de testes por uma vista no plano X-Z



Fonte: Autor

Figura 6.6 – Nuvem de pontos 3D esparso representando a bancada de testes por uma vista no plano Y-Z



Fonte: Autor

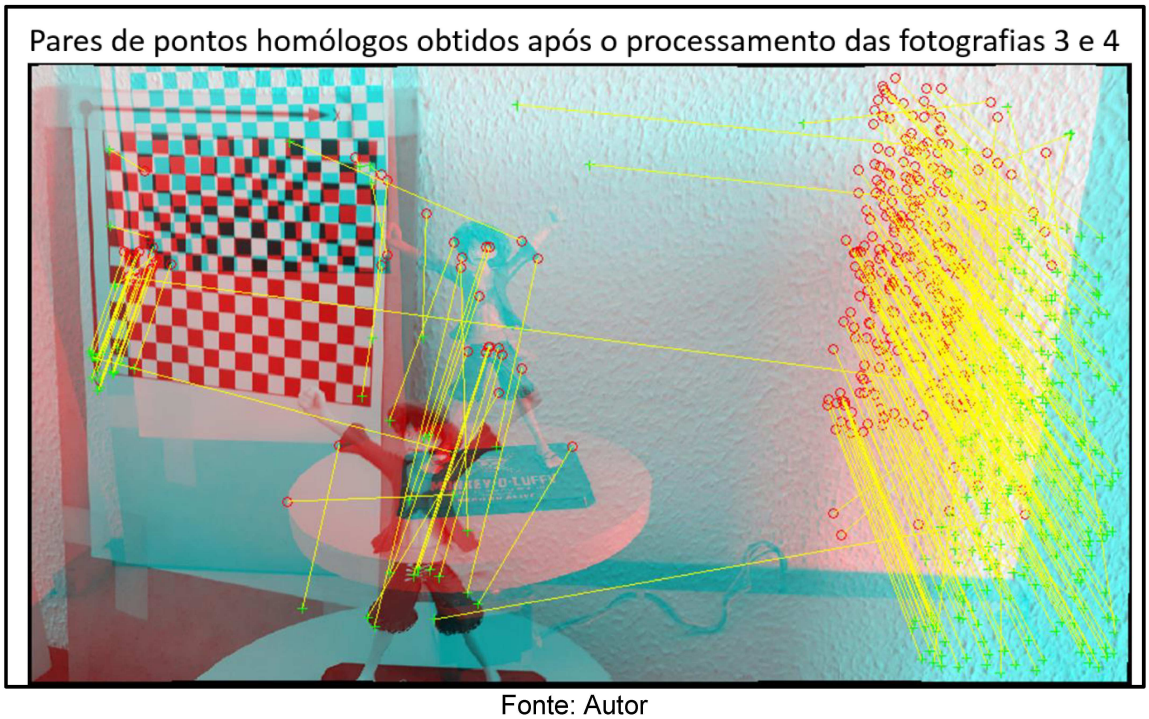
6.4 Aumento da Quantidade de Pontos da Nuvem de Pontos para a Obtenção do 3D Denso

Os resultados apresentados nas Figura 6.3, Figura 6.4, Figura 6.5 e Figura 6.6 foram obtidos utilizando os 8.713 pontos utilizados para encontrar as posições das câmeras. Esse resultado pode ser considerado um 3D esparso. A quantidade de pontos apresentados foi suficiente para calcular as posições das câmeras e, também, de representar, de forma simples, a geometria da cena. Entretanto é possível aumentar a quantidade de pontos homólogos nos pares de fotografias e obter, assim, mais detalhes da cena. O resultado do aumento do número de pontos da nuvem é denominado de 3D denso.

Para aumentar a quantidade de pontos homólogos repetiu-se o procedimento de encontrar pontos homólogos, porém essa busca foi limitada por regiões que estão próximas aos pontos homólogos utilizados para calcular as posições das câmeras. Este procedimento consegue aumentar a quantidade de pontos homólogos pois, ao reduzir a área de análise aumenta-se a certeza da localização dos pontos e, consequentemente, a chance de encontrar os pares de pontos.

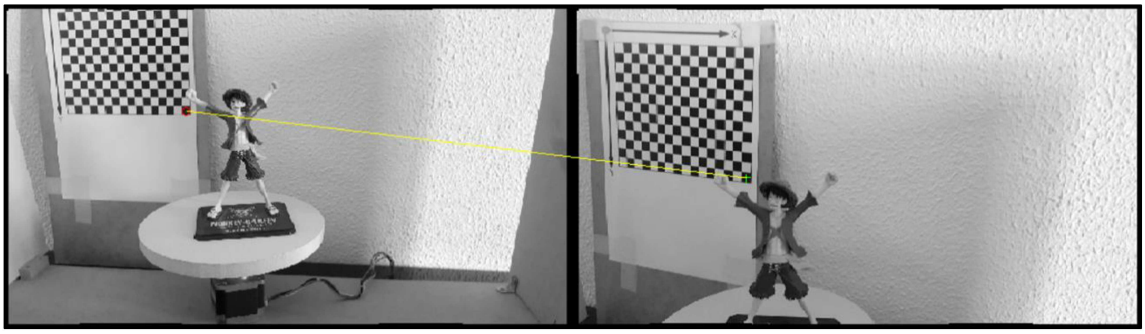
A Figura 6.7 apresenta os 289 pontos homólogos (como apresentado na Tabela 6.1) obtidos ao processar as fotografias 3 e 4 da bancada de experimentos. Ao analisar a Figura 6.7 é possível verificar que existe uma grande concentração de pontos homólogos ao lado direito da imagem enquanto o restante da imagem apresenta uma concentração menor. A Figura 6.8 apresenta um dos pontos homólogos que está localizado em uma região de menor concentração de pontos homólogos. Ao processar uma área de formato quadrado de 200x200 pixeis ao redor dos pontos homólogos da Figura 6.8 obteve-se mais três pontos homólogos, como apresentado na Figura 6.9.

Figura 6.7 – Os 289 pontos homólogos obtido do processamento das fotografias 3 e 4 para o cálculo das posições das câmeras



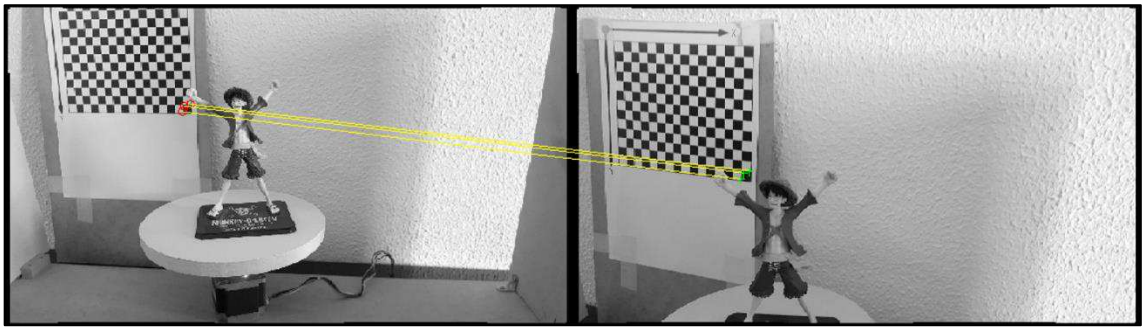
Fonte: Autor

Figura 6.8 – Exemplo de um par de pontos homólogos após o processamento das fotografias 3 e 4



Fonte: Autor

Figura 6.9 – Surgimento de três novos pontos homólogos ao processar uma área de 200x200 pixeis ao redor do ponto apresentado na Figura 6.8



Fonte: Autor

De forma a aumentar o número de pontos homólogos foi criado um algoritmo de crescimento do número de pontos. Este algoritmo avalia uma área de 200 pixeis por 200 pixeis ao redor de cada ponto dos pares de pontos homólogos e, então, o algoritmo pesquisa por pontos característicos nessas áreas e tenta fazer o casamento desses pontos. Caso o algoritmo não encontre pontos homólogos nesta região ele assume que o par que indicou esta região não é um par confiável e, assim, este par é removido da lista de pontos que serão utilizados para gerar o 3D denso.

A Tabela 6.2 apresenta o resultado do algoritmo de crescimento de pontos. Pela Tabela 6.2 é possível notar que em alguns pares de fotografia tiveram redução no número de pontos homólogos, mais o número total de pontos homólogos foi aumentado de 8.713 pontos para 83.791 pontos, ou seja, um aumento de 9,64 vezes. Este aumento indica a capacidade de obter uma nuvem de pontos quase 10 vezes mais densa que a obtida no processo de 3D esparso.

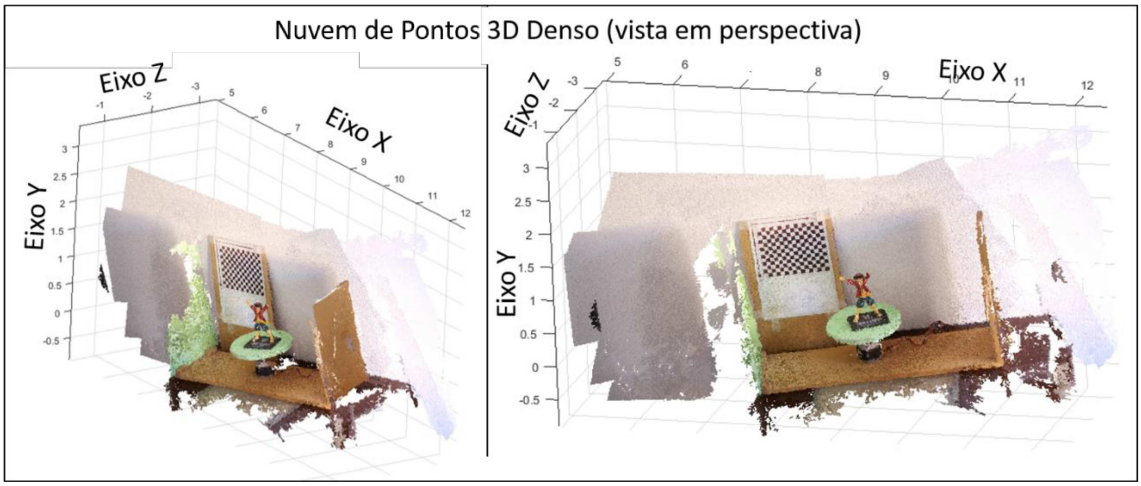
Após o procedimento de crescimento do número de pontos repetiu-se o processo de reconstrução 3D da cena. Para isso utilizou-se as mesmas posições de câmeras já calculadas para o processo de triangulação dos novos pontos. A Figura 6.10 apresenta o resultado obtido após o crescimento do número de pontos através de uma vista em perspectiva. Já as Figura 6.11, Figura 6.12 e Figura 6.13 apresentam, respectivamente, as vistas pelos planos X-Y, planos X-Z e planos Y-Z da nuvem de pontos que representa a bancada de testes. Na Figura 6.13 foi apresentado parte do eixo X pois parte da estrutura obstruiria a visão do boneco e de grande parte da cena.

Tabela 6.2 – Comparação entre o número de pontos homólogos iniciais com o resultado do algoritmo de crescimento de números de pontos

Par de imagem utilizado	Quantidade de pontos homólogos iniciais	Aumento da quantidade de pontos	Par de imagem utilizado	Quantidade de pontos homólogos encontrados	Aumento da quantidade de pontos
01->02	2110	37471	22->23	210	1542
02->03	570	3427	23->24	25	50
03->04	289	1116	24->25	198	3997
04->05	153	302	25->26	23	8
05->06	216	833	26->27	17	13
06->07	494	4455	27->28	194	1078
07->08	309	3066	28->29	194	657
08->09	151	570	29->30	113	754
09->10	114	337	30->31	144	630
10->11	19	16	31->32	29	52
11->12	968	8985	32->33	28	16
12->13	22	19	33->34	63	90
13->14	279	1492	34->35	75	19
14->15	142	442	35->36	33	44
15->16	102	315	36->37	133	463
16->17	28	24	37->38	406	1777
17->18	590	9118	38->39	26	15
18->19	87	169	39->40	10	--------
19->20	71	336	40->41	4	--------
20->21	14	25	41->42	4	--------
21->22	56	68	Total de Pontos utilizados		83.791

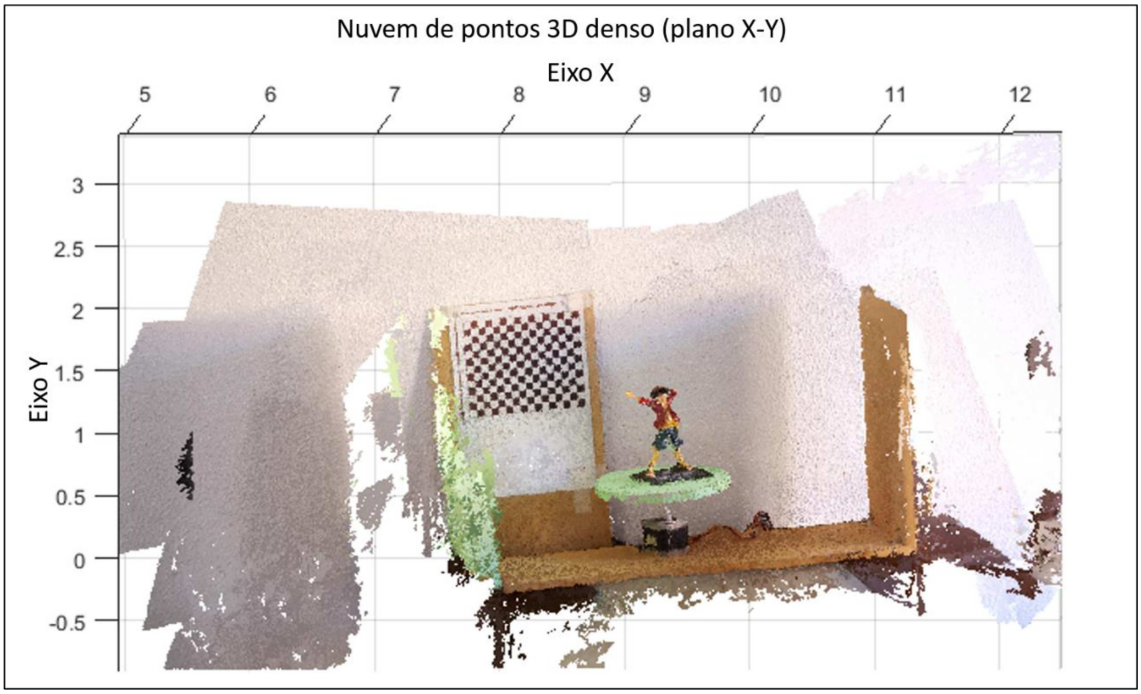
Fonte: Autor

Figura 6.10 – Nuvem de pontos 3D denso representando a bancada de testes com 83.791 pontos



