DETERMINAÇÃO DE TRAJETÓRIA DE CÂMERA PARA DETECÇÃO DE OBJETOS ABANDONADOS

Allan Freitas da Silva

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Rio de Janeiro
Março de 2015

DETERMINAÇÃO DE TRAJETÓRIA DE CÂMERA PARA DETECÇÃO DE OBJETOS ABANDONADOS

Allan Freitas da Silva

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

Prof. Hélio Côrtes Vieira Lopes, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2015

Silva, Allan Freitas da

Determinação de Trajetória de Câmera para Detecção de Objetos Abandonados/Allan Freitas da Silva. – Rio de Janeiro: UFRJ/COPPE, 2015.

IX, 61 p.: il.; 29, 7cm.

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2015.

Referências Bibliográficas: p. 56 – 61.

1. Trajetória. 2. SfM. 3. Panorâmica. I. Silva, Eduardo Antônio Barros da *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

DETERMINAÇÃO DE TRAJETÓRIA DE CÂMERA PARA DETECÇÃO DE OBJETOS ABANDONADOS

Allan Freitas da Silva

Março/2015

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Programa: Engenharia Elétrica

Apresenta-se, neste trabalho, um método para a estimação da trajetória de uma câmera em movimento e a geração de uma imagem panorâmica para auxiliar na detecção de objetos abandonados. A câmera foi montada em um robô percorrendo um trilho em movimento retilíneo e tem o objetivo de monitorar um ambiente visualmente carregado, detectando anomalias. A trajetória é estimada computando a geometria existente entre pares de imagens consecutivas. De posse da posição de onde foi registrado cada quadro de um vídeo, monta-se uma imagem ao estilo panorâmica.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

DETERMINING OF CAMERA TRAJECTORY FOR ABANDONED OBJECTS DETECTION

Allan Freitas da Silva

March/2015

Advisors: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Department: Electrical Engineering

In this work, we present a method for moving-camera trajectory estimation and panoramic image generation, to aid in abandoned objects detection. The camera was mounted on a robot performing a linear back-and-forth motion with the purpuse to survey a cluttered environment and detect anomalies. Tha trajectory is estimated by computing the geometry between consecutive images. With the knowledge of the position where each frame was recorded, we generate a panoramic image.

Sumário

Lista de Figuras **viii**

1 Introdução **1**
1.1 Organização do Texto 2

2 Trabalhos Relacionados **3**
2.1 Detecção de Objetos 3
2.2 Alinhamento 6
2.3 Estrutura da Cena 7
2.4 *Simultaneous Localization and Mapping* (SLAM) 8

3 Geometria de Múltiplas Vistas **10**
3.1 Coordenadas Homogêneas 10
3.2 Modelo de Câmera 11
3.3 Geometria Epipolar 13
3.3.1 Matriz Fundamental 14
3.3.2 Matriz Essencial 16
3.4 Cálculo da Matriz Fundamental 17
3.4.1 Algoritmo de 8 Pontos 17
3.4.2 Algoritmo de 7 Pontos 18
3.4.3 Algoritmo de 5 Pontos 18

4 Reconstrução da Cena e do Movimento **20**
4.1 Reconstrução com Duas Vistas 20
4.1.1 Decomposição em Câmeras 21
4.1.2 Triangulação 23
4.1.3 Ambiguidade na Reconstrução 24
4.2 Reconstrução para Múltiplas Vistas 26

5 Método para Estimação da Trajetória de Câmera **31**
5.1 Método Proposto 31
5.1.1 Correspondência entre Pontos 31

5.1.2 Geometria Epipolar 32
5.1.3 Trajetória de Câmera 33
5.1.4 Montagem da Imagem Panorâmica 35
5.2 Base de Dados 35

6 Resultados e Discussões **38**
6.1 Correspondência entre Pontos 38
6.2 Geometria Epipolar 40
6.3 Trajetória de Câmera 41
6.4 Montagem da Imagem Panorâmica 47

7 Conclusões **53**
7.1 Trabalho Realizado 53
7.2 Próximos Passos 53

A Lista de Artigos Derivados deste Trabalho **55**

Referências Bibliográficas **56**

Lista de Figuras

3.1 Projeção em um plano de imagem para uma câmera do tipo *pinhole*. . 12
3.2 Geometria que relaciona vistas diferentes de uma mesma cena. 14

4.1 Possíveis soluções para a decomposição de câmeras a partir da matriz essencial. 22
4.2 Triangulação de um ponto sem erros. 23
4.3 Triangulação de um ponto com erro na posições dos pontos correspondentes. 24
4.4 Reconstrução projetiva de uma cena. 25
4.5 Reconstrução métrica de uma cena. 26
4.6 Reconstrução inicial de uma cena. 27
4.7 Correspondências entre os pontos $\mathbf{x_i}$ de uma nova imagem e a nuvem de pontos reconstruídos $\mathbf{X_i}$. 28
4.8 Ampliação da nuvem de pontos. 28
4.9 Pontos equivalentes em sistemas de coordenadas diferentes. 30

5.1 Correspondências válidas entre pontos das imagens. 32
5.2 Projeções obtidas com uma câmera em movimento retilíneo, com orientação perpendicular ao movimento. 34
5.3 Montagem da imagem panorâmica. 36
5.4 Sistema para monitoramento de um ambiente industrial com uma câmera montada em uma plataforma robotica. 36
5.5 Trecho mais à direita do ambiente monitorado pelo sistema. 37
5.6 Trecho central do ambiente monitorado pelo sistema. 37
5.7 Trecho mais à esquerda do ambiente monitorado pelo sistema. 37

6.1 Par de vistas utilizadas para exemplificar a obtenção de pontos correspondentes. 38
6.2 Pares de pontos correspondentes obtidos pelo SURF. 39
6.3 Pares de pontos correspondentes obtidos pelo SURF, após eliminação dos pares com ângulo grande e escala diferente. 39

6.4 Pares de pontos correspondentes obtidos pelo SURF, após eliminação dos pares com ângulo grande, escala diferente e inconsistência entre quadros. 40
6.5 Número total de correspondências para cada par de quadros. 40
6.6 Geometria epipolar para uma câmera efetuando um movimento de translação pura. 41
6.7 Conjunto de linhas epipolares para o par de quadros 14. 42
6.8 Conjunto de linhas epipolares para o par de quadros 59. 42
6.9 Rotação Relativa entre cada par de quadros. 43
6.10 Trajetória de câmera para o vídeo ext-part02-video01. 44
6.11 Trajetória de câmera para o vídeo ext-part03-video13. 45
6.12 Trajetória de câmera para o vídeo de referência. 46
6.13 Quadros reprojetados no plano considerado. 47
6.14 Imagens panorâmicas compostas a partir do vídeo ext-part02-video01. 49
6.15 Imagens panorâmicas compostas a partir do vídeo ext-part03-video06. 50
6.16 Imagens panorâmicas compostas a partir do vídeo de referência. . . . 51
6.17 Identificação de um objeto na imagem-panorama. 52

Capítulo 1

Introdução

Em um ambiente industrial, a tecnologia pode permitir a otimização de processos e minimização de gastos. Através de diversos tipos de equipamentos sensores, é possível monitorar remotamente um ambiente de modo a fornecer um detalhamento contínuo das condições de operação de maneira automatizada. Este tipo de sistema é ainda mais interessante para ambientes de difícil acesso ou em locais perigosos, pois também pode gerar redução dos riscos trabalhistas.

Dentro deste contexto, a visão computacional pode ser aplicada na detecção e reconhecimento de objetos na cena, o que pode indicar vazamentos, fogo, etc. Utilizando câmeras estáticas, uma técnica simples de subtração de fundo pode ser utilizada para deteção de novidades no ambiente. Entretanto, com câmeras móveis é possível aumentar o alcance da câmera sem que isso implique em um aumento considerável de custo [1], necessitando, porém, de técnicas mais complexas para compensar ou estimar o movimento da câmera.

Este trabalho se destina a fornecer uma abordagem para a estimação da trajetória e orientação de uma câmera em movimento para cada quadro, também chamada de *camera pose*, utilizando algoritmos estabelecidos na literatura. Restringiu-se que o movimento da câmera seja proveniente de um deslocamento retilíneo, sendo destinado para uso com uma biblioteca específica para detecção de objetos abandonados [2].

Com a posição da câmera em cada instante de tempo, monta-se uma imagem ao estilo de uma panorâmica, mesclando todas as vistas. A imagem gerada tem por objetivo auxiliar um operador ou até mesmo ser utilizada no processo de deteção de anomalias.

As principais contribuições deste trabalho são as seguintes. Durante o processo de estimação de pontos correspondentes entre imagens, implementou-se uma nova métrica para eliminação de pares de pontos menos estáveis. Também implementou-se uma alteração no método de cálculo da geometria entre vistas, que propicia a estimação de um movimento sem variações bruscas. Por fim, o algoritmo de

estimação da trajetória foi adaptado para as restrições de movimento impostas no sistema de interesse.

1.1 Organização do Texto

O presente documento é organizado da seguinte maneira: O Capítulo 2 mostra alguns trabalhos de visão computacional que podem ser aplicados no monitoramento de um ambiente industrial. O Capítulo 3 apresenta a geometria que relaciona as vistas de um mesmo ambiente. Já o Capítulo 4 mostra como essa geometria pode ser utilizada para estimar a distribuição dos elementos da cena e das câmeras no espaço tridimensional. O Capítulo 5 exibe o método proposto para a estimação do movimento da câmera e posterior geração de uma imagem panorâmica, com os resultados presentes no Capítulo 6. Por fim, o Capítulo 7 expõe as conclusões do trabalho e os possíveis próximos passos para estender os resultados aqui alcançados.

Capítulo 2

Trabalhos Relacionados

A visão computacional é uma área amplamente estudada e possui diversas técnicas que podem ser aplicadas na identificação de anomalias em uma planta industrial. Técnicas para detecção de objetos podem ser adaptadas para alertar sobre um incêndio ou vazamento, além de informar sobre objetos abandonados em determinado ambiente. Algoritmos de alinhamento, por sua vez, permitem sincronizar vídeos gravados em instantes diferentes, possibilitando assim que um vídeo recém-adquirido seja comparado a uma gravação com prévia validação das condições normais do ambiente. A determinação da estrutura tridimensional dos objetos da cena permite mapear a cena em outro tipo de representação e encontrar trajetória e orientação da câmera. Por fim, algoritmos de SLAM (*Simultaneous Localization and Mapping*) permitem realizar um mapeamento do ambiente em si e da posição de um agente que o percorre, com o auxílio de sensores e informações estatísticas. A seguir são apresentados trabalhos relevantes com foco nas áreas citadas.

2.1 Detecção de Objetos

Para detecção de objetos utiliza-se, em geral, alguma forma de subtração das informações do ambiente em segundo plano, caracterizando o mesmo a partir de modelos determinísticos, como o uso de um vídeo de refêrencia, ou estatísticos, a partir da modelagem estocástica de elementos do vídeo. Um dos trabalhos mais importantes na modelagem estatística do ambiente para câmeras fixas foi publicado por Stauffer and Grimson [3] e serve como base para muitos estudos posteriores. Nesta proposta, cada píxel de um quadro é modelado como uma mistura de K gaussianas, cada qual com um peso associado.

As gaussianas são ordenadas a partir da razão *peso*/*variância*, de modo que as N primeiras são consideradas como pertencentes ao ambiente, porque pixels nestas regiões tendem a ser mais prováveis e possuir um valor mais exato. Os pixels de um novo quadro adquirido são comparados ao seu modelo correspondente e associados a

uma das gaussianas, o que permite classificá-los em pertencentes ou não ao ambiente. Utilizando um algoritmo de maximização do valor esperado, atualiza-se média e variância da gaussiana escolhida, e o peso de todas as gaussianas.

Uma técnica que altera parte da estrutura estrutura proposta por Stauffer and Grimson é vista em [4], que apresenta uma topologia para a detecção de objetos utilizando uma decomposição dos quadros do vídeo com uma transformada *wavelet*. Com os coeficientes da *wavelet*, constrói-se uma matriz contendo as sub-bandas que é utilizada na subtração de fundo por mistura de gaussianas, para então ocorrer a reconstrução do quadro no domínio espacial.

Com o avanço da robótica, cada vez mais buscam-se soluções para câmeras móveis que possam ser acopladas a diversos dispositivos, cujo movimento deve ser estimado ou compensado. Um exemplo está em [5], que considera que o movimento pode ser compensado com uma transformação afim entre quadros consecutivos. A seguir, calcula-se a diferença entre o quadro atual e o anterior alterado pela transformação, para determinar os trechos onde ocorre movimento. Já em [6], utiliza-se de uma transformação bilinear entre os quadros para compensação do movimento, após realizar um rastreamento de pontos nas imagens.

Técnicas de subtração de fundo são bem adequadas para o movimento realizado por câmeras do tipo PTZ (*pan-tilt-zoom*). Para compensar o efeito de rotação, realiza-se o alinhamento espacial dos quadros e cria-se um modelo de fundo similar a uma imagem paronâmica, também chamado de mosaico, que vai englobar informações do ambiente para diferentes ângulos de visão. Cada novo quadro é alinhado a esse modelo, e os trechos correspondentes são comparados. Em [7] é proposto o alinhamento através do cálculo da homografia entre os quadros, e o mosaico é modelado com mistura de gaussianas, *kernel* não-paramétrico [8] ou *codebook* [9]. O mosaico pode ser ampliado caso haja partes do quadro aparecendo em posições inéditas. De maneira similar, em [10] permite-se compensar erros geométricos do modelo ao considerar que cada observação de um píxel é representada como uma mistura de processos aleatórios dentro de uma região ao redor da posição atual, somada a ruído.

O trabalho [11] trata do efeito de *zoom* que pode ocorrer durante o funcionamento da câmera. O sistema proposto realiza a detecção de eventos via subtração de fundo, mantendo o funcionamento mesmo em trechos onde a câmera sofre um efeito de *zoom in* ou *zoom out*. Para tanto, utiliza a diferença entre os quadros como medidor do início do zoom, onde deve ocorrer um alinhamento espacial entre os quadros. Já em [12] é proposto o uso de um modelo de fundo com múltiplos *sprites* em posições ângulares diferentes diferentes, para contornar o problema de rotações superiores a $180\,^{\circ}$.

Por sua vez, o artigo [13] mostra correções a serem aplicadas à homografia, pois

pode haver erros devido à paralaxe. O algoritmo calcula a verossimilhança para a disparidade nos resultados, e engloba esse cálculo na detecção de objetos, gerando uma função de energia que deve ser minimizada através de uma otimização de rótulo. O trabalho é continuado em [14], cuja grande contribuição é implementar o registro de uma imagem na outra a partir de uma homografia com várias camadas, que representa diversas homografias distintas. Utiliza-se o RANSAC [15] para obter uma homografia, e os pontos rejeitados pelo mesmo são utilizados no cálculo de uma nova homografia, com o processo se repetindo até o máximo desejado. As transformações obtidas são aplicadas em cada quadro e, quando houver sobreposição das homografias, escolhe-se a mais adequada para cada píxel. A detecção de objetos ocorre da mesma maneira que o trabalho anterior, com a minimização de uma função de energia.

Um trabalho que trata de uma trajetória genérica é visto em [16]. A detecção de um objeto isolado é feita a partir dos pontos obtidos pelo SIFT [17]. Para o tipo de aplicação considerada, a câmera tende a seguir o objeto, como ocorre por exemplo em uma transmissão de corrida. Supõe-se então que se um ponto obtido pelo SIFT tiver correspondências de maneira consistente ao longo dos quadros, a tendência é que ele pertença a um objeto. Assim, o algoritmo vai calculando a probabilidade dos pixels pertencerem a um objeto.

O método proposto em [18] realiza a detecção de objetos abandonados em uma estrada a partir de uma câmera colocada em um veículo em movimento. Para tanto, utiliza de um vídeo de referência sem objetos, que serve como comparação a qualquer vídeo contendo objetos gravado no mesmo percurso. Para os vídeos serem comparados adequadamente, eles são sincronizados com base em um GPS contido no veículo, e um alinhamento espacial é realizado com o cálculo de uma homografia. Por fim, falsos alarmes são eliminados por uma detecção temporal. Devido à natureza do problema, também exclui-se a possibilidade de detecção de objetos em uma posição muito acima do plano da estrada.

De Carvalho *et al.* [19] expande as idéias mostradas em [18], empregando-as em uma câmera acoplada a um robô que percorre um ambiente industrial em um movimento retilíneo. Um vídeo com objetos abandonados é sincronizado com um vídeo previamente validado utilizando as mudanças de direção do movimento do robô observadas nas imagens, o que dispensa a ajuda de sinais externos. Os quadros correspondentes são alinhados por uma homografia, e aplica-se a correlação cruzada normalizada (NCC) em uma janela centrada em cada píxel da imagem, criando uma máscara com candidatos a objetos detectados. Por fim, emprega-se uma filtragem temporal e uma votação na máscara obtida para melhorar a robustez. Florentin *et al.* [20] complementam este trabalho ao realizar um estudo de 4 dos principais descritores de imagem (SIFT, SURF [21], BRISK [22], e FREAK [23]) aplicados ao

contexto de detecção de objetos, definindo aquele que apresenta melhor desempenho e propondo uma modificação para restringir o número de correspondências erradas.

2.2 Alinhamento

Devido à facilidade de implementação, diversas técnicas encontradas na literatura para alinhamento de vídeos utilizam câmeras colocadas em um veículo. Em [24] é proposta uma técnica de alinhamento e registro espacial que se baseia principalmente em intensidade de pixels para vídeos gravados em um veículo em uma mesma trajetória. O diferencial do algoritmo é que o alinhamento é feito sem necessidade de rastreamento de *features* conhecidas, o que impossibilita o cálculo de entidades como a matriz fundamental, e o registro não fornece um mapeamento puramente linear entre as imagens.

O trabalho proposto por Evangelidis e Bauckhage [25] possui similaridades com o anterior, sendo mais flexível por exigir somente que a trajetória seja a mesma em cada vídeo, não necessariamente a velocidade, permitindo inclusive o alinhamento de vídeos em direções distintas. Já em [26], os autores consideram que um dos vídeos está inteiramente disponível, pois é a referência, podendo ser processado e indexado. Desse modo, são feitas *queries* à base de dados de quadros para alinhar cada quadro de um novo vídeo adquirido. Para melhorar a robustez, as *queries* são realizadas em diferentes escalas, cada qual gerando uma correspondência, e um algoritmo chamado ECC [27] é usado para refinamento dos resultados.

Em [28] o alinhamento é realizado a partir das informações de um GPS, de maneira similar a [18]. O estudo propõe uma nova abordagem para o registro espacial que demanda menor complexidade computacional. O método de registro utiliza um descritor gerado a partir do gradiente de orientação dos pixels. Para verificar a correspondência entre os quadros, realiza-se uma busca através dos descritores de cada quadro, considerando também possíveis deslocamentos horizontais e verticais.

O artigo [29] propõe um alinhamento espaço-temporal com base em MRF (*Markov random field*) [30]. Utiliza o SIFT para buscar pontos correspondentes entre os quadros dos dois vídeos, seguida pela aplicação de uma homografia entre os quadros candidatos ao alinhamento, para então ser atribuída uma nota para cada par de quadros. A melhor correspondência é aquela cujo par possuir a melhor nota, considerando também que, para esta nota não ser um *outlier*, devemos ter uma sequência de pares de quadros consecutivos nos dois vídeos com uma boa nota.

2.3 Estrutura da Cena

A reconstrução 3D de uma cena é um assunto bem estabelecido dentro da área de visão computacional, tendo atingido soluções em tempo real [31] e comerciais [32]. As soluções têm em comum o fato de usar grande parte do arcabouço do algoritmo SfM (*Structure from Motion*) [33], que será descrito no Capítulo 4.

O trabalho [31] usa o SfM para reconstruir pontos 3D pertencentes à cena e a posição da câmera em cada quadro, com auxílio de uma rotina de *bundle adjustment* [34]. Também realiza um mapeamento de textura para melhor visualização. Entretanto, objetos distantes das superfícies das fachadas dos prédios ou do chão, como carros ou placas, não são bem representados. Em [35], um algoritmo de reconhecimento substitui os trechos com carros por modelos artificiais para melhorar a experiência visual.

Em [32] é proposto um sistema para navegação em uma cena a partir de um conjunto de imagens. O algoritmo SfM é empregado para encontrar as posições e orientações de cada câmera utilizada na captura das imagens. Como não é possível obter a posição absoluta das câmeras a partir de imagens devido a ambiguidades geométricas [15], o algoritmo utiliza alguma referência geográfica como imagens de satélites ou mapas de elevação para alinhar os modelos de cena e trajetória obtidos com as informações do ambiente. A seguir, também aplica uma renderização com interpolação entre vistas para permitir uma navegação suave na cena.

O acúmulo de erros pode impossibilitar o uso de um algoritmo sequencial para obtenção da trajetória da câmera em percursos fechados. Em [36], esse erro é reduzido ao permitir a fusão de reconstruções oriundas de vídeos diferentes, desde que exista alguma superposição. Já em [37], supõe-se que uma volta começa e termina no mesmo ponto. Cada imagem adquirida é associada a uma posição no espaço, e utiliza-se de similaridade de imagens para determinar se uma nova imagem representa um local já visitado.

A reconstrução também permite a criação de uma imagem estilo panorâmica em cenas onde ocorrem movimentos mais longos, em que as técnicas tradicionais de panorama, que assumem um centro de câmera fixo, não são eficazes. Agarwala *et al.* propõe em [38] a criação de uma imagem panorâmica centrada na fachada de prédios a partir de um vídeo capturado por uma câmera posicionada em um carro.