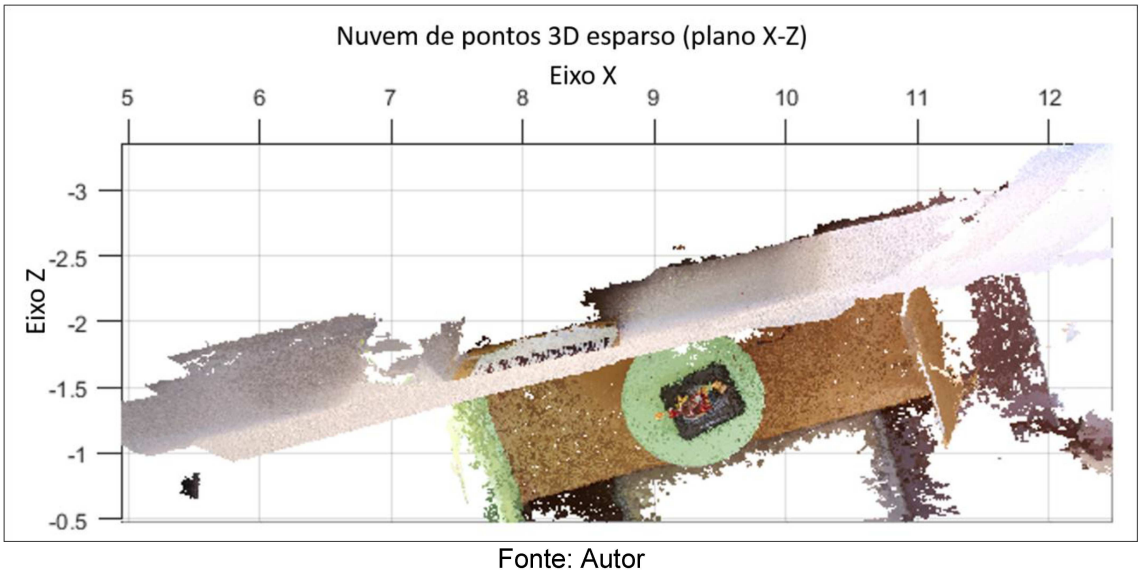
Fonte: Autor

Figura 6.11 – Nuvem de pontos 3D denso representando a bancada de testes por uma vista no plano X-Y



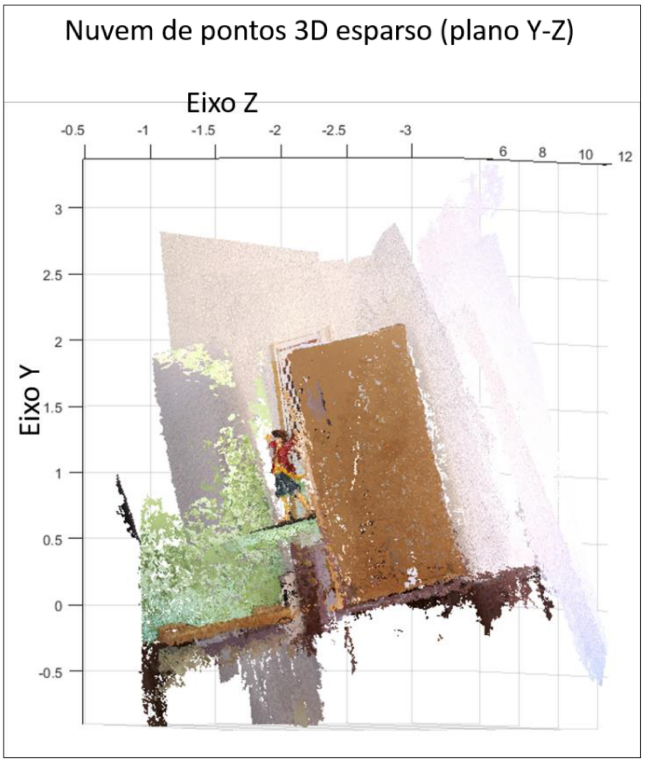
Fonte: Autor

Figura 6.12 – Nuvem de pontos 3D denso representando a bancada de testes por uma vista no plano X-Z



Fonte: Autor

Figura 6.13 – Nuvem de pontos 3D denso representando a bancada de testes por uma vista no plano Y-Z



Fonte: Autor

6.5 Validação dos Resultados Obtidos

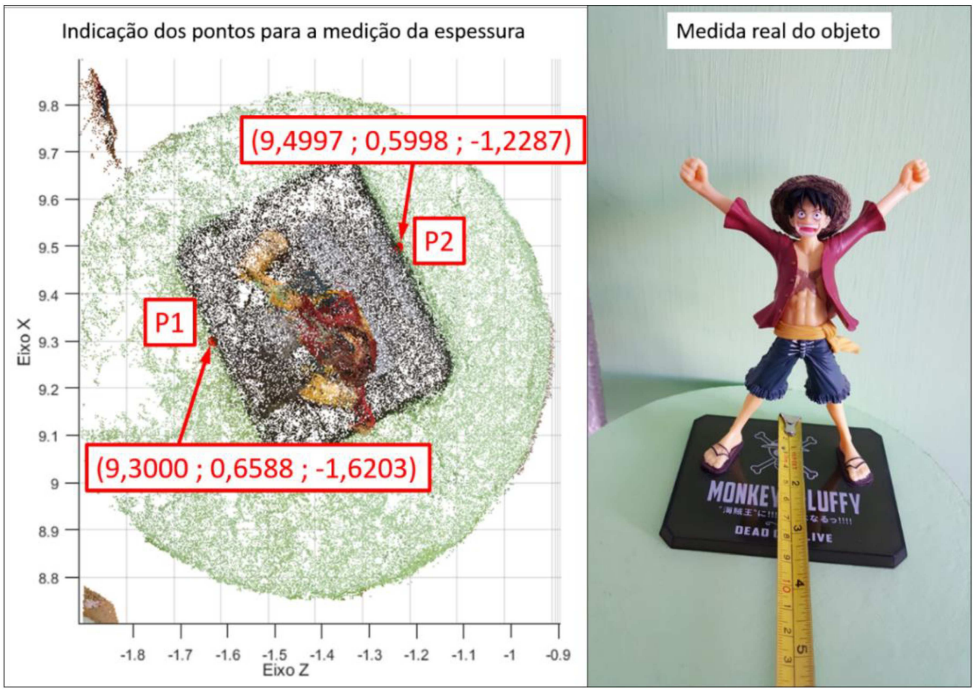
Para validar os resultados tentou-se relacionar as medidas reais do objeto com os pontos obtidos da nuvem de pontos 3D denso. Para isso buscou-se na nuvem de pontos do 3D denso pontos que possam representar a espessura, largura e altura da representação do boneco na nuvem.

A Figura 6.14 apresenta o ponto P1 e o ponto P2 escolhidos na nuvem para a medição da espessura. De acordo com a Figura 6.14 temos que o ponto P1 possui a coordenada (9,3000 ; 0,6588 ; -1,6203) e o ponto P2 possui a coordenada (9,4997 ; 0,5998 ; -1,2287), sendo a distância calculada entre os pontos P1 e P2 de 0,4435. Ainda na Figura 6.14 é possível verificar que a espessura da base do boneco é de 8,5 cm e, assim, tem-se que a espessura do objeto real é 19,1648 vezes maior do que a espessura do objeto em 3D.

Na Figura 6.15 é apresenta o ponto P3 e o ponto P4 escolhidos na nuvem para a medição da largura. De acordo com a Figura 6.15 temos que o ponto P3 possui a coordenada (9,6593 ; 0,6487 ; -1,4295) e o ponto P4 possui a coordenada (9,1683 ; 0,5682 ; -1,1898), sendo a distância calculada entre os pontos P3 e P4 de

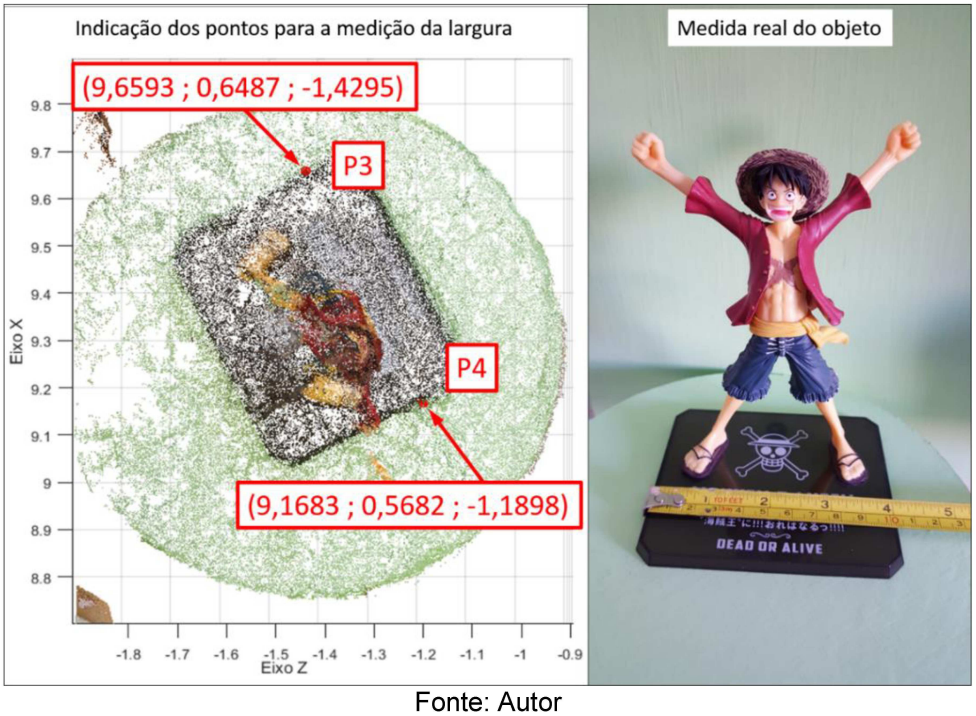
0,5523. Ainda na Figura 6.15 é possível verificar que a largura da base do boneco é de 10,5 cm e, assim, tem-se que a largura do objeto real é 19,0120 vezes maior do que a largura do objeto em 3D.

Figura 6.14 – Nuvem de pontos do 3D denso com as coordenadas dos pontos utilizados para a medição da espessura na esquerda e o objeto real com a medição real da espessura



Fonte: Autor

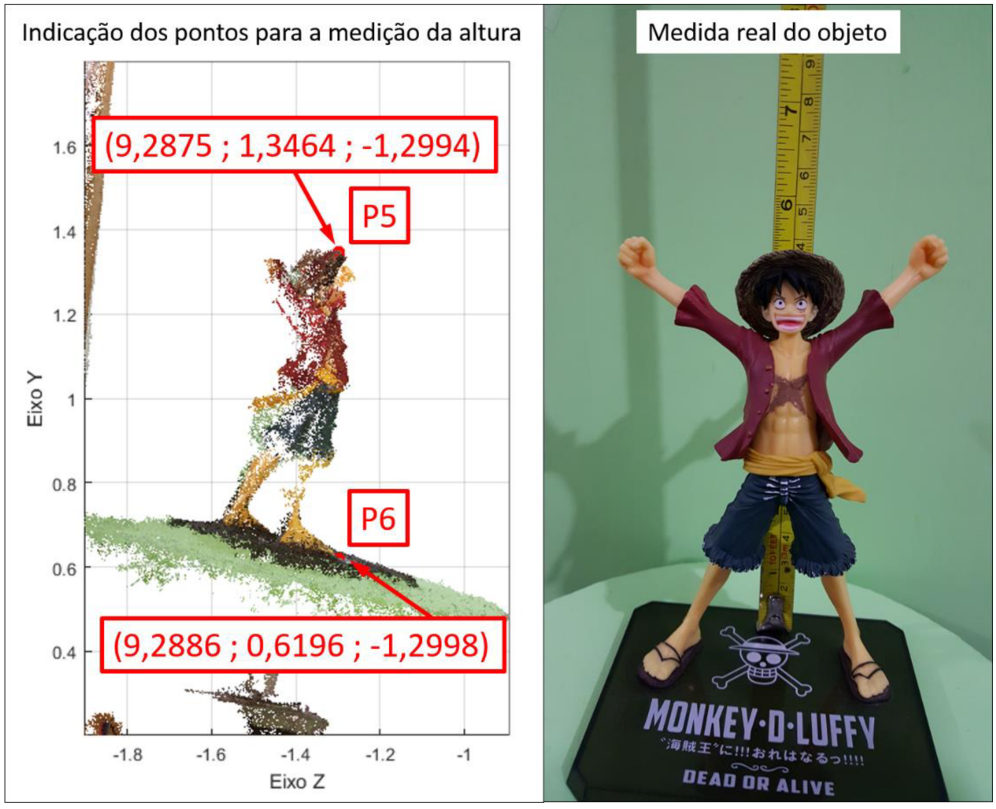
Figura 6.15 – Nuvem de pontos do 3D denso com as coordenadas dos pontos utilizados para a medição da largura na esquerda e o objeto real com a medição real da largura



Fonte: Autor

Já a Figura 6.16 apresenta o ponto P5 e o ponto P6 escolhidos na nuvem para a medição da altura do objeto. De acordo com a Figura 6.16 temos que o ponto P5 possui a coordenada (9,2875 ; 1,3464 ; -1,2994) e o ponto P6 possui a coordenada (9,2886 ; 0,6196 ; -1,2998), sendo a distância calculada entre os pontos P5 e P6 é de 0,7268. Ainda na Figura 6.16 é possível verificar que a altura da base até o ponto mais alto do objeto é de 14,0 cm e, assim, tem-se que a altura do objeto real é 19,2625 vezes maior do que a altura do objeto em 3D.

Figura 6.16 – Nuvem de pontos do 3D denso com as coordenadas dos pontos utilizados para a medição da altura na esquerda e o objeto real com a medição real da altura



Fonte: Autor

Como demonstrado pelas Figura 6.14, Figura 6.15 e Figura 6.16 o objeto modelado em 3D apresenta as mesmas proporções do objeto real, porém em uma escala menor sendo, em média, 19,1359 vezes menor. Como apresentado na subseção 2.6.2.1 a reconstrução da cena em 3D só é possível ser realizada em escala, pois existem incertezas na decomposição da matriz essencial em rotação e em translação. Sendo, assim, pode-se dizer que a metodologia proposta neste capítulo foi capaz de modelar em 3D da bancada de testes e um objeto sobre ela.

6.6 Considerações Finais

Neste capítulo foi demonstrado como realizar a reconstrução 3D de uma cena em que não se tem conhecimento *a priori* das posições de câmera. Para demonstrar o procedimento foi utilizada a mesma bancada experimental apresentada no Capítulo 4, porém, neste capítulo, as fotografias foram obtidas de várias posições diferentes de forma a não se ter um conhecimento *a priori* das posições de câmera.

O próximo capítulo demonstra como realizar a reconstrução 3D de áreas urbanas através das metodologias descritas até o momento.

Capítulo 7

7 RECONSTRUÇÃO 3D DE ÁREAS URBANAS ATRAVÉS DE IMAGENS OBTIDAS POR CÂMERAS COM MOVIMENTOS NÃO DEFINIDOS

No Capítulo 6 foi demonstrado, utilizando a bancada de teste, uma metodologia capaz de fazer a reconstrução 3D de uma cena através de fotografias que foram tiradas de diversas posições e sem que tivesse o conhecimento *a priori* dessas posições. Esta metodologia apresentou-se eficiente para a reconstrução 3D da bancada de teste.

Devido ao resultado obtido, esta metodologia foi aplicada a uma cena de área urbana que contém uma edificação para avaliar o resultado da reconstrução 3D de uma edificação.

7.1 Introdução

Como apresentado no Capítulo 6, a metodologia para a reconstrução 3D de cenas através de fotografias deve seguir seis etapas quando não se conhece as posições das câmeras. Essas etapas são:

Etapa 1 - Obter os parâmetros intrínsecos da câmera (calibração da câmera);
Etapa 2 - Obtenção dos pontos característicos da imagem;
Etapa 3 - Casamento dos pontos característicos;
Etapa 4 - Ordenar a sequência das imagens;
Etapa 5 - Calcular, pelos pares das imagens, os parâmetros extrínsecos de cada posição de câmera;
Etapa 6 - Triangulação para a obtenção do 3D.

Para o processo de reconstrução 3D de uma área urbana por fotografias foi utilizado a mesma câmera do processo que permitiu a reconstrução 3D da bancada destes. Isso permitiu que fosse utilizada a mesma calibração. Assim, para esta etapa a calibração da câmera já é conhecida.

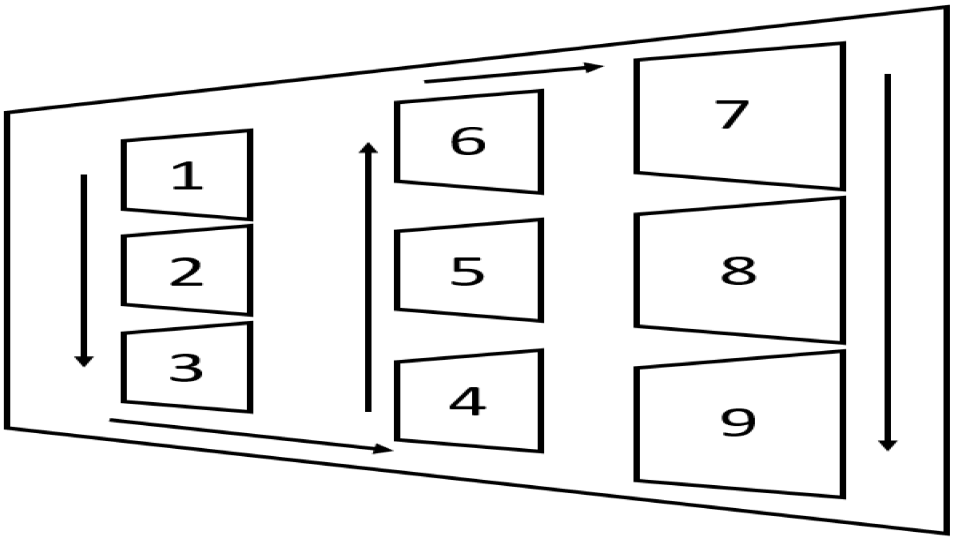
Optou-se por utilizar esses parâmetros intrínsecos (calibração da câmera) para não ter que adicionar à cena urbana o padrão de calibração, pois este padrão poderia obstruir detalhes da cena. Isso foi possível pois utilizou-se a mesma câmera nas duas situações e a câmera fora configurada para utilizar o mesmo foco nas duas situações.

7.2 Obtenção do Banco de Imagens de uma Área Urbana e Obtenção dos Pontos Homólogos

Quando foram realizados os testes do Capítulo 6 não se preocupou nas posições e na sequência em que as fotos foram tiradas. Assim, para encontrar os pares de imagem com mais pontos homólogos, a fim de permitir o cálculo das posições de câmera, foi feita a combinação de todas as imagens até obter os pares com o maior número de pontos homólogos. Apesar de ter apresentado resultados satisfatórios esse procedimento tem alto custo computacional, já que ele precisa processar todos as combinações possíveis de pares.

Com o intuito de reduzir a quantidade de cálculos que o computador precisa fazer tomou-se cuidado ao criar o banco de fotografias para a reconstrução 3D de uma área urbana. Para evitar o excesso de cálculos as fotografias do banco de dados da cena urbana foram tiradas seguindo uma movimentação que lembra a letra S inclinado a 90°, como apresentado na Figura 7.1.

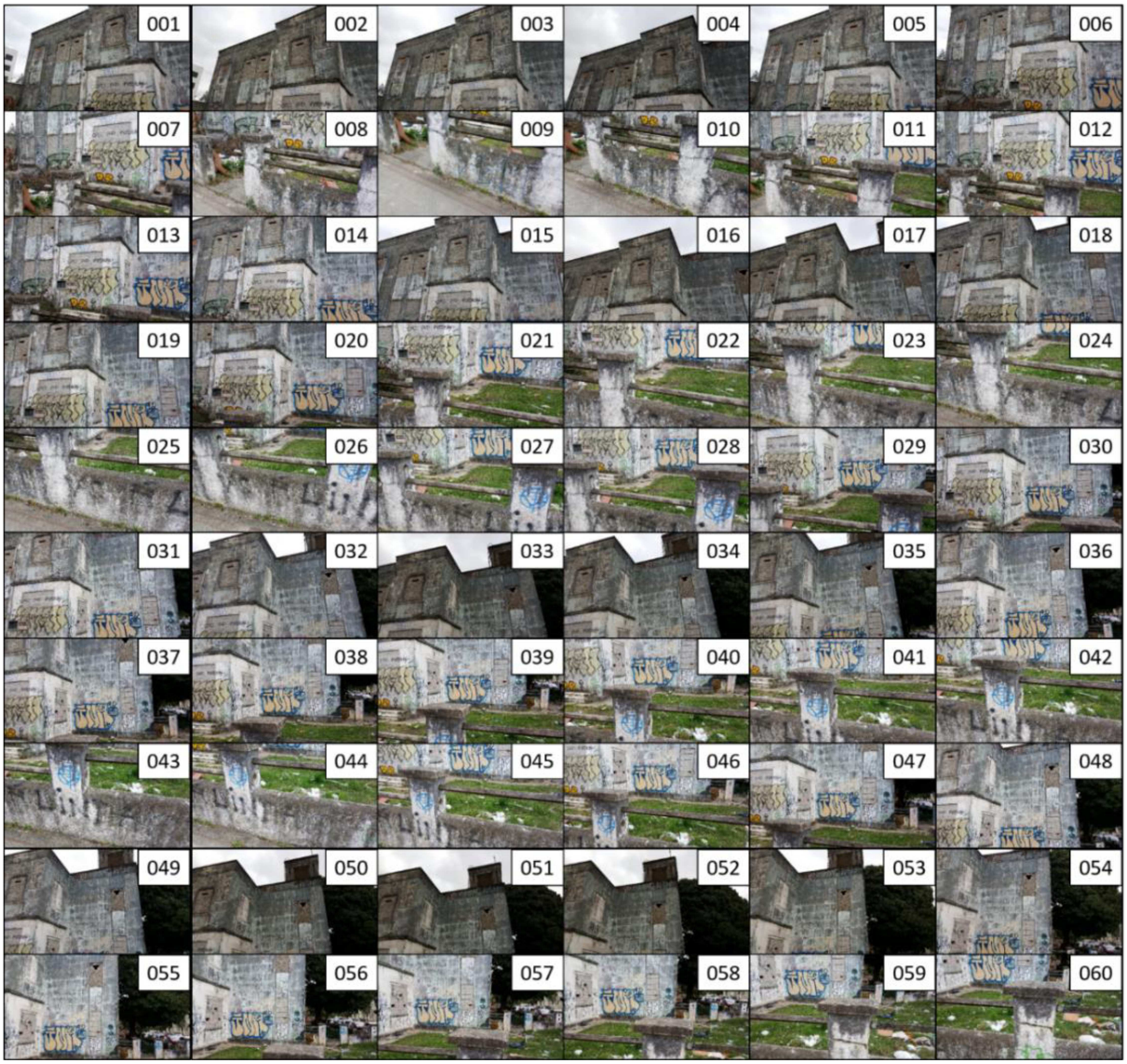
Figura 7.1 – Sequência utilizada para criar o banco de imagens para a reconstrução 3D de áreas urbanas



Fonte: Autor

Vale ressaltar que, apesar de ter sido utilizado uma ordem determina para a criação do banco de imagens, as posições de câmera não são conhecidas, porém pode-se afirmar que os pares de imagens a serem avaliados são conhecidos e, assim, não é necessário processar todo o banco de imagens a fim de encontrar os pares de imagens com mais pontos homólogos, sendo os pares utilizados obtidos a partir da sequência em que foram tiradas as fotografias. A Figura 7.2 apresenta as 60 primeiras imagens das 189 imagens do banco de imagens criado para reconstrução 3D de áreas urbanas através de imagens.

Figura 7.2 – Parte do banco de imagens utilizadas nos testes de reconstrução 3D de áreas urbanas



Fonte: Autor

Em posse do banco de imagem e conhecido a sequência em que as fotos foram tiradas, encontrou-se os pontos característicos de cada imagem e, então, foi feito o casamento de pontos entre os pares de imagens (pares conhecidos *a priori* pela sequência que foram tomadas as fotografias) a fim de encontrar os pontos homólogos que serão utilizados para o cálculo das posições de câmera e para a reconstrução 3D da cena. A quantidade de pontos homólogos encontrados em cada par de imagem é apresentada na Tabela 7.1.

Tabela 7.1 – Número de pontos homólogos encontrados para cada par de imagem utilizado na reconstrução 3D de uma cena urbana

(Continua)

Par de imagem	Pontos homólogos	Par de imagem	Pontos homólogos	Par de imagem	Pontos homólogos
1->2	6.661	37->38	10.967	73->74	4.292
2->3	7.294	38->39	11.579	74->75	8.310
3->4	5.081	39->40	10.761	75->76	7.578
4->5	4.366	40->41	12.043	76->77	6.684
5->6	8.091	41->42	14.406	77->78	3.436
6->7	7.350	42->43	11.404	78->79	2.066
7->8	6.922	43->44	4.699	79->80	3.704
8->9	1.007	44->45	10.664	80->81	661
9->10	555	45->46	7.188	81->82	7.514
10->11	5.588	46->47	5.636	82->83	4.005
11->12	11.414	47->48	4.739	83->84	4.064
12->13	10.041	48->49	5.557	84->85	7.325
13->14	11.134	49->50	4.187	85->86	6.431
14->15	9.221	50->51	3.520	86->87	5.384
15->16	6.820	51->52	2.469	87->88	2.357
16->17	5.963	52->53	3.704	88->89	1.991
17->18	7.314	53->54	5.092	89->90	3.005
18->19	10.288	54->55	6.463	90->91	4.884
19->20	11.612	55->56	7.647	91->92	4.541
20->21	6.753	56->57	7.429	92->93	7.950
21->22	12.562	57->58	7.550	93->94	4.856
22->23	13.269	58->59	6.229	94->95	6.272
23->24	15.214	59->60	6.755	95->96	6.008
24->25	16.539	60->61	8.653	96->97	3.626
25->26	3.589	61->62	7.514	97->98	6.491
26->27	10.405	62->63	11.780	98->99	6.718
27->28	13.275	63->64	2.137	99->100	5.179
28->29	9.543	64->65	6.180	100->101	7.825
29->30	9.396	65->66	5.939	101->102	5.963
30->31	10.184	66->67	5.903	102->103	4.569
31->32	8.033	67->68	7.263	103->104	2.414
32->33	5.926	68->69	6.946	104->105	1.901
33->34	4.907	69->70	7.776	105->106	2.785
34->35	8.222	70->71	4.734	106->107	5.533
35->36	8.323	71->72	2.691	107->108	5.244
36->37	10.955	72->73	3.758	108->109	8.165

Fonte: Autor

Tabela 7.1 – Número de pontos homólogos encontrados para cada par de imagem utilizado na reconstrução 3D de uma cena urbana

(Conclusão)