Inicialmente o algoritmo SfM é utilizado para recuperar as posições e orientações da câmera em cada quadro. Define-se então uma superfície tridimensional, de forma automática ou interativa, que deve ser o plano dominante na imagem. Cada quadro é reprojetado na superfície, que é amostrada, gerando os pixels que compõem a imagem panorâmica. Por fim, uma otimização MRF é aplicada no processo de escolha do píxel caso haja sobreposição, considerando três fatores:

- um píxel deve ser melhor representado pela imagem vinda da câmera mais próxima e que aponte para ele de forma mais perpendicular; e
- deve haver uma transição suave entre as regiões da imagem montada; e
- podem ocorrer imperfeições devido a variações de luminosidade ou oclusão, de modo que a imagem criada deve ser bem fiel à cena nos trechos onde os pontos no espaço estão próximos do plano que recebe a projeção.

Com uma proposta similar, o trabalho [39] se propõe a criar uma panorâmica das fachadas de prédios em locais do Japão destruídos por uma *tsunami*. A partir da trajetória da câmera e dos pontos que compõe a estrutura da cena obtidos pelo SfM, o algoritmo calcula uma superfície que aproxima a fachada dos prédios e outra que representa o chão, incluindo a possibilidade de inclusão de uma nova estrutura para compor outros objetos. Realiza-se um ajuste de uma curva suave que se adapta à trajetória obtida, e as imagens são projetadas nas superfícies consideradas. Para montar o panorama, escolhe-se para cada píxel se o mesmo pertence ao chão, à fachada de prédios ou a outros objetos, com uma otimizaçao por MRF.

2.4 *Simultaneous Localization and Mapping* (SLAM)

O SLAM remete ao problema da estimação do movimento de um robô simultaneamente ao mapeamento do ambiente ao redor com base em qualquer tipo de sensor presente, como laser [40], radar [41] ou um Kinect ™[42]. A vertente do SLAM que utiliza câmeras como sensores primários é denominada *visual* SLAM ou *monocular* SLAM, e por vezes se assemelha à estrutura do SfM. A principal diferença reside no fato de que o primeiro é voltado para situações em que o procesamento deve ser sequencial, ao passo que o último lida com o problema de forma genérica.

O trabalho [43] apresenta um algoritmo para sistemas de baixo custo, com uma câmera posicionada em um robo aéreo e apontada para baixo, além de um sensor de altitude. Utiliza uma formulação baseada em grafos, onde cada nó é representado pela posição da câmera. Para cada nova imagem, são encontradas correspondências entre os pontos no mapa calculado e suas projeções na imagem, para então computar a transformação ocorrida na câmera, com o auxílio do RANSAC.

Outro estudo voltado para aplicações de baixo custo é visto em [44], para uma câmera acoplada a um robô limpador. Diversas otimizações são implementadas apoiadas no fato de que a aplicação é *indoor* e a câmera está apontada para cima. O algoritmo para cálculo da orientação do robô detecta retas ortogonais na imagem para auxiliar na localização, excluindo as que forem verticais, que se cruzam no ponto

de fuga. Além disso, considera que não existe variação de escala entre as imagens ao encontrar correspondências de *keypoints*. Por fim, também simplifica as técnicas de grafos presentes no SLAM, motivado pelo fato de que para a configuração proposta sempre deve ser possível encontrar correspondências entre imagens adjacentes, em virtude de sua simplicidade.

Em [45] realiza-se um estudo das formas de mapear um ambiente em 3 dimensões: com grid de *voxel* ou com representações de *keyframe* baseadas nas imagens. Propõe-se então uma forma híbrida que combina as vantagens de cada forma.

Davison [46, 47], por sua vez, trata do problema do SLAM a partir de uma única câmera. As velocidades linear e angular da câmera são estimadas a cada novo quadro, considerando que a cada intervalo de tempo pode ocorrer uma aceleração aleatória que gera variações na velocidade. Algumas *features* visuais são mapeadas, e a comparação entre a posição estimada pelo modelo e a encontrada pelas imagens fornece informação sobre a posicão real da câmera.

A proposta vista em [48] mostra um SLAM orientado a objetos. Através de algoritmos de reconhecimento, identifica objetos na cena que são utilizados como as *features* visuais rastreadas. Devido às limitações inerentes aos algoritmos de reconhecimento, este método é mais indicado para o ambientes fechados com elementos repetitivos.

Capítulo 3

Geometria de Múltiplas Vistas

A relação geométrica existente entre vistas de uma cena ou entre as imagens e o espaço tridimensional é um assunto estabelecido na literatura, sendo apresentado em livros como os de Hartley e Zisserman [15] ou Faugeras, Luong e Papadopoulo [49]. Este capítulo apresenta os conceitos básicos referentes às relações geométricas entre as entidades citadas. A Seção 3.1 fala da forma de representação dos pontos em coordenadas homogêneas. A Seção 3.2 mostra a formulação de um modelo matemático para a projeção realizada por uma câmera. Já a Seção 3.3 trata da geometria existente entre duas vistas, representada na forma da matriz fundamental, enquanto a Seção 3.4 aborda alguns algoritmos para o cálculo dessa matriz.

3.1 Coordenadas Homogêneas

Tradicionalmente, um ponto em um espaço $\mathbb{R}^2$ é representado por um vetor $(x, y)^T$. Por sua vez, uma linha pode ser considerada o conjunto de pontos que obedecem à relação $ax + by + c = 0$. Desse modo, uma forma de representação de uma reta é $(a, b, c)^T$.

Utilizando estas definições, um ponto $(x, y)^T$ pertencerá à reta $\mathbf{l} = (a, b, c)^T$ caso ele obedeça à equação $ax + by + c = 0$, que pode ser escrita na forma de um produto interno entre vetores representando o ponto e a reta, da forma:

$$(x, y, 1)(a, b, c)^T = \mathbf{x}^T\mathbf{l} = 0. \tag{3.1}$$

O vetor $(x, y, 1)^T$ é uma extensão da representação de um ponto em $\mathbb{R}^2$ que permiteo utras formas de manipulação matemática, e diz-se nesse caso que o vetor está em coordenadas homogêneas. Deve-se notar que qualquer vetor de formato $(kx, ky, k)^T$ irá satisfazer a condição exibida na Equação (3.1), de modo que esse conjunto de vetores representa, em coordenadas homogêneas, o mesmo ponto $(x, y)^T$. Como o fator k pode ser arbitrário, costuma-se representá-lo com o valor unitário.

A representação em coordenadas homogêneas também permite uma forma simples para expressar pontos no infinito. Sejam as retas paralelas $\mathbf{l_1} = (a, b, c)^T$ e $\mathbf{l_2} = (a, b, c')^T$. O ponto de interseção entre duas retas pode ser encontrado a partir da expressão [15]:

$$\mathbf{x} = \mathbf{l_1} \times \mathbf{l_2} = (c - c')(b, -a, 0)^T. \tag{3.2}$$

Se tentássemos passar o ponto $\mathbf{x}$ para coordenadas homogêneas, teríamos $\mathbf{x} = (b/0, -a/0, 1)^T$, o que é uma impossibilidade matemática. Entretanto, sabe-se que este ponto foi originado da interseção entre duas retas paralelas, que se encontram no infinito. Desse modo, o vetor $(b, -a, 0)^T$ representa um ponto no infinito.

3.2 Modelo de Câmera

Uma camera é um dispositivo que vai mapear os pontos no espaço em um plano de projeção que compõe uma imagem. O modelo matemático mais básico para o mapeamento dos pontos é chamada camera *pinhole*, que pode ser visto na Figura 3.1. Por convenção, considera-se que o plano de imagem fica entre o centro de câmera e o espaço que será projetado.

A projeção ocorre quando um um raio de luz passa pelo ponto no espaço e cruza o plano da imagem em direção ao centro de camera, formando o ponto da imagem. Considerando o centro de câmera na origem, o ponto no espaço $\mathbf{X} = (X, Y, Z, 1)^T$ e o ponto na imagem $\mathbf{x} = (x, y, 1)^T$, e sabendo que o plano de projeção é descrito pela equação $Z = f$, a equação que descreve a projeção se torna:

$$\mathbf{x} = \mathbf{PX} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} fX/Z \\ fY/Z \\ 1 \end{bmatrix}, \tag{3.3}$$

onde a matriz que representa a transformação aplicada é chamada de matriz de câmera.

Na Figura 3.1 o eixo de coordenadas do plano considera a origem como o ponto de projeção do centro de câmera, chamado de ponto principal. Entretanto, na prática as imagens costumam ser retratadas com a origem nos extremos, de modo que os valores das coordenadas sejam sempre positivos, como os índices de uma matriz. Assim, o valor da posição do ponto principal deve ser compensado na Equação (3.3):

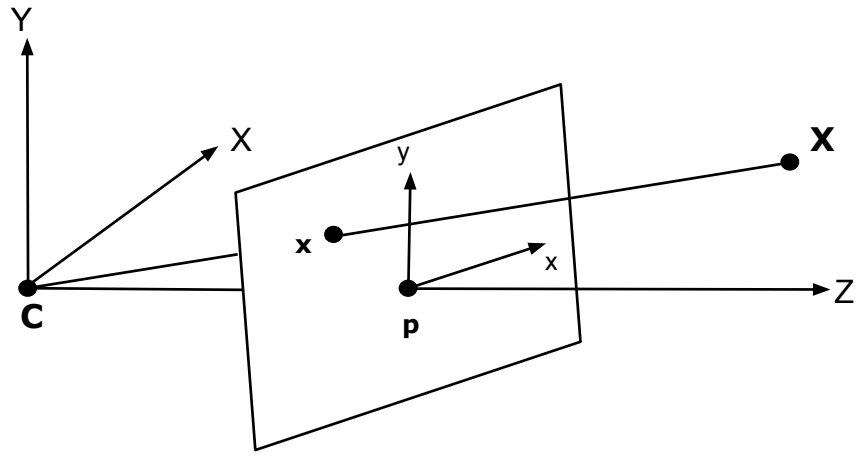


Figura 3.1: Projeção em um plano de imagem para uma câmera do tipo *pinhole*. O ponto no espaço $\mathbf{X}$ é projetado para o ponto na imagem $\mathbf{x}$ através do raio que cruza o centro de câmera $\mathbf{C}$.

$$\mathbf{x} = \begin{bmatrix} fX/Z + p_x \\ fY/Z + p_y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{3.4}$$

Caso estejamos trabalhando com mais de uma câmera, ou já exista um espaço euclidiano pré-determinado, o modelo de câmera apontado na Equação (3.4) deve ser alterado para considerar as coordenadas reais do espaço. Isto é feito aplicando-se uma transformação no ponto $\mathbf{X}$ que reflete a translação e a rotação que levam o sistema de coordenadas do espaço a coincidir com um sistema de coordenadas centrado na câmera. Assim, o novo modelo se torna:

$$\mathbf{x} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \mathbf{X} = \mathbf{K} \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix} \mathbf{X} = \mathbf{P}\mathbf{X}. \tag{3.5}$$

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

Os termos $\mathbf{R}$ e $\mathbf{t}$ representam, respectivamente, a orientação e a posição da câmera em relação às coordenadas do espaço, e são chamados de parâmetros extrínsecos. Já a matriz $\mathbf{K}$, chamada de matriz de calibração, guarda os parâmetros intrínsecos da câmera, que estão relacionados à forma como a câmera realiza a projeção.

Também é possível se obter a posição do centro de câmera $\mathbf{C}$, dada a matriz de projeção $\mathbf{P}$. Supondo os pontos $\mathbf{A}$ e $\mathbf{C}$, sendo que $\mathbf{C}$ possui a propriedade $\mathbf{PC} = 0$, um ponto qualquer na reta que liga $\mathbf{A}$ e $\mathbf{C}$ será da forma:

$$\mathbf{X} = \lambda\mathbf{A} + (1 - \lambda)\mathbf{C}. \tag{3.7}$$

A projeção do ponto $\mathbf{X}$ na imagem será:

$$\mathbf{x} = \mathbf{PX} = \lambda\mathbf{PA} + (1 - \lambda)\mathbf{PC} = \lambda\mathbf{PA}. \tag{3.8}$$

Percebe-se que qualquer ponto $\mathbf{X}$ pertencente à reta considerada possuirá a mesma projeção no ponto $\mathbf{x}$, ou seja, esta reta já representa um raio de projeção, passando pelo centro de câmera. Desse modo, como o ponto $\mathbf{A}$ não possui qualquer restrição, intui-se que o ponto $\mathbf{C}$ que obedece a $\mathbf{PC} = 0$ representa o centro de câmera.

Um modelo mais genérico para uma câmera considera ainda que outros efeitos podem existir. Em câmeras do tipo CCD, considera-se que um píxel em uma imagem pode não ser quadrado, o que equivale à existência de focos diferentes nas direções x e y. Também pode existir um fator s relacionado ao *skew*, que ocorre quando os eixos x e y em uma imagem não estão perpendiculares, o que pode acontecer em casos específicos, como quando se tira a foto de uma foto. O formato geral para a matriz de calibração é:

$$\mathbf{K} = \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.9}$$

3.3 Geometria Epipolar

Se duas imagens forem capturadas da mesma cena em posições diferentes, existe uma geometria intrínseca entre elas. Como pode ser visto na Figura 3.2, há uma restrição para a posição dos pontos em cada imagem que são relacionados a um mesmo ponto no espaço. Esta restrição não depende da cena em si, e sim da forma como foi feita a projeção em cada imagem.

Considerando que somente o ponto $\mathbf{x}$ na imagem da esquerda é conhecido, deseja-se encontrar as restrições para as posições do ponto no espaço e do seu equivalente na imagem da direita. Como mostra a Figura 3.2, o ponto $\mathbf{X}$ no espaço se projeta à imagem da esquerda a partir de um raio de projeção que liga $\mathbf{X}$ e o centro de câmera $\mathbf{C}$, sendo o ponto na imagem aquele onde o raio cruza o plano da imagem. Entretanto, qualquer ponto pertencente a essa reta pode ter gerado o ponto $\mathbf{x}$, pois

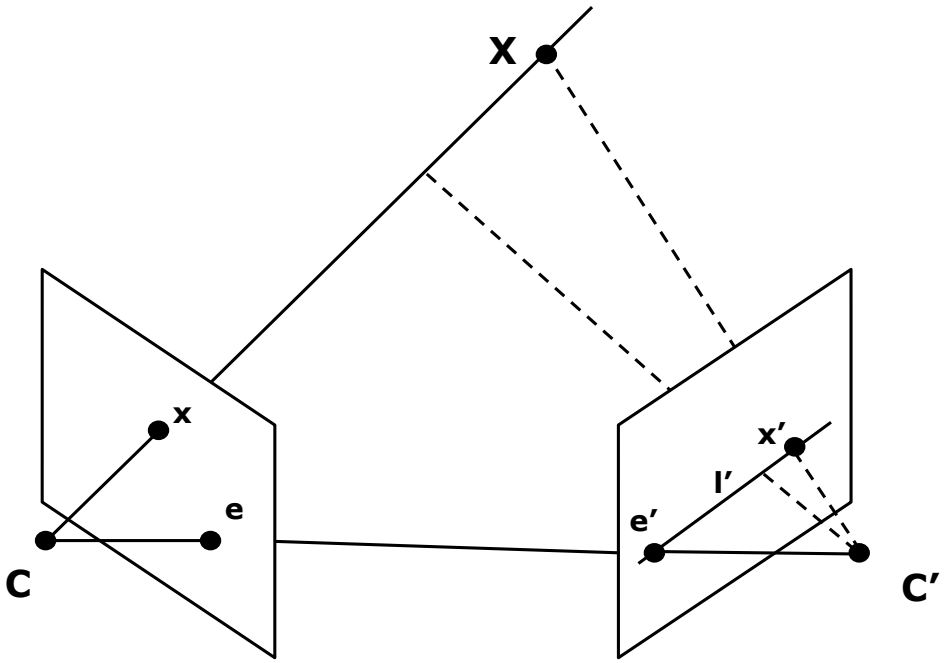


Figura 3.2: Geometria que relaciona vistas diferentes de uma mesma cena.

o raio de projeção seria o mesmo. Assim, o conhecimento de um ponto em uma imagem restringe o ponto equivalente no espaço a pertencer à reta que cruza o ponto na imagem e o centro de câmera.

Por sua vez, o ponto $\mathbf{x}'$ na imagem da direita também é a projeção deste mesmo ponto no espaço. Uma vez que a posição para o ponto no espaço foi limitada a pertencer a uma reta, o ponto $\mathbf{x}'$ deve pertencer à projeção dessa reta. Ou seja, dada uma posição de um ponto $\mathbf{x}$ em em uma imagem, o ponto equivalente $\mathbf{x}'$ na outra imagem deve pertencer a uma reta, que é denominada linha epipolar.

Como todos os raios de projeção de uma imagem passam pelo centro de câmera, todas as linhas epipolares na imagem da direita devem passar pela projeção de $\mathbf{C}$ na mesma, que recebe o nome de epipolo. A linha que liga os dois centros de câmera e determina a posição dos respectivos epipolos é chamada de *baseline*

3.3.1 Matriz Fundamental

Como visto na Figura 3.2, dado um ponto em uma vista, todos os possíveis pontos equivalentes na outra vista pertencem a uma reta que contém a imagem do centro de câmera. A matriz fundamental é a entidade geométrica que simboliza essa relação existente entre as imagens.

Considerando duas vistas distintas, com matrizes de câmera $\mathbf{P}$ e $\mathbf{P}'$, o conjunto de pontos no espaço que são projetados no ponto $\mathbf{x}$ para a primeira vista obedecem, em coordenadas homogêneas, à relação $\mathbf{x} = \mathbf{PX}$. Esta reta pode ser determinada por

dois pontos: o centro de câmera, dado por $\mathbf{PC} = 0$, e um ponto distinto qualquer. Pelo desenvolvimento de Xu and Zhang [50], um segundo ponto pode ser obtido pela pseudo-inversa de $\mathbf{P}$, fazendo $\mathbf{X}^+ = \mathbf{P}^+\mathbf{x}$, e o raio de projeção obedece à equação:

$$\mathbf{X}(\lambda) = \mathbf{X}^+ + \lambda\mathbf{C}. \tag{3.10}$$

Na segunda vista, a linha epipolar equivalente ao ponto $\mathbf{x}$ é a projeção da reta definida na Equação (3.10). Desse modo, as projeções dos dois pontos conhecidos, $\mathbf{X}^+$ e $\mathbf{C}$, irão pertencer à linha epipolar, e podem ser usados para parametrizá-la. A imagem dos dois pontos será:

$$\mathbf{e}' = \mathbf{P}'\mathbf{C}. \tag{3.11}$$

$$\mathbf{x}^+ = \mathbf{P}'\mathbf{X}^+ = \mathbf{P}'\mathbf{P}^+\mathbf{x}. \tag{3.12}$$

Com base no ponto $\mathbf{x}^+$ e no epipolo $\mathbf{e}'$, o vetor que representa a linha epipolar pode ser escrito por [15]:

$$\mathbf{l}' = \mathbf{e}' \times \mathbf{x}^+ = [\mathbf{e}']_\times \mathbf{P}'\mathbf{P}^+\mathbf{x} = \mathbf{F}\mathbf{x}. \tag{3.13}$$

$$\mathbf{F} = [\mathbf{e}']_\times \mathbf{P}'\mathbf{P}^+. \tag{3.14}$$

onde $[\mathbf{e}']_\times$ é uma matrix antissimétrica gerada a partir de $\mathbf{e}'$ tal que $[\mathbf{e}']_\times\mathbf{P}' = \mathbf{e}' \times \mathbf{P}'$. A matriz $\mathbf{F}$ é chamada de matriz fundamental, e estabelece uma relação entre vistas diferentes de uma mesma cena, pois fornece um mapeamento de um ponto $\mathbf{x}$ em uma vista para a linha epipolar $\mathbf{l}'$ na outra.

Uma propriedade importante da matriz fundamental é a que relaciona pontos correspondentes entre duas imagens. Supondo que foram obtidos os pontos $\mathbf{x}$ e $\mathbf{x}'$ em duas imagens que são projeções de um mesmo ponto. A Equação (3.13) mostra que o ponto $\mathbf{x}$ define uma reta $\mathbf{l}'$ na outra imagem. Como o ponto $\mathbf{x}'$ é o equivalente nesta imagem, ele deve pertencer à reta obtida. Desse modo:

$$0 = \mathbf{x}'^{\mathbf{T}}\mathbf{l}' = \mathbf{x}'^{T}\mathbf{F}\mathbf{x} \tag{3.15}$$

Outra propriedade diz respeito ao número de graus de liberdade das componentes de $\mathbf{F}$. Como mostra a Equação 3.14, esta matriz é expressa por um produto contendo um fator $[\mathbf{e}']_\times$ antissimétrico, o que também a torna antissimétrica, e assim $det(\mathbf{F}) = 0$. Além disso, por ser uma matriz 3×3 em coordenadas homogêneas, somente 8 componentes são linearmente independentes, visto que a escala é irrelevante. Estas duas restrições levam $\mathbf{F}$ a possuir $9 - 2 = 7$ graus de liberdade.

3.3.2 Matriz Essencial

A matriz essencial é um caso particular da matriz fundamental quando se trabalha com coordenadas normalizadas, que são aquelas onde foi compensado o efeito da matriz de calibração. Ela apresenta algumas outras propriedades que, entre outras coisas, facilitam a reconstrução de um par de câmeras.