Par de imagem	Pontos homólogos	Par de imagem	Pontos homólogos	Par de imagem	Pontos homólogos
109->110	7.304	138->139	1.431	164->165	4.791
110->111	5.970	139->140	887	165->166	7.263
111->112	6.602	136->137	3.097	166->167	6.058
112->113	11.050	137->138	2.629	167->168	2.931
113->114	8.059	140->141	1.660	168->169	766
114->115	2.282	141->142	1.698	169->170	6.474
115->116	7.469	142->143	1.612	170->171	5.189
116->117	7.660	143->144	4.727	171->172	5.053
117->118	5.650	145->146	5.276	172->173	3.535
118->119	4.788	146->147	4.120	173->174	5.025
119->120	6.559	147->148	6.639	174->175	3.754
120->121	3.459	148->149	9.543	175->176	1.952
121->122	1.841	149->150	2.969	176->177	1.402
122->123	1.093	150->151	6.080	177->178	189
123->124	1.489	151->152	6.872	178->179	3.401
124->125	4.151	152->153	1.562	179->180	3.056
125->126	4.730	153->154	2.947	180->181	7.858
126->127	2.632	154->155	3.002	181->182	6.157
127->128	3.825	155->156	1.946	182->183	5.393
128->129	4.935	156->157	1.988	183->184	6.189
129->130	761	157->158	1.179	184->185	6.082
130->131	7.327	158->159	664	185->186	6.379
131->132	7.790	159->160	1.691	186->187	7.856
132->133	4.944	160->161	1.667	187->188	7.304
133->134	5.835	161->162	2.331	188->189	6.609
134->135	4.789	162->163	1.730	Total de números homólogos	
135->136	5.959	163->164	4.688	1.084.419	

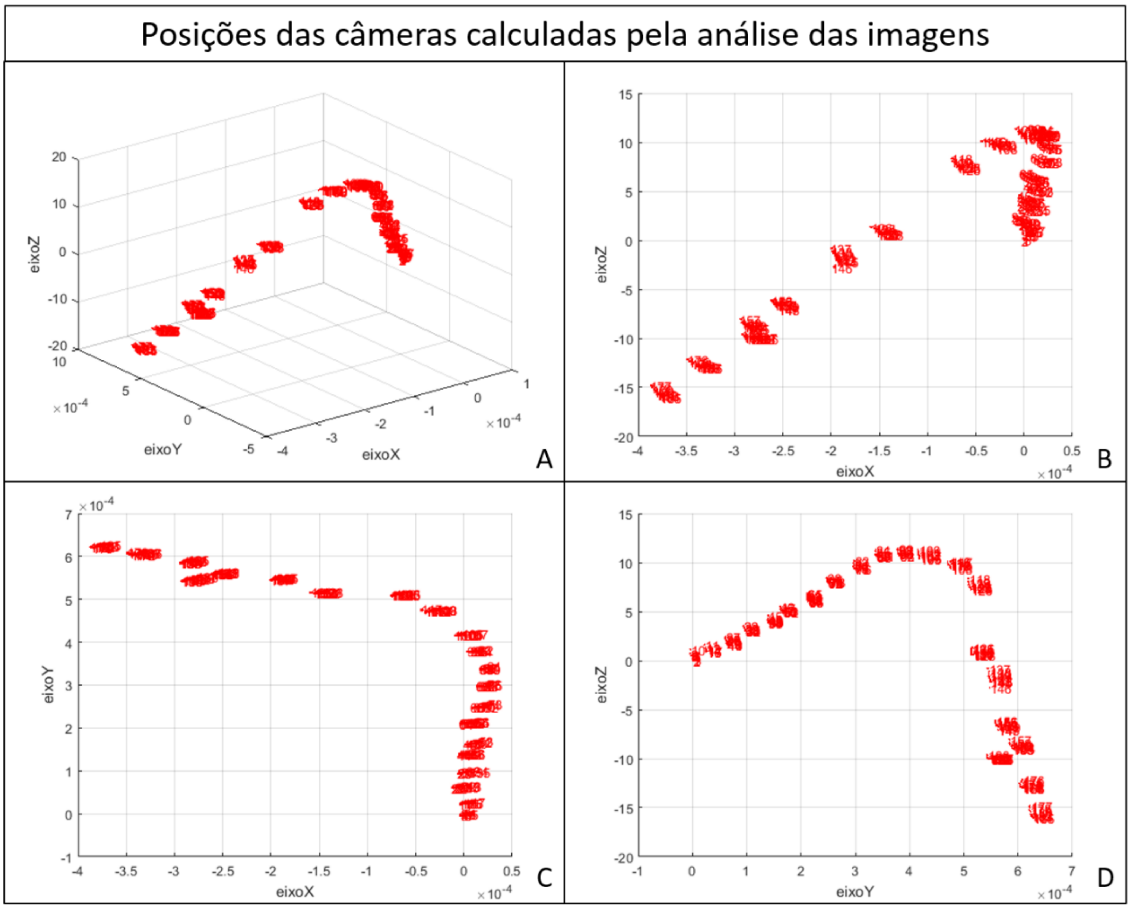
Fonte: Autor

7.3 Cálculo das Posições das Câmeras e Obtenção do 3D Esparso

O banco de imagens, que parte dele é apresentado na Figura 7.2, é constituído de fotografias do casarão da antiga Capitania dos Portos na cidade de Pelotas/RS, um prédio histórico da cidade. Este casarão está localizado na esquina da rua Benjamin Constant com a rua Álvaro Chaves.

A Figura 7.3 apresenta as posições de câmera calculadas após a análise das fotografias desse casarão. É interessante visualizar na Figura 7.3 que ao calcular as posições de câmera foi possível detectar a movimentação da câmera pela esquina na qual se localiza este casarão.

Figura 7.3 – Representação das posições da câmera calculadas pela análise das imagens da cena urbana

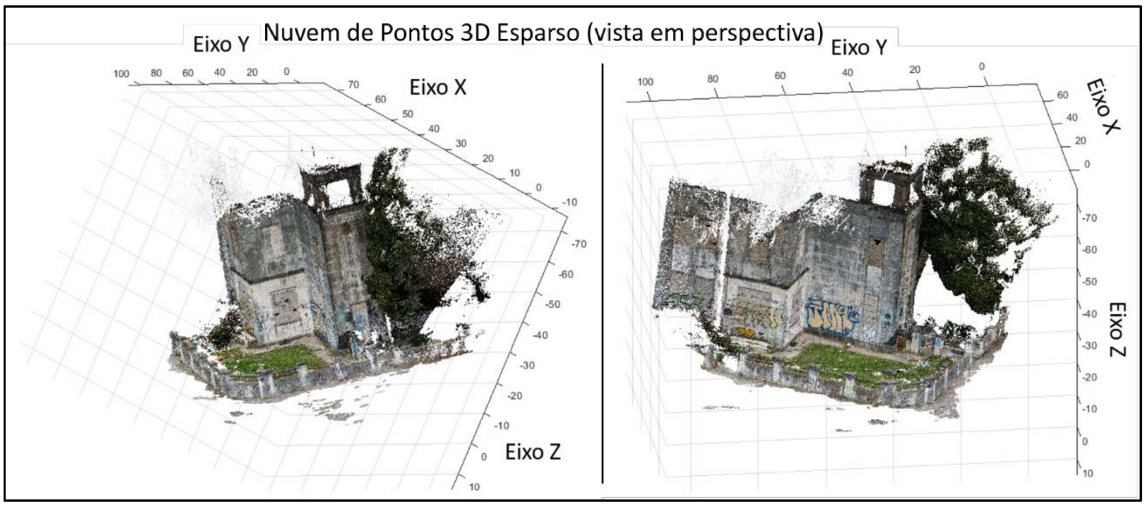


A) vista em perspectiva das posições; B) vista do plano X-Z; C) Vista do plano X-Y; D) Vista do plano Y-Z
Fonte: Autor

Após obter os diversos pontos característicos, ter encontrado os pontos homólogos nos pares de imagem e, por fim, calculado as posições das câmeras realizou-se o processo de triangulação, utilizando os pontos homólogos e as posições de câmeras calculadas.

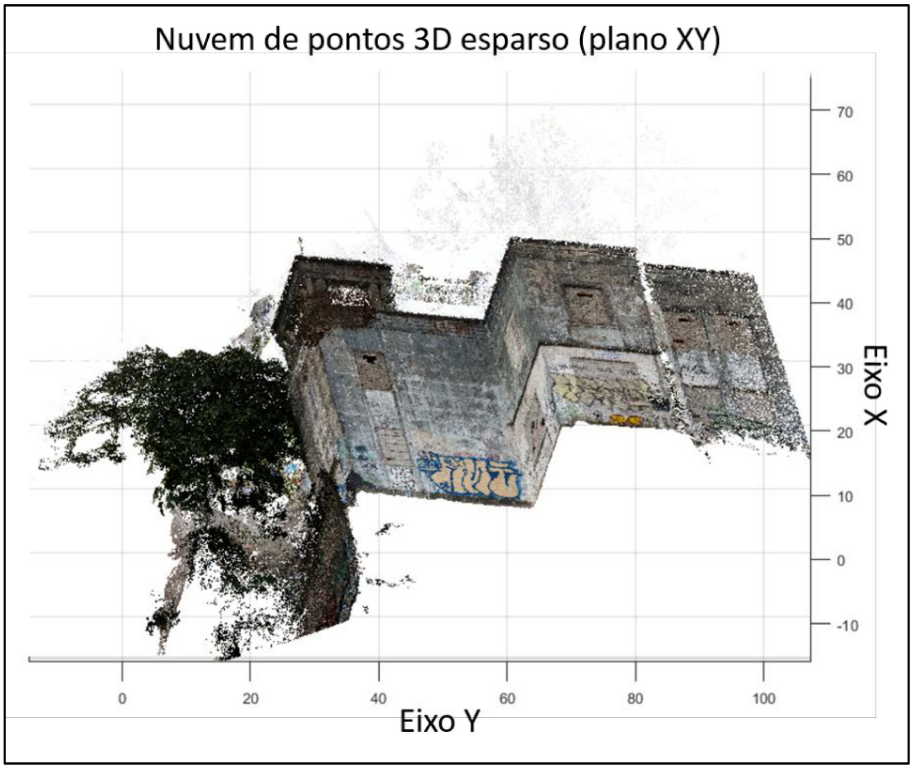
Como resultado obteve-se a nuvem de pontos 3D apresentada na Figura 7.4. Uma visão mais clara da nuvem de pontos é fornecida pela Figura 7.5 que apresenta a nuvem de pontos detalhando o plano X-Y; pela Figura 7.6 na qual é possível visualizar a nuvem de pontos pelo plano X-Z; e pela Figura 7.7 na qual é possível visualizar na nuvem de pontos tendo como referência o plano Y-Z.

Figura 7.4 – Nuvem de pontos 3D esparso representando uma área urbana com 1.084.419 pontos



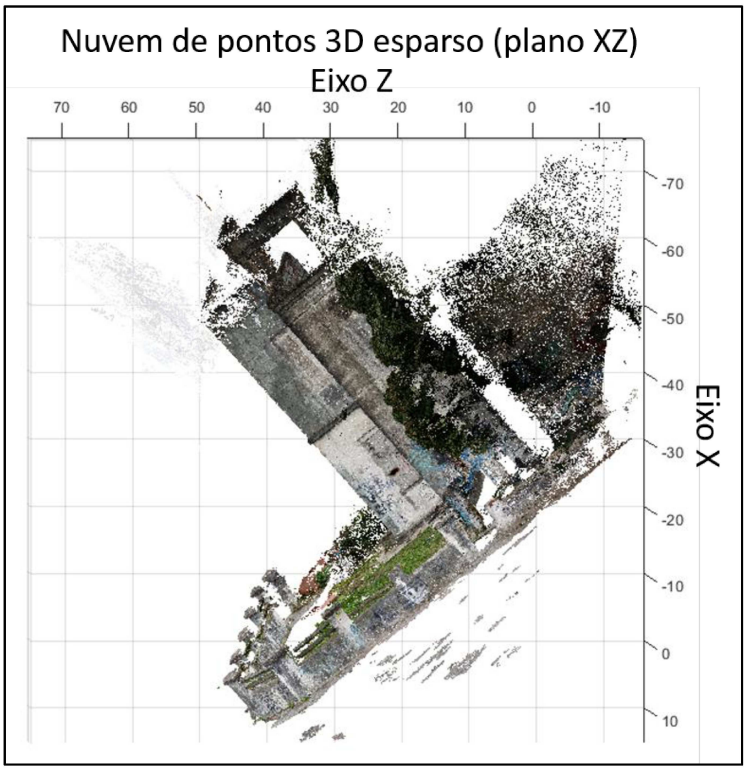
Fonte: Autor

Figura 7.5 – Nuvem de pontos 3D esparso representando uma área urbana vista no plano X-Y



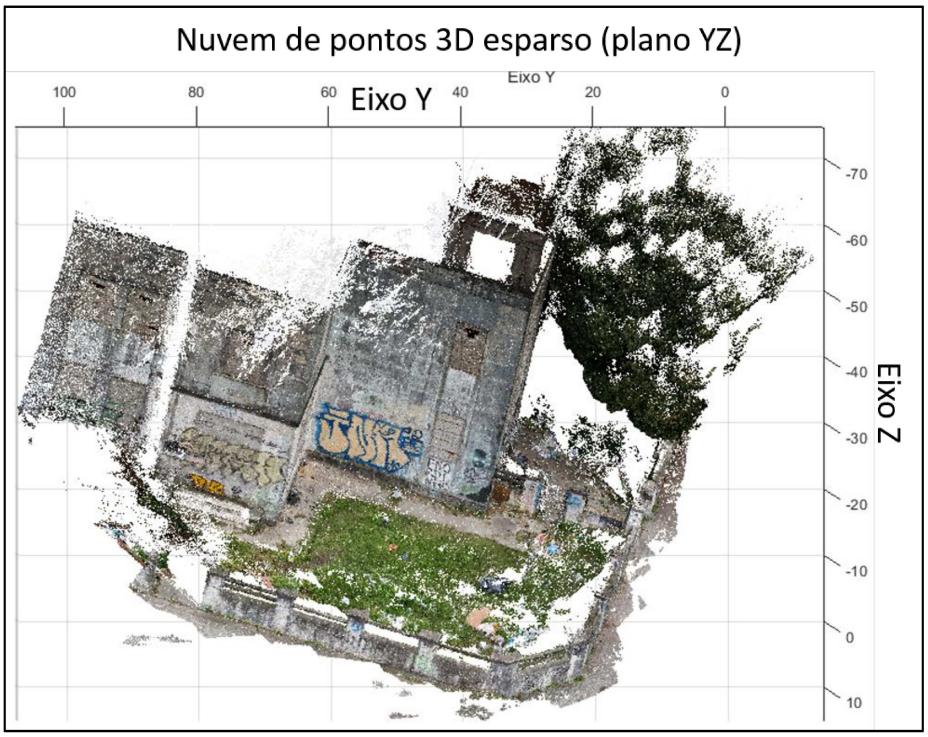
Fonte: Autor

Figura 7.6 – Nuvem de pontos 3D esparso representando uma área urbana vista no plano X-Z



Fonte: Autor

Figura 7.7 – Nuvem de pontos 3D esparso representando uma área urbana vista no plano Y-Z



Fonte: Autor

As direções de alguns eixos das figuras foram invertidas para possibilitar uma melhor visão da estrutura a ser representada.