Uma vez que a matriz de câmera é dada pela expressão $P = \mathbf{K}\left[\mathbf{R} \quad | \quad \mathbf{t}\right]\mathbf{X}$, a matriz de câmera e o ponto na imagem em coordenadas normalizadas serão:

$$\hat{\mathbf{P}} = \mathbf{K}^{-1}\mathbf{P} = \left[\mathbf{R} \quad | \quad \mathbf{t}\right]. \tag{3.16}$$

$$\hat{\mathbf{x}} = \mathbf{K}^{-1}\mathbf{P}\mathbf{X} = \mathbf{K}^{-1}\mathbf{x}. \tag{3.17}$$

Dado um par de câmeras normalizadas, $\hat{\mathbf{P}} = \left[\mathbf{I} \quad | \quad \mathbf{0}\right]$ e $\hat{\mathbf{P}'} = \left[\mathbf{R} \quad | \quad \mathbf{t}\right]$, a matriz essencial que representa a relação entre as duas câmeras é dada pela seguinte expressão, a partir do desenvolvimento da Equação (3.14):

$$\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R} = \mathbf{R}[\mathbf{R^T t}]_{\times}. \tag{3.18}$$

Esta equação mostra que a matriz essencial depende unicamente da rotação e translação existente entre uma câmera e outra. Desse modo, ao contrário da matriz fundamental, ela possui somente 5 graus de liberdade, sendo 3 da translação e 3 da rotação, com 1 descontado por estar em coordenadas homogêneas.

De maneira similar à Equação (3.15), também existe uma relação entre os pontos correspondentes normalizados:

$$\hat{\mathbf{x}'}^{\mathbf{T}}\mathbf{E}\hat{\mathbf{x}} = 0. \tag{3.19}$$

Substituindo-se (3.17) em (3.15), encontra-se a expressão que permite a conversão da matriz fundamental na essencial, dada a matriz de calibração:

$$\mathbf{E} = \mathbf{K'^T}\mathbf{F}\mathbf{K}. \tag{3.20}$$

Uma matriz essencial possui uma propriedade adicional, que diz que além de possuir um autovalor nulo, os autovalores restantes são iguais. Desse modo, ela obedece à seguinte equação [51]:

$$\mathbf{E}\mathbf{E}^T\mathbf{E} - \frac{1}{2}trace(\mathbf{E}\mathbf{E}^T)\mathbf{E} = 0. \tag{3.21}$$

3.4 Cálculo da Matriz Fundamental

A Equação (3.14) mostra como obter a matriz fundamental que relaciona duas vistas a partir das respectivas matrizes de câmera. No entanto, é comum que se possua somente as imagens de cada vista e, a partir delas, deseja-se estimar alguma informação sobre a geometria das câmeras e da cena. Neste caso, o cálculo se baseia na Equação (3.15), que impõe uma restrição na matriz fundamental quando se conhece um par de pontos correspondentes.

Algoritmos descritores de imagem, como o SIFT [17], SURF [21], BRISK [22] ou FREAK [23], detectam pontos representativos na imagem, que associados a um determinado atributo, podem caracterizar regiões da imagem. Estes algoritmos também permitem a obtenção de pontos correspondentes entre as imagens, que são aqueles que apresentam atributos similares, retratando características em comum.

A partir de duas vistas de uma mesma cena, utiliza-se de algoritmos descritores de imagem para obter possíveis pares de pontos correspondentes entre as imagens. Cada par fornece uma equação com os elementos da matriz $\mathbf{F}$, e com um número suficiente de pares é possível obter uma solução exata para a matriz fundamental. Os algoritmos a seguir diferem na quantidade de pares de pontos necessários e nas restrições que assumem para o cálculo da matriz fundamental.

3.4.1 Algoritmo de 8 Pontos

O algoritmo de 8 pontos [52] parte da expansão da Equação (3.14), tornando explícita a dependência dos elementos que compõem $\mathbf{F}$:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0.$$

$$x'xf_{11} + x'yf_{12} + xf_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0. \quad (3.22)$$

Percebe-se que um par de pontos fornece uma equação no sistema de 9 incógnitas. Como a matriz deve estar em coordenadas homogêneas, o que oferece uma restrição ao problema, 8 pares de pontos são necessários para definir unicamente uma solução.

Entretanto, o resultado obtido pode não possuir outras propriedades de uma matriz fundamental, como o fato de ela possuir posto 2. Neste caso, realiza-se a decomposição do resultado em SVD, e zera-se o autovalor de menor energia, seguido da inversão da transformação. A resposta final será a matriz fundamental válida que minimiza a distância de Froebenius [15] em relação ao resultado do sistema.

Para evitar problemas matemáticos, já que os dados de entrada podem possuir ordens de grandezas diferentes, os vetores $\mathbf{x}$ e $\mathbf{x}'$ são normalizados de forma que o

centróide dos pontos esteja na origem e o valor RMS seja $\sqrt{2}$. Na prática, também é recomendável que se possua uma quantidade muito maior do que 8 pares de pontos. Neste caso, o RANSAC [15] é aplicado na eliminação de *outliers*, e se utiliza de mínimos quadrados para resolver o sistema sobredeterminado para os pares restantes.

3.4.2 Algoritmo de 7 Pontos

O algoritmo de 7 pontos é uma extensão do anterior, considerando o caso mínimo que ainda possibilita uma resposta única, uma vez que a matriz fundamental possui 7 graus de liberdade. Utilizando 8 pares de pontos, a resolução do sistema $\mathbf{Af} = 0$ gera uma solução única, a menos da escala. Com 7 pontos, e ignorando a escala, o sistema possui como solução um espaço bi-dimensional, que pode ser descrito por:

$$\mathbf{F}(\alpha) = \alpha\mathbf{F_1} + (1-\alpha)\mathbf{F_2}. \tag{3.23}$$

Uma vez que a solução deve ser uma matriz fundamental, ela deve obedecer à restrição $det(\mathbf{F}) = 0$, e quando aplicada à Equação (3.23), fornece:

$$det(\mathbf{F}(\alpha)) = det(\alpha\mathbf{F_1} + (1-\alpha)\mathbf{F_2}) = 0. \tag{3.24}$$

que é uma equação cúbica em α. Resolvendo a equação para α, encontra-se uma ou três soluções reais. Substituindo em (3.23), são obtidas uma ou três possíveis soluções para a matriz fundamental.

3.4.3 Algoritmo de 5 Pontos

O algoritmo de 5 pontos [53] é a solução mínima para o caso em que a matriz de calibração da câmera é conhecida e propicia o cálculo direto da matriz essencial. A partir da normalização vista na Equação (3.17), monta-se, para os 5 pares de pontos correspondentes, o sistema de equações $\mathbf{\hat{A}e} = 0$ com 5 equações e 9 incógnitas, com base no desenvolvimento da Equação (3.19). A solução para este sistema de equações é um espaço que pode ser parametrizado por quatro bases vetoriais, $\mathbf{E_w}$, $\mathbf{E_x}$, $\mathbf{E_y}$ e $\mathbf{E_z}$. Uma matriz fundamental no espaço das soluções é da seguinte forma:

$$\mathbf{E}(x,y,z) = \mathbf{E_w} + x\mathbf{E_x} + y\mathbf{E_y} + z\mathbf{E_z}. \tag{3.25}$$

Para resolução da Equação (3.25), é aplicada a condição vista na Equação (3.21), que é uma condição que deve ser atendida por qualquer matriz essencial, além de assumir que $det(\mathbf{F}) = 0$. Por consequência, é montado um sistema com o seguinte vetor de incógnitas:

$$[x^3, y^3, x^2y, xy^2, x^2z, x^2.y^2z, y^2, xyz, xy, xz^2, xz, x, yz^2, yz, y, z^3, z^2, z, 1], \quad (3.26)$$

que não permite uma simples solução linear, sendo necessário algum procedimento matemático. Em [53], procura-se resolver o sistema a partir de eliminação de Gauss-Jordan, seguido de procedimentos *ad hoc*. Já [54] propõe o uso de variáveis escondidas, para isolar uma das variáveis e assim permitir o cálculo.

Em ambos os casos, o sistema gera um polinômio de décima ordem. Assim sendo, o algoritmo pode gerar até 10 soluções para a matriz essencial, de modo que é necessário alguma métrica de decisão, que para [53] é a contagem do número de pontos triangulados que ficam na frente das câmeras.

Capítulo 4

Reconstrução da Cena e do Movimento

O conhecimento da geometria que relaciona as vistas de uma mesma cena pode fornecer informação sobre a distribuição das câmeras e o posicionamento dos objetos no espaço. Entretanto, ao se dispor somente de um conjunto de imagens, não é possível recuperar as posições reais em um ambiente tridimensional, sendo a melhor solução aquela que se relaciona com a real por um fator de escalamento. Este capítulo trata dos métodos para obtenção da disposição das câmeras e dos objetos na cena a partir do conhecimento de vistas de um mesmo ambiente, e todos os problemas relacionados a essa estimação. A Seção 4.1 trata da obtenção de câmeras e reconstrução de pontos na cena a partir de duas vistas, além de mostrar as ambiguidades inerentes à reconstrução. Já a Seção 4.2 apresenta um algoritmo que adapta a reconstrução anterior para operar com várias vistas.

4.1 Reconstrução com Duas Vistas

O Capítulo 3 mostra a relação existente entre pontos no espaço, câmeras e projeções nas imagens. Entretanto, em geral possuímos somente as imagens que representam as diferentes vistas de uma cena, e algumas estimativas de pontos que são correspondentes, ou seja, que são projeções de um mesmo ponto no espaço. A partir da obtenção da matriz fundamental que relaciona as vistas, através de algoritmos como os presentes na Seção 3.4, deseja-se recuperar a informação sobre a posição das câmeras e dos pontos no espaço, que é desconhecida.

A formulação do problema é a que segue. Dado um conjunto estimado de pontos correspondentes entre as imagens, $\mathbf{x}_i$ e $\mathbf{x'}_i$, que define uma matriz fundamental $\mathbf{F}$, calcula-se a reconstrução $\{\mathbf{P}, \mathbf{P'}, \mathbf{X}_i\}$, composta das matrizes de câmera em cada vista e os pontos no espaço que geram as projeções $\mathbf{x}_i$ e $\mathbf{x'}_i$.

4.1.1 Decomposição em Câmeras

A Equação (3.14) mostra que existe uma relação direta entre a matriz fundamental e as matrizes de câmera, onde as matrizes $\mathbf{P}$ e $\mathbf{P}'$ definem uma matriz $\mathbf{F}$. Além disso, partindo da Equação (3.15), encontra-se a seguinte restrição entre as matrizes:

$$0 = \mathbf{x}'^T\mathbf{F}\mathbf{x} = (\mathbf{P}'\mathbf{X})^T\mathbf{F}(\mathbf{P}\mathbf{X}) = \mathbf{X}^T(\mathbf{P}'^T\mathbf{F}\mathbf{P})\mathbf{X}, \tag{4.1}$$

que, para ser válida para todo $\mathbf{X}$, implica que $\mathbf{P}'^T\mathbf{F}\mathbf{P}$ deve ser anti-simétrica.