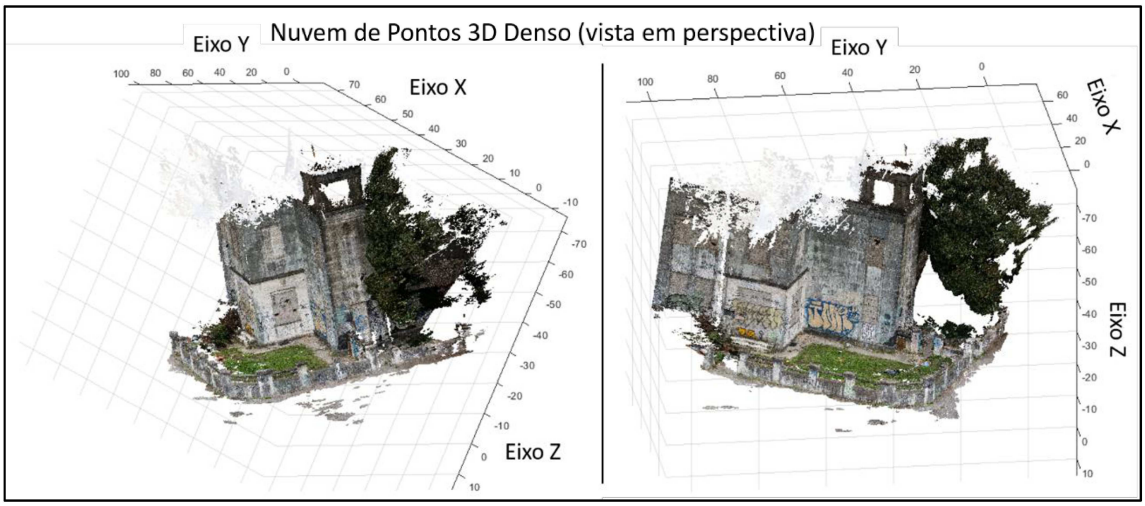
Como pode ser observado nas Figura 7.4, Figura 7.5, Figura 7.6 e Figura 7.7 a metodologia utilizada no Capítulo 6 também conseguiu construir, em 3D, uma áreas urbanas apenas utilizando imagens obtidas por câmeras em que a posição em que cada foto fora tomada não é representada por uma função.

7.4 Crescimento da Quantidade de Pontos e Geração do 3D Denso

Assim como descrito no Capítulo 6 é possível aumentar a quantidade de pontos da nuvem de pontos que representa o casarão da Capitania dos Portos de Pelotas ao limitar áreas de interesse próximas aos pontos em que já foram encontrados os pontos homólogos e, com isso, obter uma nuvem 3D denso.

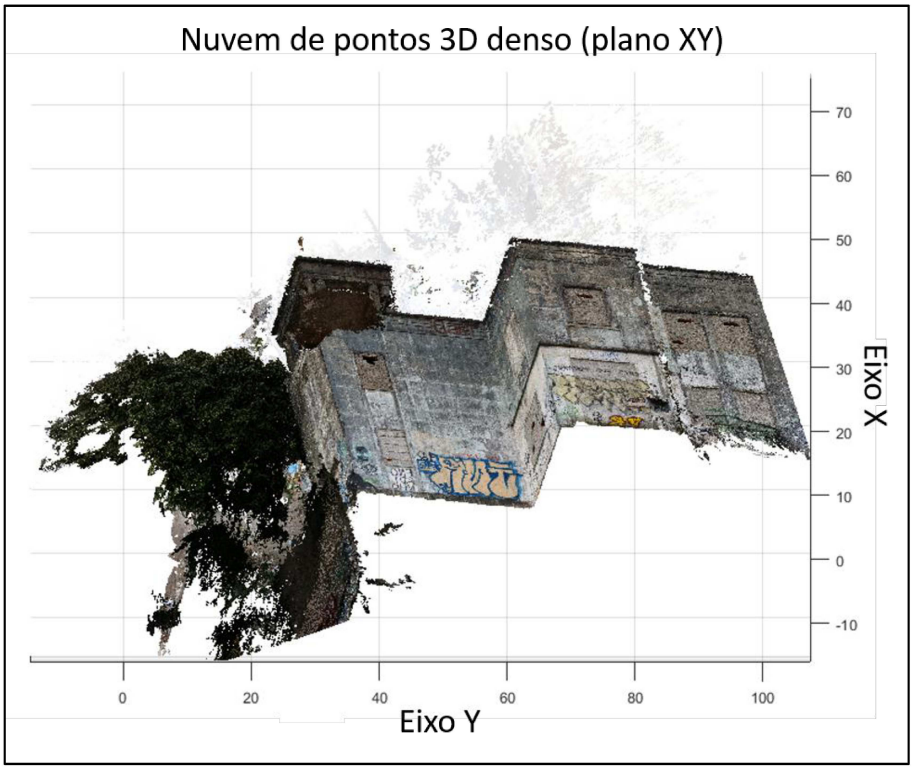
Seguindo a metodologia de crescimento de pontos descrita no Capítulo 6 foi obtido uma nuvem de pontos composta por 9.001.946 pontos que representa a reconstrução 3D da Capitania dos Portos da cidade de Pelotas/RS. A Figura 7.8 apresenta duas vistas em perspectiva da nuvem de pontos do 3D denso da edificação. Já a Figura 7.9 apresenta uma vista no plano X-Y da nuvem de pontos, enquanto que na Figura 7.10 é possível visualizar o resultado do crescimento de pontos pelo plano X-Z e, por fim, a Figura 7.11 apresenta a nuvem de pontos do 3D denso pelo plano Y-Z. O resultado da metodologia de crescimento do número pontos para a obtenção do 3D denso é apresentado na Tabela 7.2.

Figura 7.8 – Nuvem de pontos 3D denso representando uma área urbana com 9.001.946 pontos



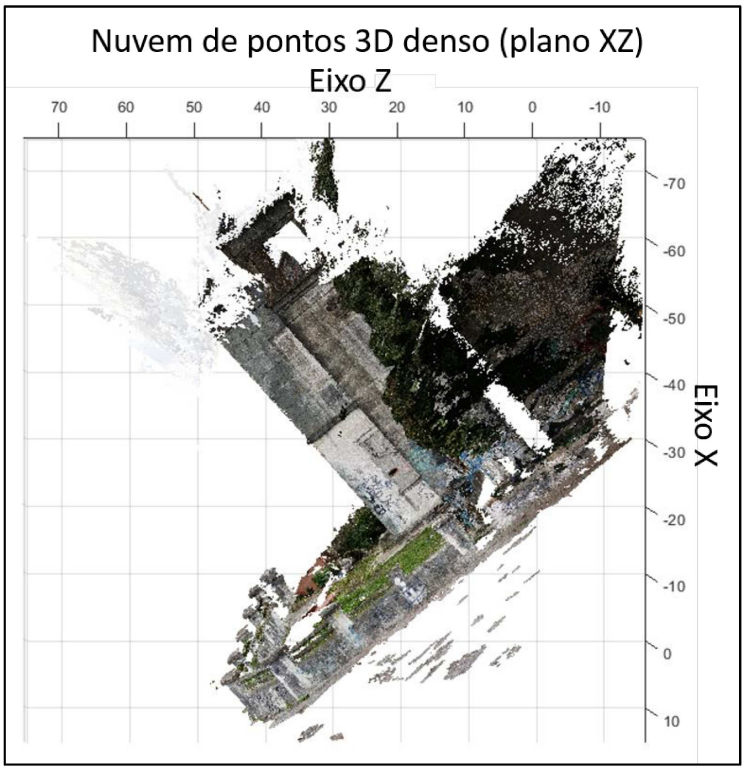
Fonte: Autor

Figura 7.9 – Nuvem de pontos 3D denso representando uma área urbana vista no plano X-Y



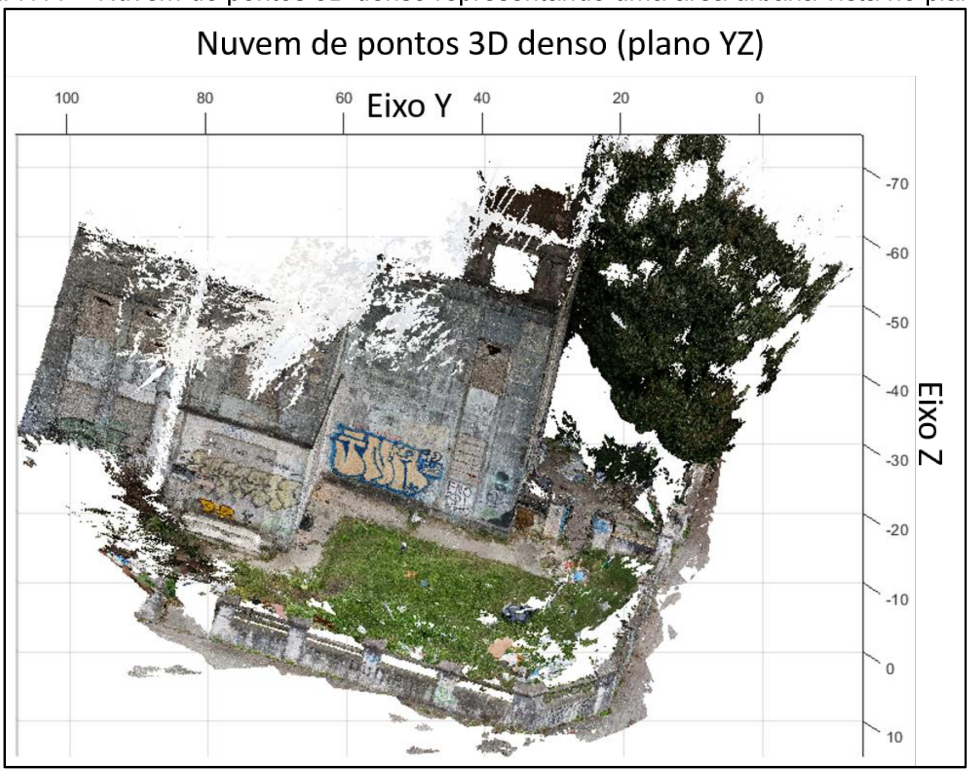
Fonte: Autor

Figura 7.10 – Nuvem de pontos 3D denso representando uma área urbana vista no plano X-Z



Fonte: Autor

Figura 7.11 – Nuvem de pontos 3D denso representando uma área urbana vista no plano Y-Z



Fonte: Autor

Novamente, inverteu-se as direções de alguns eixos das figuras acima para possibilitar uma melhor visão da estrutura a ser representada.

Tabela 7.2 – Número de pontos homólogos encontrados após a metodologia de crescimento do número de pontos na reconstrução 3D de uma cena urbana

(Continua)

Par de imagem	Pontos homólogos	Par de imagem	Pontos homólogos	Par de imagem	Pontos homólogos
1->2	55293	45->46	59669	89->90	24943
2->3	60545	46->47	46787	90->91	40539
3->4	42179	47->48	39343	91->92	37699
4->5	36243	48->49	46127	92->93	65994
5->6	67159	49->50	34751	93->94	40305
6->7	61011	50->51	29221	94->95	52060
7->8	57456	51->52	20498	95->96	49873
8->9	8357	52->53	30744	96->97	30098
9->10	4610	53->54	42269	97->98	53884
10->11	46389	54->55	53653	98->99	55770
11->12	94744	55->56	63481	99->100	42991
12->13	83353	56->57	61669	100->101	64960
13->14	92428	57->58	62670	101->102	49503
14->15	76545	58->59	51710	102->103	37927
15->16	56617	59->60	56075	103->104	20042
16->17	49496	60->61	71829	104->105	15777
17->18	60718	61->62	62372	105->106	23121
18->19	85402	62->63	97788	106->107	45927
19->20	96389	63->64	17735	107->108	43531
20->21	56054	64->65	51303	108->109	67776
21->22	104280	65->66	49301	109->110	60627
22->23	110143	66->67	49003	110->111	49555
23->24	126294	67->68	60289	111->112	54804
24->25	137289	68->69	57657	112->113	91729
25->26	29792	69->70	64546	113->114	66899
26->27	86377	70->71	39296	114->115	18939
27->28	110201	71->72	22335	115->116	61999
28->29	79215	72->73	31193	116->117	63589
29->30	77999	73->74	35629	117->118	46905
30->31	84536	74->75	68983	118->119	39746
31->32	66687	75->76	62907	119->120	54443
32->33	49192	76->77	55484	120->121	28715
33->34	40734	77->78	28518	121->122	15285
34->35	68251	78->79	17153	122->123	9069
35->36	69090	79->80	30748	123->124	12362
36->37	90941	80->81	5482	124->125	34460
37->38	91041	81->82	62378	125->126	39265
38->39	96114	82->83	33250	126->127	21850
39->40	89330	83->84	33738	127->128	31748
40->41	99974	84->85	60806	128->129	40968
41->42	119589	85->86	53385	129->130	6317
42->43	94664	86->87	44689	130->131	60825
43->44	39004	87->88	19568	131->132	64668
44->45	88523	88->89	16527	132->133	41042

Fonte: Autor

Tabela 7.2 – Número de pontos homólogos encontrados após a metodologia de crescimento do número de pontos na reconstrução 3D de uma cena urbana

(Conclusão)

Par de imagem	Pontos homólogos	Par de imagem	Pontos homólogos	Par de imagem	Pontos homólogos
133->134	48435	153->154	24464	172->173	29343
134->135	39756	154->155	24923	173->174	41710
135->136	49464	155->156	16158	174->175	31161
136->137	25703	156->157	16506	175->176	16207
137->138	21826	157->158	9785	176->177	11633
138->139	11878	158->159	5515	177->178	1569
139->140	7362	159->160	14034	178->179	28230
140->141	13781	160->161	13841	179->180	25367
141->142	14090	161->162	19354	180->181	65230
142->143	13382	162->163	14358	181->182	51110
143->144	39241	163->164	38910	182->183	44771
145->146	38640	164->165	39767	183->184	51374
146->147	43794	165->166	60292	184->185	50483
147->148	34203	166->167	50290	185->186	52951
148->149	55112	167->168	24334	186->187	65216
149->150	79217	168->169	6358	187->188	60631
150->151	24648	169->170	53736	188->189	54866
151->152	50472	170->171	43072	Total de números homólogos	
152->153	57048	171->172	41941	9.001.847	

Fonte: Autor

7.5 Validação dos Resultados Obtidos na Reconstrução 3D de Áreas Urbanas

Para validação dos resultados relacionou-se medidas reais da construção com alguns dos pontos obtidos da nuvem de pontos 3D denso. Para isso buscou-se na nuvem do 3D denso pontos que possam representar a largura e altura de parte da estrutura que compõe o Casarão da Capitania dos Portos de Pelotas/RS.