Desse modo, um possível resultado obtido a partir da matriz fundamental tem a forma:

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} & | & \mathbf{0} \end{bmatrix} \qquad \text{e} \qquad \mathbf{P}' = \begin{bmatrix} \mathbf{SF} & | & \mathbf{e}' \end{bmatrix}, \tag{4.2}$$

visto que

$$\begin{bmatrix} \mathbf{SF} & | & \mathbf{e}' \end{bmatrix}^T \mathbf{F} \begin{bmatrix} \mathbf{I} & | & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^T\mathbf{S}^T\mathbf{F} & \mathbf{0} \\ \mathbf{e}'^T\mathbf{F} & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{F}^T\mathbf{S}^T\mathbf{F} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}, \tag{4.3}$$

que é anti-simétrica se $\mathbf{S}$ for anti-simétrica. Uma escolha proposta por [55] para $\mathbf{S}$ é $\mathbf{S} = [\mathbf{e}']_\times$.

Entretanto, como visto em [15], existem várias formas de decomposição da matriz $\mathbf{F}$. A forma genérica para o par de matrizes de câmera obtidas a partir da matriz fundamental é:

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} & | & \mathbf{0} \end{bmatrix} \qquad \text{e} \qquad \mathbf{P}' = \begin{bmatrix} [\mathbf{e}']_\times\mathbf{F} + \mathbf{e}'\mathbf{v}^T & | & \lambda\mathbf{e}' \end{bmatrix}, \tag{4.4}$$

para qualquer vetor $\mathbf{v}$ e escalar λ.

Caso a matriz de calibração das câmeras seja conhecida, um par de câmeras pode ser obtido de uma maneira simples, através da matriz essencial. Como mostra a Equação (3.18), a matriz essencial possui uma decomposição que fornece os vetores $\mathbf{R}$ e $\mathbf{t}$, que estão relacionados às translação e rotações sofridas pela câmera entre uma vista e outra.

Considerando que a decomposição por SVD da matriz essencial é da forma $\mathbf{E} = \mathbf{U}diag(1,1,0)\mathbf{V}^T$, a fatoração de $\mathbf{E} = \mathbf{SR}$ será [15]:

$$\mathbf{S} = \mathbf{UZU^T} = [\mathbf{t}]_\times \qquad \text{e} \qquad \mathbf{R} = \mathbf{UWV^T} \text{ ou } \mathbf{UW^TV^T} \tag{4.5}$$

onde

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{e} \qquad \mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.6}$$

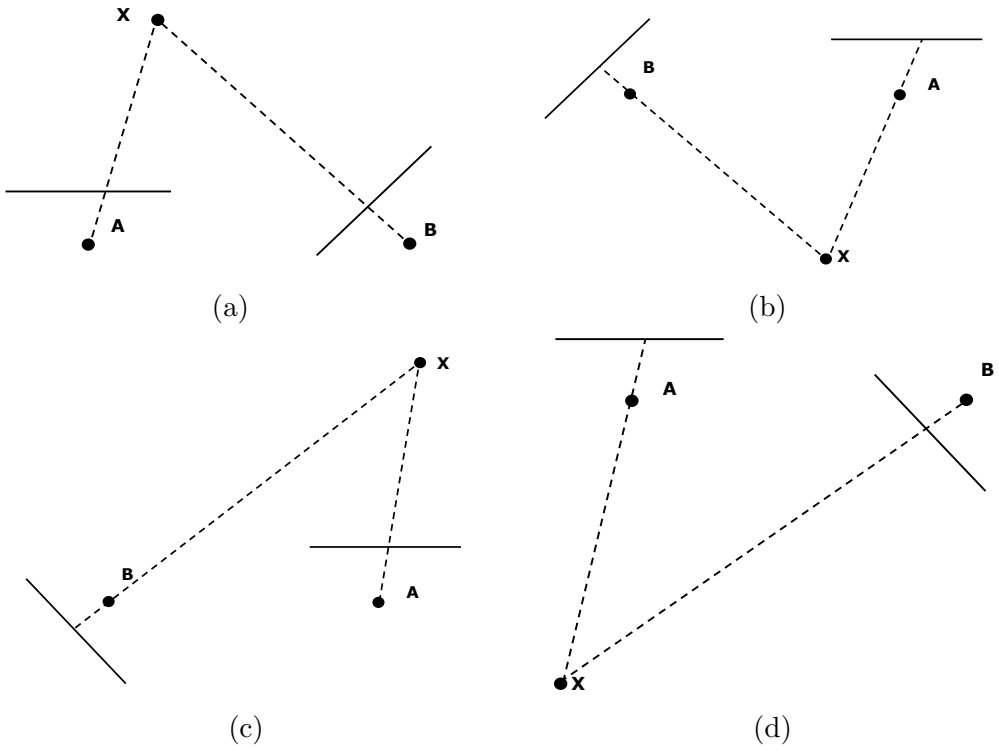


Figura 4.1: Possíveis soluções para a decomposição de câmeras a partir da matriz essencial. (a) Projeção na frente de ambas as câmeras. (b) Projeção atrás de ambas as câmeras. (c) Projeção na frente da câmera **A** e atrás da câmera **B**. (d) Projeção na frente da câmera **B** e atrás da câmera **A**.

A Equação (4.5) mostra que existem duas configurações possíveis para a matriz **R**. Além disso, a matriz **S**, pela sua composição, deve possuir norma de Frobenius igual a $\sqrt{2}$ [15]. Por consequência, o vetor **t** deve possuir norma unitária, o que mostra que a decomposição da matriz essencial em duas câmeras não permite obter a translação real da cena, mas sim uma componente normalizada.

Uma vez que a matriz **E** está em coordenadas homogêneas, também não se pode determinar o sinal da componente **t**. Considerando também os valores para a rotação, existem 4 soluções possíveis para esta decomposição, que apresentam alguma simetria, como mostra a Figura 4.1. Entretanto, somente em um dos casos (Figura 4.1(a)) a projeção ocorre com os pontos da cena na frente de ambas as câmeras. Assim, através de uma técnica de triangulação, pode-se obter a solução correta.

Em analogia à Equação (4.4), a forma genérica para o par de câmeras será

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} & | & \mathbf{0} \end{bmatrix} \qquad \text{e} \qquad \mathbf{P}' = \begin{bmatrix} \mathbf{R} & | & \lambda\mathbf{t} \end{bmatrix}, \tag{4.7}$$

para um escalar λ.

4.1.2 Triangulação

Uma vez que possuímos as matrizes $\mathbf{P}$ e $\mathbf{P}'$ representando as duas câmeras e os pontos $\mathbf{x}$ e $\mathbf{x}'$ correspondentes entre cada imagem, é possível calcular a posição do respectivo ponto no espaço tridimensional. Supondo que não existam erros, um ponto em uma imagem e sua respectiva matriz de câmera definem uma reta de projeção. A interseção das retas de projeção vai definir o ponto $\mathbf{X}$ no espaço, que é aquele que obedece às projeções $\mathbf{x} = \mathbf{PX}$ e $\mathbf{x}' = \mathbf{P}'\mathbf{X}$, como mostra a Figura 4.2.

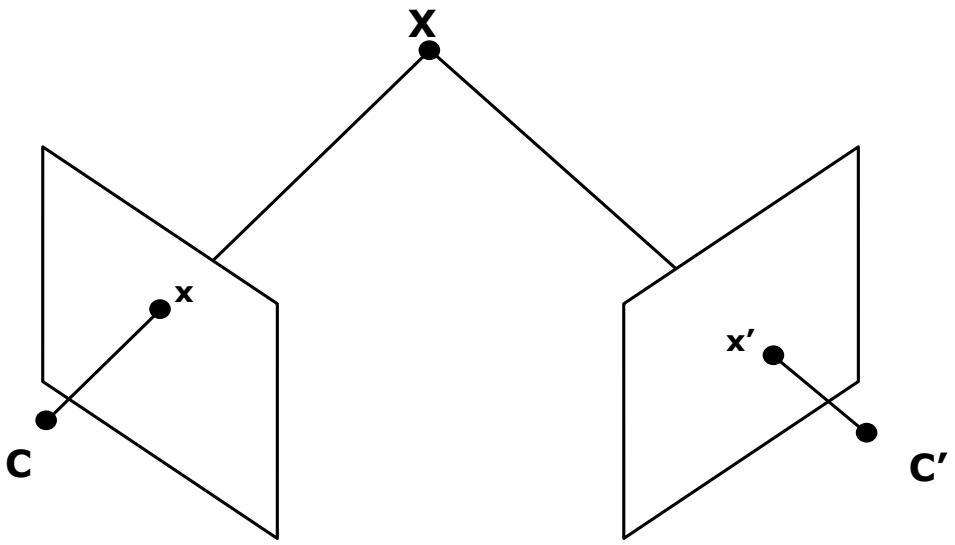


Figura 4.2: Triangulação de um ponto sem erros.

Assumindo que existam erros na obtenção dos pontos correspondentes entre as imagens, a geometria epipolar não será satisfeita e não será possível encontrar um ponto $\mathbf{X}$ que se projete em ambas as imagens. Neste caso, alguma otimização deve ser feita para obter o ponto $\mathbf{X}$ que melhor satisfaça às projeções, ou alguma melhoria na posição dos pares de pontos $\mathbf{x}$ e $\mathbf{x}'$ que obedeça à geometria epipolar entre as vistas. A Figura 4.3 exibe o erro existente na triangulação e as correções necessárias nos pontos.

O algoritmo mais simples de otimização é a triangulação linear. Uma vez que o ponto $\mathbf{X}$ deve se projetar em cada imagem, as equações $\mathbf{x} = \mathbf{PX}$ e $\mathbf{x}' = \mathbf{P}'\mathbf{X}$ podem ser combinadas para gerar um sistema de equações da forma $\mathbf{A}\hat{\mathbf{X}} = 0$, que é linear. Este método tem a vantagem de ser fácil de estender para mais vistas, visto que isto só implicaria em um número maior de equações.