Assim, no processo de verificação foram selecionados pontos de parte da estrutura que se destacam com facilidade e, com isso, escolheu-se uma janela do Casarão para facilitar a identificação dos pontos para realizar as medições e, também, na nuvem de pontos 3D espaço.

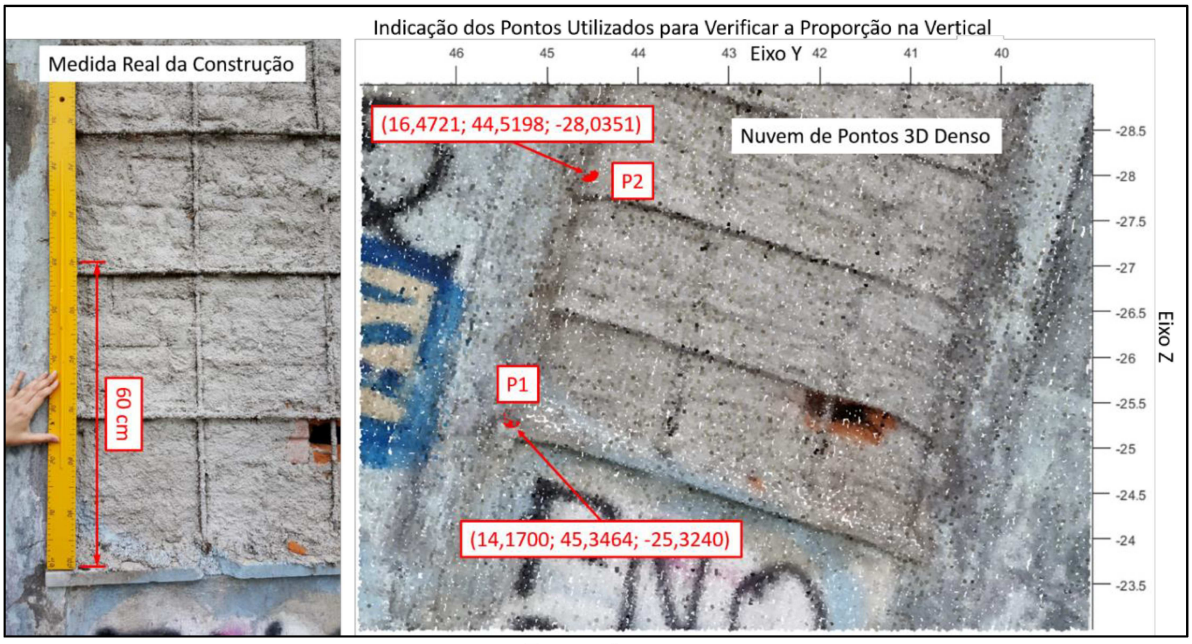
Os pontos selecionados estão apresentados na Figura 7.12, na qual é possível ter uma visão macro da região escolhida. Estes pontos correspondem à largura da janela e a distância da base da janela até a uma barra na janela. Para uma melhor visualização desses pontos a Figura 7.13 apresenta os pontos escolhidos para o cálculo da proporção na vertical, em que é apresentado a distância real e as coordenadas dos pontos utilizados, enquanto que a Figura 7.14 apresenta os pontos

escolhidos para o cálculo da proporção na horizontal, juntamente com a distância real e as coordenadas dos pontos utilizados.

Figura 7.12 – Cena urbana em 3D denso e os pontos escolhidos para verificar as proporções

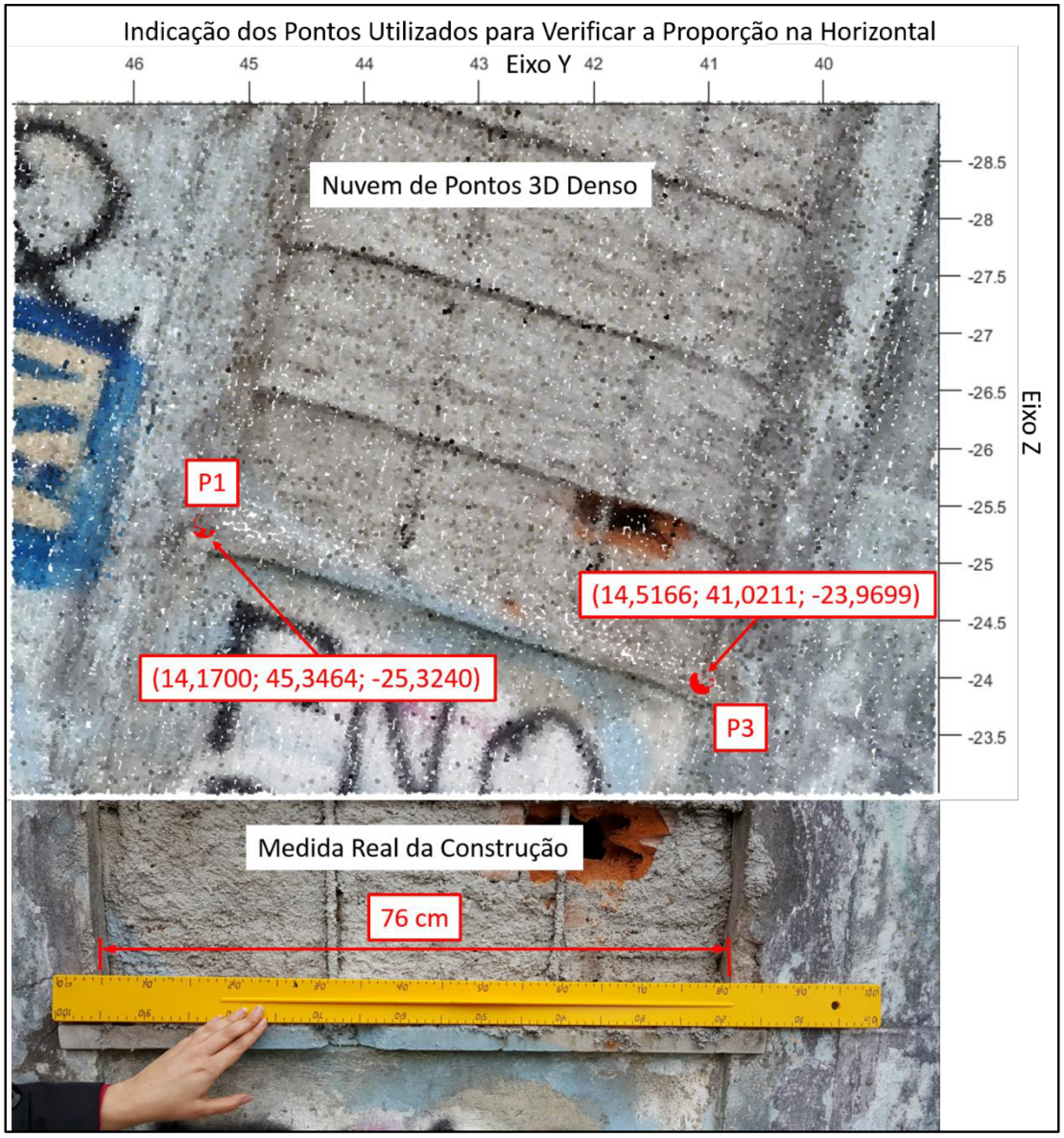
Fonte: Autor

Figura 7.13 – Os pontos, suas coordenadas e a distância real para a verificação da proporção na direção vertical



Fonte: Autor

Figura 7.14 – Os pontos, suas coordenadas e a distância real para a verificação da proporção na direção horizontal



Fonte: Autor

A Figura 7.13 apresenta os pontos P1 e P2 escolhidos na nuvem de pontos para verificar a relação entre a distância dos pontos da nuvem com as medidas reais da janela do Casarão. De acordo com a Figura 7.13 o ponto P1 está localizado na coordenada (14,1700 ; 45,3464 ; -25,3240) e o ponto P2 está localizado na coordenada (16,4721 ; 44,5198 ; -28,0351). A distância calculada entre os pontos P1 e P2 de 3,6514. Ainda na Figura 7.13 é possível verificar que a distância real desses pontos medidos diretamente na construção do casarão é de 60 cm e, assim, tem-se que a medida real é 16,4319 vezes maior do que a obtida na nuvem de pontos medido na direção vertical.

Já a Figura 7.14 apresenta os pontos P1 e P3 da nuvem de pontos que representa o Casarão escolhidos para outra verificação da relação entre a distância dos pontos da nuvem com as medidas reais da janela do Casarão. De acordo com a Figura 7.14 o ponto P1 está localizado na coordenada (14,1700 ; 45,3464 ; -25,3240) e o ponto P3 está localizado na coordenada (14,5166 ; 41,0211 ; -23,9699). A distância calculada entre os pontos P1 e P3 de 4,5455. Pela Figura 7.14 também é possível verificar a distância real desses pontos. Medindo diretamente na construção do casarão encontra-se que a distância entre P1 e P3 é de 76 cm. Com isso, tem-se que a medida real na horizontal é 16,7197 vezes maior do que a obtida pela nuvem de pontos.

Os resultados obtidos estão de acordo com o apresentado na revisão bibliográfica do Capítulo 2 em que a reconstrução é possível, entretanto o resultado obtido ficará em escala devido as incertezas inerentes do processo de calcular a profundidade dos pontos da cena.

Assim, baseado nas figuras apresentadas nesse capítulo e nas medições realizadas, pode-se dizer que a metodologia utilizada foi capaz de reconstruir em 3D uma edificação de área urbana com boa qualidade, porém em uma escala média de aproximadamente 16,5758 vezes menor.

Capítulo 8

8 CONCLUSÃO E TRABALHOS FUTUROS

Este capítulo apresenta a conclusão dos resultados obtidos pelo trabalho e, também, são apresentadas algumas propostas para trabalhos futuros.

8.1 Conclusão

Baseado nas condições e metodologias aplicadas nesse trabalho, nos resultados obtidos e na análise deles, obteve-se as seguintes conclusões.

O procedimento para realizar a modelagem 3D de uma edificação urbana deve seguir as seis etapas descritas no Capítulo 7, a saber:

Etapa 1 - Obter os parâmetros intrínsecos da câmera (calibração da câmera);

Etapa 2 - Obter dos pontos característicos da imagem;

Etapa 3 - Casar os pontos característicos;

Etapa 4 - Ordenar a sequência das imagens;

Etapa 5 - Calcular, pelos pares das imagens, os parâmetros extrínsecos de cada posição de câmera;

Etapa 6 - Realizar a triangulação para a obtenção do 3D.

O custo computacional da Etapa 4 pode ser eliminado ao obter as fotografias de forma ordenada. Entretanto a ordenação das imagens deve ser informada ao sistema que irá fazer a reconstrução 3D da estrutura de interesse.

O procedimento de reconstrução 3D descrito nesse trabalho obtém uma nuvem. Quanto mais pontos estão presentes nessa nuvem de pontos mais fiel é a reconstrução 3D. Os pontos dessa nuvem correspondem aos pontos homólogos encontrados nos pares de fotografias e quanto menor for a distância entre as câmeras mais pontos homólogos são obtidos. Assim, quanto maior for o número de fotografias tiradas mais fiel será a reconstrução 3D.

8.2 PROPOSTA PARA TRABALHOS FUTUROS

1) Utilizando a visão computacional, em especial as técnicas de casamento de pontos característicos, desenvolver técnicas de mapeamento e localização para robôs móveis como o SLAM (Simultaneous Localization And Mapping);

2) Estudar o funcionamento de câmeras omnidirecionais e técnicas de reconstrução 3D, utilizando tais câmeras omnidirecionais, de forma a aproveitar fotografias urbanas de 360 graus na reconstrução 3D de áreas urbanas;

3) Combinar técnicas de SLAM com a metodologia de reconstrução 3D através de fotografias com a intensão de potencializar o resultado e conseguir abranger uma área urbana maior;

4) Verificar o efeito de, após identificada as posições de câmera, utilizar transformações projetivas das áreas próximas aos pontos homólogos com o intuito aumentar o crescimento do número de pontos homólogos para a geração do 3D denso;

5) Desenvolver uma metodologia para realizar a segmentação de artefatos em imagens de edificações.

9 REFERENCIAS

ABDEL-AZIZ, Y. I.; KARARA, H. M. Direct linear transformation into object space coordinates in close-range photogrammetry. In: SYMPOSIUM ON CLOSE-RANGE PHOTOGRAMMETRY, 1., 1971, Urbana, Illinois. **Proceedings.** Urbana, Illinois, 1971. p. 1 – 18.