De acordo com [15], o método ótimo minimiza uma função-custo que leva os pontos correspondentes a satisfazerem à geometria epipolar. Considerando os pontos $\mathbf{x}$ e $\mathbf{x}'$, pretende-se obter os pontos $\hat{\mathbf{x}}$ e $\hat{\mathbf{x}}'$ mais próximos, sujeitos à condição $\hat{\mathbf{x}}^{\mathbf{T}}\mathbf{F}\hat{\mathbf{x}}' = 0$, a partir da minimização da expressão:

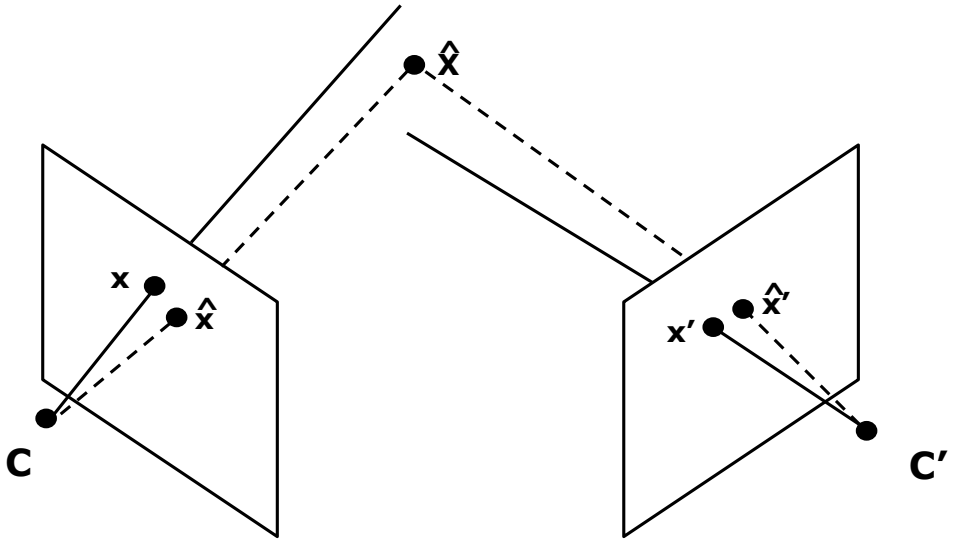


Figura 4.3: Triangulação de um ponto com erro nas posções dos pontos correspondentes. Os pontos $\hat{\mathbf{x}}$ e $\hat{\mathbf{x}}'$ representam estimativas dos pontos $\mathbf{x}$ e $\mathbf{x}'$ que obedecem à geometria epipolar, e o ponto $\hat{\mathbf{X}}$ é a triangulação resultante.

$$C = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2, \tag{4.8}$$

onde o operador $d(\mathbf{x}, \mathbf{y})$ representa o módulo da distância euclidiana entre $\mathbf{x}$ e $\mathbf{y}$.

Uma vez que os pontos corrigidos $\hat{\mathbf{x}}$ e $\hat{\mathbf{x}}'$ se situam nas respectivas linhas epipolares, a minimização também pode ser feita considerando a distância entre o ponto $\mathbf{x}$ e alguma linha epipolar $\mathbf{l}$, e analogamente para a outra vista. A geometria epipolar garante uma equivalência entre linhas $\mathbf{l}$ e $\mathbf{l}'$, de modo que ambas podem ser parametrizadas por uma variável t. Assim, a otimização consiste em encontrar o valor de t que minimiza a função-custo:

$$C = d(\mathbf{x}, \mathbf{l}(t))^2 + d(\mathbf{x}', \mathbf{l}'(t))^2, \tag{4.9}$$

que reduz-se à resolução de um polinômio em t de ordem 6.

4.1.3 Ambiguidade na Reconstrução

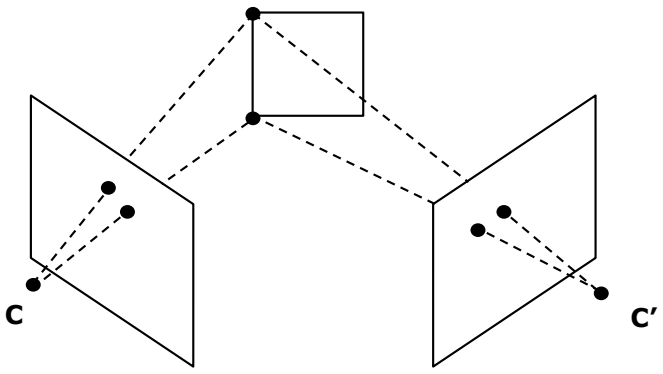
A Seção 4.1.1 mostrou que partindo de uma matriz que represente as relações geométricas entre duas vistas, seja ela a matriz fundamental ou a essencial, não é possível definir de forma unívoca um par de câmeras que represente o sistema. De fato, se não houver nenhum conhecimento sobre a real disposição dos pontos no sistema de coordenadas do espaço, não é possível recuperar a posição absoluta dos mesmos a partir das vistas, sendo a melhor solução aquela que se relaciona com a real por uma transformação de similaridade.

A partir de um conjunto de pontos $\mathbf{x_i}$ e $\mathbf{x'_i}$ entre as imagens, é possível obter uma reconstrução $\{\mathbf{P}, \mathbf{P'}, \mathbf{X}_i\}$. Entretanto, para qualquer transformação $\mathbf{H}$, se for feita a substituição $\bar{\mathbf{X}}_\mathbf{i} = \mathbf{H}\mathbf{X_i}$ e $\bar{\mathbf{P}} = \mathbf{P}\mathbf{H}^{-1}$, os pontos nas imagens serão os mesmos, visto que:

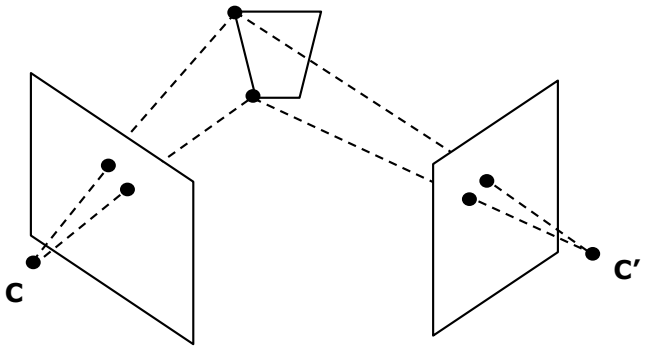
$$\bar{\mathbf{P}}\bar{\mathbf{X}}_\mathbf{i} = \mathbf{P}\mathbf{H}^{-1}\mathbf{H}\mathbf{X_i} = \mathbf{P}\mathbf{X_i} = \mathbf{x}, \tag{4.10}$$

de forma análoga para outra vista.

Percebe-se que, uma vez que só se possuam *a priori* os pontos $\mathbf{x}$ e $\mathbf{x'}$, não é possível distinguir as reconstruções $\{\mathbf{P}, \mathbf{P'}, \mathbf{X}_i\}$ e $\{\bar{\mathbf{P}}, \bar{\mathbf{P}}', \bar{\mathbf{X}}_i\}$, visto que ambas geram as mesmas relações geométricas entre as imagens. Diz-se que, se uma reconstrução difere da verdadeira por uma transformação projetiva $\mathbf{H}$, esta é uma reconstrução projetiva, como mostrado na Figura 4.4. Este resultado é compatível com a Equação (4.4), que apresenta uma decomposição em duas câmeras com vários graus de liberdade. Ao adicionar informações extras, como a identificação de retas perpendiculares na imagem, é possível aprimorar o resultado para uma reconstrução afim ou métrica.



(a)

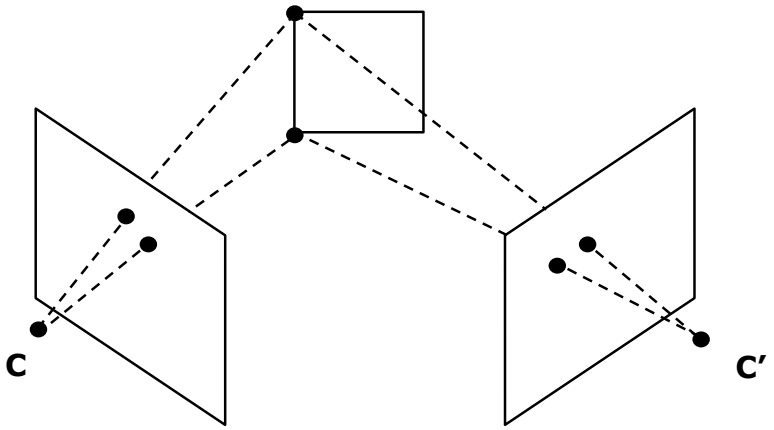


(b)

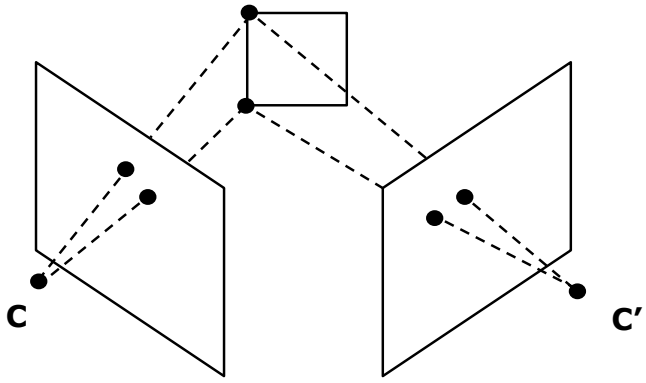
Figura 4.4: Reconstrução projetiva de uma cena. A reconstrução (b) obtida difere da reconstrução real (a) por uma transformação projetiva.

Caso a matriz de calibração seja conhecida, qualquer reconstrução deve, além de

apresentar todas as projeções nos mesmos pontos nas imagens, respeitar que os raios de projeção possuam sempre o mesmo ângulo. Neste caso, diz-se que a reconstrução é metrica, pois difere de outra reconstrução por uma transformação de similaridade, como visto na Figura 4.5. Este é o caso apresentado na Equação (4.7), visto que a segunda câmera apresenta uma rotação fixa em relação à primeira, mas não se pode determinar a distância absoluta entre elas.



(a)



(b)

Figura 4.5: Reconstrução métrica de uma cena. A reconstrução (b) obtida difere da reconstrução real (a) por uma transformação de similaridade.

4.2 Reconstrução para Múltiplas Vistas

Quando se trabalha com um número maior de vistas, a complexidade do problema aumenta consideravelmente. Os métodos vistos na Seção 4.1 podem ser estendidos para três vistas através do uso de tensores [15] em substituição à matriz fundamental, ou outras entidades geométricas para uma quantidade maior de vistas, o que implica em um problema de maior dimensionalidade. A solução nestes casos, em geral,

consiste em dividir o conjunto de entrada, realizando a reconstrução para duas vistas e depois adicionando informação de outras vistas.

O algoritmo *Structure from Motion* (SfM) oferece um método para a reconstrução de múltiplas vistas de uma cena. A partir de um par inicial de imagens, obtêm-se diversos pares de pontos correspondentes, e a geometria imposta por eles leva à obtenção de um possível par de matrizes de câmera para cada vista, que são utilizadas na triangulação de pontos no espaço. A Figura 4.6 mostra a estrutura inicial montada. Os pontos correspondentes são representados pela mesma cor, as matrizes de câmera influenciam na posição dos centros de câmera $\mathbf{C_1}$ e $\mathbf{C_2}$ e na posição e orientação do plano da imagem, e os pontos triangulados são os pontos $\mathbf{X_i}$.