ALOIMONOS, J. Y.; SHULMAN, D. A. **Integration of Visual Modules:** An Extension of the Marr Paradigm. Massachusetts: Academic Press, 1989. 322 p.

ALOIMONOS, Y.. What I Have Learned. **Cvgip: Image Understanding,** [S.L.], v. 60, n. 1, p.74-85, jul. 1994. Elsevier BV. DOI: 10.1006/ciun.1994.1032. Disponível em: <http://api.elsevier.com/content/article/PII:S1049966084710321?httpAccept=text/xml>. Acesso em: 02 ago. 2014.
https://doi.org/10.1006/ciun.1994.1032

ALOIMONOS, Y.; ROSENFELD, A.. Principles of computer vision. In: YOUNG, T. Y.. **Handbook of pattern recognition and image processing (vol. 2):** computer vision. Orlando: Academic Press, 1994. Cap. 1. p. 1-15.

ARMAN, F.; AGGARWAL, J. K.. Model-based object recognition in dense-range images---a review. **Acm Computing Surveys (csur),** [S.L.], v. 25, n. 1, p.5-43, 1 mar. 1993. Association for Computing Machinery (ACM).
https://doi.org/10.1145/151254.151255

ARTHUR, D; VASSILVITSKII, S. K-means++: the advantages of careful seeding. In: PROCEEDINGS OF THE EIGHTEENTH ANNUAL ACM-SIAM SYMPOSIUM ON DISCRETE ALGORITHMS, 18., 2007, New Orleans. **Proceedings.** Philadelphia: Soda, 2007. p. 1027 – 1035. ISBN: 978-0-898716-24-5

BAJCSY, R.. Active perception. **Proceedings of The IEEE,** [S.L.], v. 76, n. 8, p.966-1005, ago. 1988. Institute of Electrical & Electronics Engineers (IEEE).
https://doi.org/10.1109/5.5968

BANSAL, S; AGGARWAL, D. Color Image Segmentation Using CIELab Color Space Using Ant Colony Optimization. **International Journal of Computer Applications,** [S. L.], v. 29, n. 9, p.28-34, set. 2011.
https://doi.org/10.5120/3590-4978

BARBOSA, T. C. P.; COSTA, H. S. M.. **Plano Diretor Participativo:** Guia para a elaboração pelos municípios e cidadãos. Brasília: Tecnopop, 2004. 160 p. Guia de práticas do Ministério das Cidades

BAY, H.; TUYTELAARS, T.; GOOL, L. V.. SURF: Speeded Up Robust Features. In: 9TH EUROPEAN CONFERENCE ON COMPUTER VISION, 9., 2006, Graz. **Proceedings.** Graz: Springer Berlin Heidelberg, 2006. p. 404 – 417.

BERTERO, M.; POGGIO, T.a.; TORRE, V.. Ill-posed problems in early vision. **Proceedings of The IEEE,** [S.L.], v. 76, n. 8, p.869-889, 1988. Institute of Electrical & Electronics Engineers (IEEE).
https://doi.org/10.1109/5.5962

BESL, P. J.; JAIN, R. C.. **Surfaces in Range Image Understanding.** New York: Springer Verlag, 1988. 339 p.
https://doi.org/10.1007/978-1-4612-3906-2

BESL, P. J.; JAIN, R. C.. Three-dimensional object recognition. **Acm Computing Surveys (csur): Annals of discrete mathematics,** [S.L.], v. 17, n. 1, p.75-145, 1 mar. 1985. Association for Computing Machinery (ACM).
https://doi.org/10.1145/4078.4081

BEYMER, D.; POGGIO, T.. Image Representations for Visual Learning. **Science,** [S.L.], v. 272, n. 5270, p.1905-1909, 28 jun. 1996. American Association for the Advancement of Science (AAAS).
https://doi.org/10.1126/science.272.5270.1905

BOWYER, K. et al. Why aspect graphs are not (yet) practical for computer vision. **Cvgip: Image Understanding,** [S.L.], v. 55, n. 2, p.212-218, mar. 1992. Elsevier BV.
https://doi.org/10.1016/1049-9660(92)90018-X

BRASIL. Lei nº 10.257, de 10 de junho de 2001. **Estatuto da Cidade e Legislação Correlata**. Brasília, SENADO FEDERAL: Subsecretaria de Edições Técnicas, p. 32-35. ISBN 85-7018-223-6

BROOKS, R. A.; CREINER, R.; BINFORD, T. O.. The ACRONYM model-based vision system. In: 6TH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., 1979, San Francisco. **Proceedings.** San Francisco: Morgan Kaufmann Publishers, 1979. p. 105 - 113.

BROWN, M.; LOWE, D. G.. Recognising panoramas. In: NINTH IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 9., 2003A, Washington. **Proceedings.** Washington: IEEE, 2003A. v. 2, p. 1218 - 1225.
https://doi.org/10.1109/ICCV.2003.1238630

BROWN, M.; LOWE, D. G.. **AutoStitch:** a new dimension in automatic image stitching. 2003B. Disponível em: <http://www.cs.bath.ac.uk/brown/autostitch/autostitch.html>. Acesso em: 15 maio 2014.

BUXTON, H.; HOWARTH, R. J.. Spatial and Temporal reasoning in the generation of dynamic scene representations. In: PROCEEDINGS OF SPATIAL AND TEMPORAL REASONING, 1995, Montreal. **Proceedings.** Montreal: Ijcai, 1995. p. 107 - 115.

C. C. SLAMA (Usa). American Society of Photogrammetry (Ed.). **Manual of Photogrammetry.** 4. ed. Falls Church, Virginia: Asprs Pubns, 1980

CAIXIA, D et al. The Improved Algorithm of Edge Detection Based on Mathematics Morphology. **International Journal of Signal Processing, Image Processing and Pattern Recognition,** [S. L.], v. 7, n. 5, p.309-322, 05 jul. 2014

CARRILLO, L. R. G. et al. Combining Stereo Vision and Inertial Navigation System for a Quad-Rotor UAV. **Journal of Intelligent & Robotic Systems**, [S. L.], v. 65, n. 1-4, p.373-387, jan. 2012.
https://doi.org/10.1007/s10846-011-9571-7

CARVALHO, C. S. et al. **O Estatuto da Cidade:** Comentado. São Paulo: Ministério das Cidades: Aliança das Cidades, 2010. 120 p

CORNELIUS, H. et al. Towards Complete Free-Form Reconstruction of Complex 3D Scenes from an Unordered Set of Uncalibrated Images. **Lecture Notes In Computer Science,** [S.L.], v. 3247, p.1-12, maio 2004. Springer Science + Business Media.
https://doi.org/10.1007/978-3-540-30212-4_1

CVGIP B: Image Understanding - Special issue on purposive, qualitative, active vision. Orlando: Academic Press, v. 56, n. 1, jul. 1992.

DI ZHANG; JIAZHONG HE. Face super-resolution reconstruction and recognition from low-resolution image sequences. **Computer Engineering And Technology (iccet)**, Chengdu, v. 2, p.620-624, 16 abr. 2010.
https://doi.org/10.1109/ICCET.2010.5485651

EMMERT, F. et al. Geoprocessamento como ferramenta de apoio à gerência de pavimentos em estradas florestais. **Ciência Florestal**, Santa Maria, v. 20, n. 1, p.81-94, 28 jul. 2010.
https://doi.org/10.5902/19805098

FAUGERAS, O. D.; LUONG, Q.; MAYBANK, S. J.. Camera Self-Calibration: Theory and Experiments. In: SECOND EUROPEAN CONFERENCE ON COMPUTER VISION, 2., 1992, Santa Margherita Ligure. **Proceedings.** London: Springer-verlag, 1992. p. 321 - 334.

FAUGERAS, O.. **Three-dimensional Computer Vision:** A Geometric Viewpoint. Massachusetts: The MIT Press, 1993. 663 p.

FERNYHOUGH, J. H.. **Generation of qualitative spation-temporal representations from visual input.** 1997. 166 f. Tese (Doutorado) - Curso de Computer Studies, University of Leeds, Leeds, 1997.

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,** [S.L.], v. 24, n. 6, p.381-395, 1 jun. 1981. Association for Computing Machinery (ACM).
https://doi.org/10.1145/358669.358692

FITZGIBBON, A. W.; ZISSERMAN, A.. Automatic Camera Recovery for Closed or Open Image Sequences. In: 5TH EUROPEAN CONFERENCE ON COMPUTER VISION, 5., 1998, London. **Proceedings.** London: Springer-verlag, 1998. p. 311 - 326.
https://doi.org/10.1007/BFb0055675

FLYNN, P. J.; JAIN, A. K. CAD-based computer vision: from CAD models to relational graphs. **IEEE Transactions On Pattern Analysis And Machine Intelligence,** [S.L.], v. 13, n. 2, p.114-132, fev. 1991. Institute of Electrical & Electronics Engineers (IEEE). https://doi.org/10.1109/34.67642

FLYNN, P. J.; JAIN, A. K.. 3D object recognition using invariant feature indexing of interpretation tables. **Cvgip: Image Understanding - Special Issue On Directions In Cad-based Vision,** Orlando, v. 55, n. 2, p.119-129, mar. 1992. https://doi.org/10.1016/1049-9660(92)90012-R

FORSYTH, D. A.; PONCE, J.. **Computer Vision:** A Modern Approach. New York: Pearson, 2002. 693 p.

GIMEL'FARB, G.. Stereo Terrain Reconstruction by Dynamic Programming. In: JÄHNE, B.; HAUßECKER, H.; GEIßLER, P. (Ed.). **Handbook of Computer Vision and Applications Volume 2:** Signal Processing and Pattern Recognition. San Diego: Academic Press, 1999. Cap. 18. p. 505-530.

GLUCKMAN, J.; NAYAR, S. K.. Rectifying transformations that minimize resampling effects. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 16., 2001, Kauai. **Proceedings.** [S.L.]: IEEE, 2001. p. 111 - 117. https://doi.org/10.1109/CVPR.2001.990463

GOLUB, G. H.; VAN LOAN, C. F.. **Matrix Computations.** 2. ed. Baltimore, Maryland: Johns Hopkins University Press, 1989. 728 p.

HARRIS, C; STEPHENS, M. A combined corner and edge detector. In: 4TH ALVEY VISION CONFERENCE, 4., 1988, Manchester. **Proceedings.** Manchester: The British Machine Vision Conference (bmvc), 1988. p. 147 – 151. https://doi.org/10.5244/C.2.23

HARTLEY, R. I.. In defense of the eight-point algorithm. **IEEE Transactions On Pattern Analysis And Machine Intelligence,** [S.L.], v. 19, n. 6, p.580-593, jun. 1997. Institute of Electrical & Electronics Engineers (IEEE). https://doi.org/10.1109/34.601246

HARTLEY, R. I.. Estimation of relative camera positions for uncalibrated cameras. In: SECOND EUROPEAN CONFERENCE ON COMPUTER VISION, 2., 1992, Santa Margherita Ligure. **Proceedings.** London: Springer-verlag, 1992. p. 579 - 587. https://doi.org/10.1007/3-540-55426-2_62

HARTLEY, R. I.. Self-calibration from multiple views with a rotating camera. In: THIRD EUROPEAN CONFERENCE ON COMPUTER VISION, 3., 1994, Stockholm. **Proceedings.** New York: Springer-verlag, 1994. p. 471 - 478. https://doi.org/10.1007/3-540-57956-7_52

HARTLEY, R.; ZISSERMAN, A.. **Multiple View Geometry in Computer Vision.** 2. ed. Cambridge: Cambridge University Press, 2004. 655 p. ISBN: 9780521540513. https://doi.org/10.1017/CBO9780511811685

HEIKKILÄ, J.; SILVÉN, O.. Advanced Search Include Citations A four-step camera calibration procedure with implicit image correction. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 97., 1997, San Juan. **Proceedings.** San Juan: IEEE, 1997. p. 1106 – 1113.

HENN, A. et al. Automatic classification of building types in 3D city models. **Geoinformatica**, [S. L.], v. 16, n. 2, p.281-306, 01 abr. 2012. https://doi.org/10.1007/s10707-011-0131-x

HORAUD, R. et al. The Advantage of Mounting a Camera onto a Robot Arm. In: THE EUROPE-CHINA WORKSHOP ON GEOMETRICAL MODELLING AND INVARIANTS FOR COMPUTER VISION, 1., 1995, Xian. **Proceedings.** Xian: Inrialpes, 1995. p. 206 - 213.

HOWARTH, R.. **Spatial Representation, Reasoning and Control for a Surveillance System.** 1994. 338 f. Tese (Doutorado) - Curso de Computer Science, Queen Mary And Westfield College, London, 1994.

JULESZ, B.. Binocular Depth Perception of Computer-Generated Patterns. **Bell System Technical Journal,** [S.L.], v. 39, n. 5, p.1125-1162, set. 1960. Institute of Electrical & Electronics Engineers (IEEE). https://doi.org/10.1002/j.1538-7305.1960.tb03954.x

KAMITOMO, H.; LU, C.. 3-D face recognition method based on optimum 3-D image measurement technology. **Artificial Life And Robotics**, [S. L.], v. 16, n. 4, p.551-554, 01 fev. 2012. https://doi.org/10.1007/s10015-011-0982-0

KATO, T.; NINOMIYA, Y.; MASAKI, I.. Preceding vehicle recognition based on learning from sample images. **Intelligent Transportation Systems**, [S. L.], v. 3, n. 4, p.252-260, dez. 2002. https://doi.org/10.1109/TITS.2002.804752

KLETTE, R.; KOSCHAN, A.; SCHLÜNS, K.. **Computer Vision:** Räumliche Information aus digitalen Bildern. Auflage: Vieweg Verlagsgesellschaft, 1996. 382 p.

KLIR, G.. **Facets of Systems Science.** 2. ed. New York: Springer Science & Business Media, 2001. 740 p. https://doi.org/10.1007/978-1-4615-1331-5

KOENDERINK, J. J.. **Solid Shape.** Massachusetts: The MIT Press, 1990. 715 p.

LANDY, M. S.; MALONEY, L. T.; PAVEL, M. (Ed.). **Exploratory Vision:** The Active Eye. [S.L.]: Springer, 1996. 344 p. https://doi.org/10.1007/978-1-4612-3984-0

LAURINI, R.. French Local Planning Practice. In: BATTY, Michael; HUTCHINSON, Bruce. **Systems Analysis in Urban Policy-Making and Planning.** Oxford: Plenum Press, 1982A. Cap. 13. p. 193-203. (Volume 12).

LAURINI R. **Nouveaux outils informatiques pour l'élaboration conjointe des plans d'urbanisme. Proceedings** ofthe 9th European Symposium on Urban Data Management Symposium (UDMS) Valencia, Spain, October 26-29, 1982B.

LAURINI, R.. **Information Systems for Urban Planning:** a Hypermedia Cooperative Approach. London: Taylor And Francis, 2001. 349 p. ISBN: 0-203-48506-8.

LEUTENEGGER, S.; CHLI, M.; SIEGWART, R.. BRISK: Binary Robust invariant scalable keypoints. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 1., 2011, Barcelona. **Proceedings.** Barcelona: IEEE, 2011. p. 2548 – 2555.

LONGUET-HIGGINS, H. C.. A computer algorithm for reconstructing a scene from two projections. **Nature,** [S.L.], v. 293, n. 5828, p.133-135, 10 set. 1981. Nature Publishing Group.
https://doi.org/10.1038/293133a0

LOOP, C.; ZHANG, Z.. Computing rectifying homographies for stereo vision. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 14., 1999, Fort Collins. **Proceedings.** [S.L.]: IEEE, 1999. p. 125 - 131.
https://doi.org/10.1109/CVPR.1999.786928

LOVE, N. S.; MASAKI, I.; HORN, B. K. P. Recognition of 3D compressed images and its traffic monitoring applications. **Intelligent Vehicles Symposium**, Dearborn, Mi, p.463-467, 3 out. 2000.
https://doi.org/10.1109/IVS.2000.898386

LÜTKEPOHL, H.. **Handbook of Matrices.** New York: Wiley, 1997. 320 p.

MA, Y. et al. **An Invitation to 3-D Vision:** From Images to Geometric Models. Orlando: Springer, 2004. 528 p.
https://doi.org/10.1007/978-0-387-21779-6

MARR, D. **Vision:** A Computational Investigation into the Human Representation and Processing of Visual Information. Massachusetts: The MIT Press, 1982. 432 p.

MARR, D.; HILDRETH, E.. Theory of Edge Detection. **Proceedings of The Royal Society B: Biological Sciences,** [S.L.], v. 207, n. 1167, p.187-217, 29 fev. 1980. The Royal Society.
https://doi.org/10.1098/rspb.1980.0020

MARR, D.; POGGIO, T.. A Computational Theory of Human Stereo Vision. **Proceedings of The Royal Society B: Biological Sciences,** [S.L.], v. 204, n. 1156, p.301-328, 23 maio 1979. The Royal Society.
https://doi.org/10.1098/rspb.1979.0029

MATAS, J. et al. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In: THE BRITISH MACHINE VISION CONFERENCE, 13., 2002, Cardiff. **Proceedings.** Cardiff: Bmvc, 2002. p. 1 – 10.
https://doi.org/10.5244/C.16.36

MATHWORKS SUPPORT TEAM. Suporte Técnico (Org.). **Does MATLAB support USB communication?** 2012. Forum da MathWorks sobre dúvidas referentes ao MATLAB. Disponível em: <http://www.mathworks.com/matlabcentral/answers/98146-does-matlab-support-usb-communication>. Acesso em: 05 maio 2015.

MATOUSEK, M.; SARA, R.; HLAVAC, V.. Data-optimal rectification for fast and accurate stereovision. In: THIRD INTERNATIONAL CONFERENCE ON IMAGE AND GRAPHICS (ICIG'04), 3., 2004, Hong Kong. **Proceedings.** [S.L.]: IEEE, 2004. p. 212-215.
https://doi.org/10.1109/ICIG.2004.52

MAYBANK, S. J.; FAUGERAS, O. D.. A theory of self-calibration of a moving camera. **Int J Comput Vision,** [S.L.], v. 8, n. 2, p.123-151, ago. 1992. Springer Science + Business Media.
https://doi.org/10.1007/BF00127171

MELEN, T. **Geometrical modelling and calibration of video cameras for underwater navigation.** 1994. 129 f. Tese (Doutorado) – Curso de Norges Tekniske Høgskole, Institutt For Teknisk Kybernetikk, Trondheim, 1994.

MOHR, R.. Projective geometry and computer vision. In: CHEN, C. H.; PAU, L. F.; WANG, P. S. P.. **Handbook of pattern recognition & computer vision.** River Edge: World Scientific Publishing, 1993. p. 369-393.
https://doi.org/10.1142/9789814343138_0013

NEWMAN, T. S.; FLYNN, P. J.; JAIN, A. K.. Model-Based Classification of Quadric Surfaces. **Cvgip: Image Understanding,** [S.L.], v. 58, n. 2, p.235-249, set. 1993. Elsevier BV.
https://doi.org/10.1006/ciun.1993.1040

NI QIAKAI. GUO CHAO. YANG JING. Research of face image recognition based on probabilistic neural networks. **Control And Decision Conference (ccdc)**, Taiyuan, p.3885-3888, 23 maio 2012.
https://doi.org/10.1109/CCDC.2012.6243102

NISHIHARA, H. K.. Practical Real-Time Imaging Stereo Matcher. **Optical Engineering,** [S.L.], v. 23, n. 5, p.536-545, 1 out. 1984. SPIE-Intl Soc Optical Eng.
https://doi.org/10.1117/12.7973334

OTSU, N. A Threshold Selection Method from Gray-Level Histograms. **IEEE Transactions on Systems, Man and Cybernetics,** [S. L.], v. 9, n. 1, p.62-66, jan. 1979. Institute of Electrical & Electronics Engineers (IEEE).
https://doi.org/10.1109/TSMC.1979.4310076

PAJDLA, T.; HLAVÁC, V.. Camera Calibration and Euclidean Reconstruction from Known Translations. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1., 1998, Santa Barbara. **Proceedings.** [S.L.]: IEEE, 1998. p. 421 - 426.
https://doi.org/10.1109/CVPR.1998.698640

PAUL, R. P.. **Robot Manipulators:** Mathematics, Programming, and Control. Massachusetts: The MIT Press, 1981. 279 p.

PLACHT, S. et al. ROCHADE: Robust Checkerboard Advanced Detection for Camera Calibration. In: FLEET, D. et al. **Computer Vision – ECCV 2014:** Lecture Notes in Computer Science. 8692. ed. [S.L.]: Springer International Publishing, 2014. p. 766-779.

POGGIO, T.; TORRE, V.; KOCH, C.. Computational vision and regularization theory. **Nature,** [S.L.], v. 317, n. 6035, p.314-319, 26 set. 1985. Nature Publishing Group. https://doi.org/10.1038/317314a0

POLLARD, S. B; MAYHEW, J. e W; FRISBY, J. P. PMF: A stereo correspondence algorithm using a disparity gradient limit. **Perception,** [S.L.], v. 14, n. 4, p.449-470, mar. 1985. Pion Ltd. https://doi.org/10.1068/p140449

POLLEFEYS, M.; KOCH, R.; VAN GOOL, L.. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In: SIXTH INTERNATIONAL CONFERENCE ON COMPUTER VISION, 6., 1998, Bombay. **Proceedings.** [S.L.]: IEEE, 1998. p. 90 - 95. https://doi.org/10.1109/ICCV.1998.710705

PRESS, W. H. et al. **Numerical Recipes in C:** The Art of Scientific Computing. 2. ed. Cambridge: Cambridge University Press, 1992. 994 p.

PRODUCCINE. Técnico em Edição de Filmes. **Faça um estúdio Chroma Key caseiro!!** 2013. Video explicativo da utilização da técnica de Chroma Key. Disponível em: <https://www.youtube.com/watch?v=RYSd-UECd7I>. Acesso em: 05 maio 2015.

RAHMAN, M. T. et al. On Interactive Image Recognition for Human-Machine Interfacing. **Convergence Information Technology**, Gyeongju, p.1807-1812, 21 nov. 2007. https://doi.org/10.1109/ICCIT.2007.339

ROERDINK, J. B. T. M.; MEIJSTER, A. The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. **Fundamenta Informaticae,** Amsterdam, v. 41, n. 1, p.187-228, 02 abr. 2000.

ROSTEN, E; DRUMMOND, T. Machine Learning for High-Speed Corner Detection. In: 9TH EUROPEAN CONFERENCE ON COMPUTER VISION, 9., 2006, Graz. **Proceedings.** Graz: Springer Berlin Heidelberg, 2006. p. 430 – 443. https://doi.org/10.1007/11744023_34

ŠÁRA, R.. Finding the Largest Unambiguous Component of Stereo Matching. In: 7TH EUROPEAN CONFERENCE ON COMPUTER VISION, 7., 2002, Copenhagen. **Proceedings.** London: Springer-verlag, 2002. p. 900 - 914. https://doi.org/10.1007/3-540-47977-5_59

SEMPLE, J. G.; KNEEBONE, G. T.. **Algebraic Projective Geometry.** Oxford: Clarendon Press, 1998. 412 p.

SHI, J.; MALIK, J.. Normalized cuts and image segmentation. **IEEE Trans. Pattern Anal. Machine Intell.,** [S. L.], v. 22, n. 8, p.888-905, 06 ago. 2000. Institute of Electrical & Electronics Engineers (IEEE).
https://doi.org/10.1109/34.868688

SMEETS, D et al. Objective 3D face recognition: Evolution, approaches and challenges. **Forensic Science International**, [S. L.], v. 201, n. 1, p.125-132, 10 set. 2010.
https://doi.org/10.1016/j.forsciint.2010.03.023

SONKA, M.; HLAVAC, V.; BOYLE, R.. **Image Processing, Analysis, and Machine Vision.** 3. ed. Toronto: Thomson Learning, 2006. 829 p. ISBN: 9780495244287.

SOUCY, M.; LAURENDEAU, D.. Surface modeling from dynamic integration of multiple range views. In: 11TH IAPR INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, CONFERENCE A: COMPUTER VISION AND APPLICATIONS, PROCEEDINGS.,, 11., 1992, Piscataway. **Proceedings.** Piscataway: IEEE, 1992. p. 449 - 452.
https://doi.org/10.1109/ICPR.1992.201597

TANAKA, S.; KAK, A. C.. A Rule-Based Approach to Binocular Stereopsis. In: JAIN, R. C.; JAIN, A. K.. **Springer Series in Perception Engineering.** New York: Springer, 1990. p. 33-139.
https://doi.org/10.1007/978-1-4612-3360-2_2

TIKHONOV, A. N.; ARSENIN, V. Í.. **Solutions of ill-posed problems.** Wshington: Winston, 1977. 258 p.

TRIGGS, B. et al. Bundle Adjustment: A Modern Synthesis. In: TRIGGS, Bill; ZISSERMAN, A.; SZELISKI, R. (Ed.). **Vision Algorithms:** Theory and Practice. London: Springer-verlag, 2000. p. 298-372.
https://doi.org/10.1007/3-540-44480-7_21

TRINH, H.; KIM, D.; JO, K.. Supervised training database for building recognition by using cross ratio invariance and SVD-based method. **Applied Intelligence**, Hingham, Ma, v. 32, n. 2, p.216-230, abr. 2010.
https://doi.org/10.1007/s10489-010-0221-8

TSAI, R.. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. **IEEE J. Robot. Automat.,** [S.L.], v. 3, n. 4, p.323-344, ago. 1987. Institute of Electrical & Electronics Engineers (IEEE).
https://doi.org/10.1109/JRA.1987.1087109

ULLMAN, S.; BASRI, R.. Recognition by linear combinations of models. **IEEE Transactions On Pattern Analysis And Machine Intelligence,** [S.L.], v. 13, n. 10, p.992-1006, out. 1991. Institute of Electrical & Electronics Engineers (IEEE).
https://doi.org/10.1109/34.99234

VIDAL, B. Chroma Key Visual Feedback Based on Non-Retroreflective Polarized Reflection in Retroreflective Screens. **IEEE Transactions On Broadcasting,** [s. L.], v. 58, n. 1, p.144-150, mar. 2012. https://doi.org/10.1109/TBC.2011.2174275

WANG, W.. A Study on Contour Feature Algorithm for Vehicle Type Recognition. **International Joint Conference Artificial Intelligenceon**, Hainan Island, p.452-455, 25 abr. 2009. https://doi.org/10.1109/JCAI.2009.56

WECHSLER, H. **Computational vision.** Massachusetts: Academic Press, 1990. 558 p.

WERNER, T.; HERSCH, R. D.; HLAVÁC, V.. Rendering Real-World Objects Using View Interpolation. In: 5TH INTERNATIONAL CONFERENCE ON COMPUTER VISION, 5., 1995, Boston. **Proceedings.** Boston: IEEE Computer Society, 1995. p. 957 - 962. https://doi.org/10.1109/ICCV.1995.466831

XIAO, J.; QUAN, L.. Multiple view semantic segmentation for street view images. In: INTERNATIONAL CONFERENCE ON COMPUTER VISION, 12., 2009, Kyoto. **Proceedings.** Kyoto: IEEE, 2009. p. 686 – 693.

ZHANG, W.; KOŠECKÁ, J.. Hierarchical building recognition. **Image And Vision Computing**, [S. L.], v. 25, n. 5, p.704-716, 1 maio 2007. https://doi.org/10.1016/j.imavis.2006.05.016

ZHANG, Z.. A flexible new technique for camera calibration. **IEEE Transactions On Pattern Analysis And Machine Intelligence,** [S.L.], v. 22, n. 11, p.1330-1334, 2000. Institute of Electrical & Electronics Engineers (IEEE). https://doi.org/10.1109/34.888718

ZHU-YU, Z.; TIAN-MIN, D.; XIAN-YANG, L.. Study for Vehicle Recognition and Classification Based on Gabor Wavelets Transform & HMM. **2011 International Conference On Consumer Electronics, Communications And Networks**, Xianning, v. 6, p.5272-5276, 16 abr. 2011. ISBN: 978-1-61284-458-9. https://doi.org/10.1109/CECNET.2011.5768716