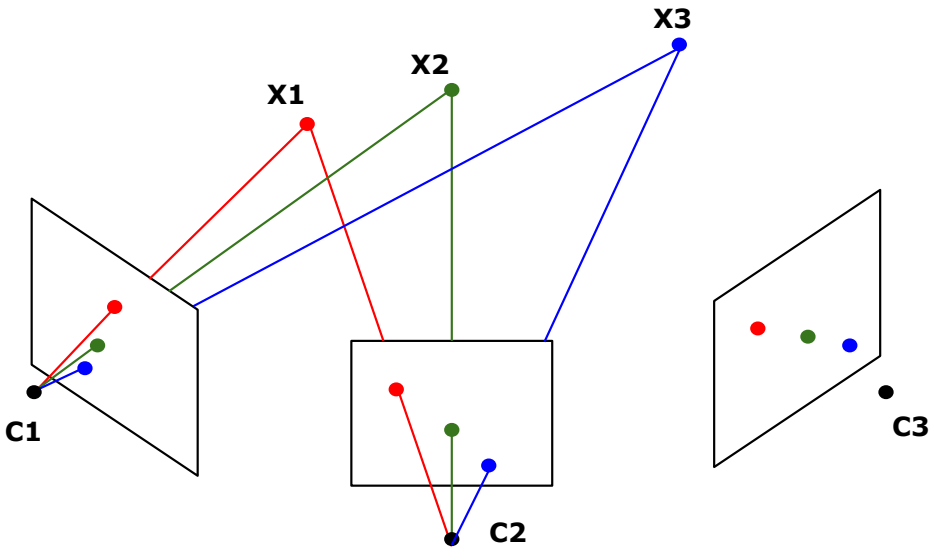


Figura 4.6: Reconstrução inicial de uma cena.

Para cada nova imagem a ser incorporada à reconstrução obtida, encontram-se inicialmente os pontos correspondentes entre a vista considerada e todas as vistas já incluídas na reconstrução. Caso algum dos pontos das vistas anteriores tenham sido utilizados na geração de um dos pontos triangulados, cria-se uma relação entre um ponto $\mathbf{X_i}$ e um ponto $\mathbf{x_i}$ da imagem atual, como pode ser visto na Figura 4.7. Com um número suficiente de correspondências entre pontos no espaço e pontos na imagem, a matriz de câmera referente à nova vista pode ser calculada a partir da Equação (3.5), que é compatível com a reconstrução obtida. Por fim, para novas correspondências que não tinham um equivalente no espaço, obtêm-se novos pontos que são incluídos na nuvem de pontos triangulados, como retratado na Figura 4.8.

Outra abordagem para o SfM envolve obter uma reconstrução independente para cada par de imagens, e depois normalizar todas para um mesmo sistema de coorde-

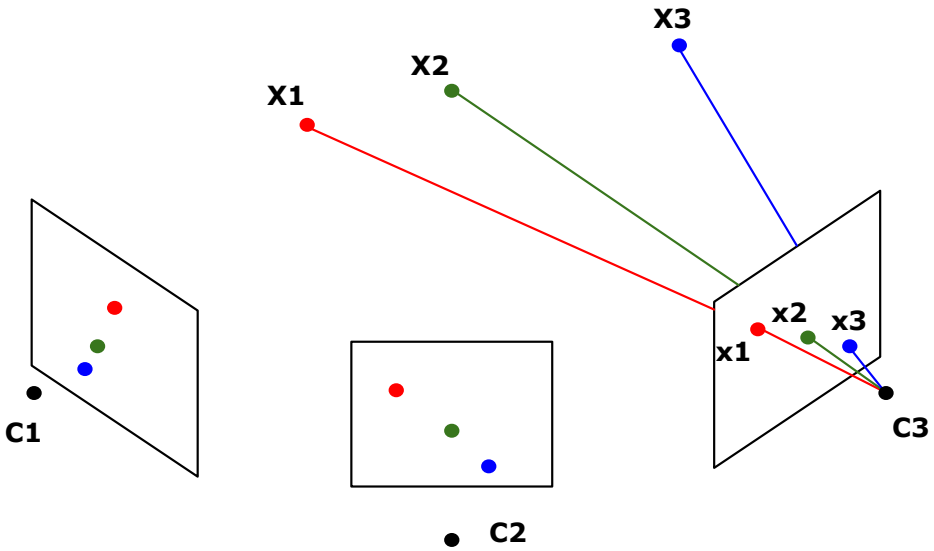


Figura 4.7: Correspondências entre os pontos $\mathbf{x_i}$ de uma nova imagem e a nuvem de pontos reconstruídos $\mathbf{X_i}$.

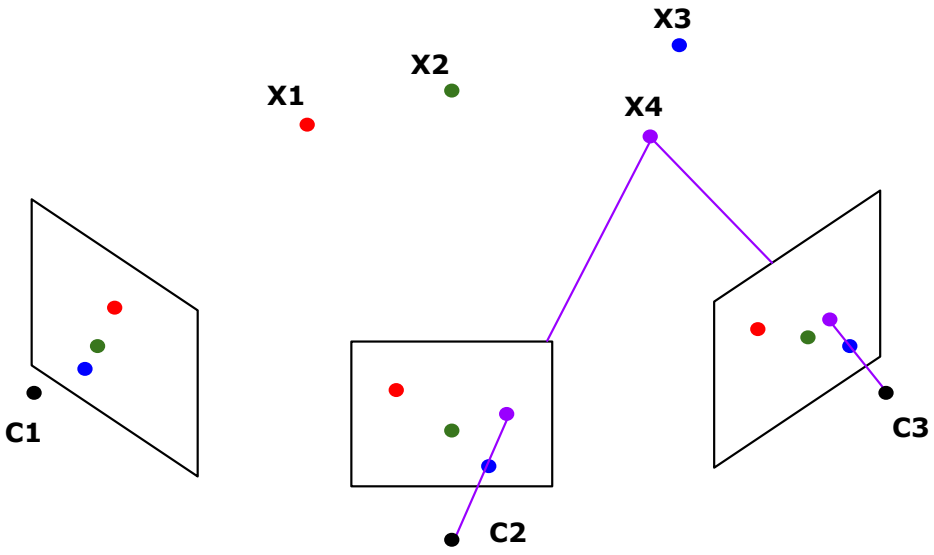


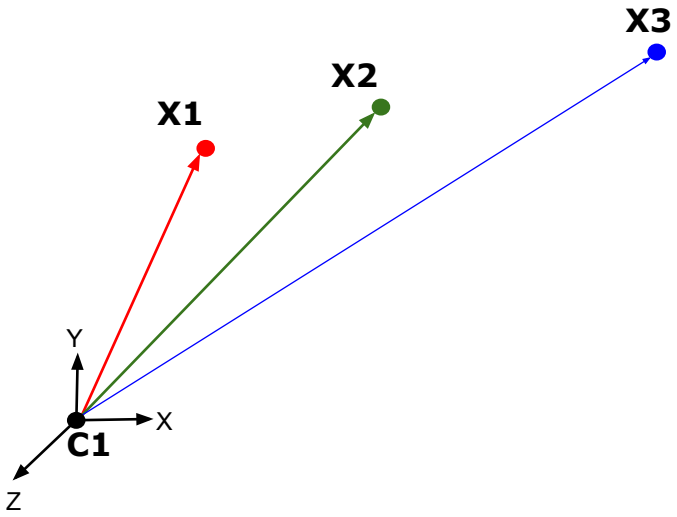
Figura 4.8: Ampliação da nuvem de pontos, com a inclusão de pontos triangulados a partir de novas correspondências.

nadas, uma vez que existe ambiguidade. Para um primeiro par de imagens, obtém-se uma matriz fundamental que vai ser utilizada na decomposição em duas câmeras $\mathbf{P_1}$ e $\mathbf{P_2}$, e posterior triangulação dos pontos $\mathbf{X_i}$, conforme a Figura 4.6. Estes pontos no espaço serão escritos em função de um sistema de coordenadas centrado em uma das câmeras.

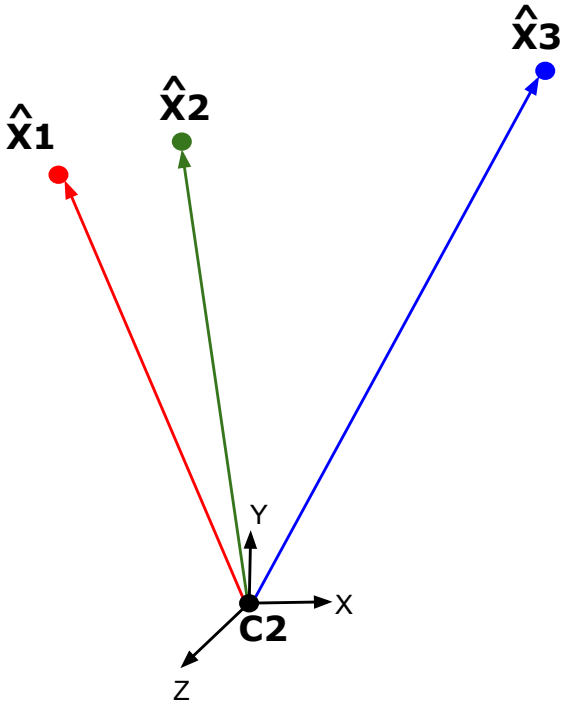
Cada nova vista forma um par com a anterior, e utiliza-se o mesmo procedimento

para reconstrução dos pontos $\hat{\mathbf{X}}_\mathbf{i}$ no espaço e obtenção das matrizes de câmera $\hat{\mathbf{P}}_\mathbf{2}$ e $\hat{\mathbf{P}}_\mathbf{3}$. Uma vez que existe ambiguidade, esta nova reconstrução apresenta pontos determinados em função de um novo sistema de coordenadas. Entretanto, através das correspondências entre as imagens, definem-se pontos $\mathbf{X}_\mathbf{i}$ que são equivalentes a pontos $\hat{\mathbf{X}}_\mathbf{i}$. As Figuras 4.9(a) e 4.9(b) exibem os pontos equivalentes, cada qual relacionado a seu eixo.

Uma vez que se deseja que todos os pontos sejam agrupados em uma reconstrução única, calcula-se a transformação $\mathbf{X}_\mathbf{i} = \mathbf{H}\hat{\mathbf{X}}_\mathbf{i}$, que vai ser responsável pela transformação no sistema de coordenadas. Para que as projeções nas imagens se mantenham as mesmas, a matriz de câmera da nova vista deve ser $\mathbf{P}_\mathbf{3} = \hat{\mathbf{P}}_\mathbf{3}\mathbf{H}$. Esta matriz representa a matriz de câmera $\hat{\mathbf{P}}_\mathbf{3}$ convertida para ser escrita em função do sistema de coordenadas utilizado na primeira reconstrução.



(a)



(b)

Figura 4.9: Pontos equivalentes em sistemas de coordenadas diferentes. Em (a), os pontos $\mathbf{X_i}$ são expressos em função de um sistema de coordenadas obtido da primeira reconstrução e centrado em $\mathbf{C_1}$. Em (b), os pontos $\mathbf{\hat{X}_i}$ são expressos em função de um sistema de coordenadas obtido da nova reconstrução e centrado em $\mathbf{C_2}$. Os pontos de mesma cor em reconstruções diferentes representam as mesmas estruturas.

Capítulo 5

Método para Estimação da Trajetória de Câmera

Um vídeo gerado por uma câmera em movimento pode ser interpretado como uma junção de diversas vistas gravadas em posições diferentes. Dessa forma, os quadros do vídeo estão sujeitos às mesmas relações geométricas existentes no Capítulo 3. Este trabalho se propõe a recuperar a trajetória de uma câmera em movimento utilizando as técnicas de reconstrução presentes no Capítulo 4, seguida da geração de uma imagem panorâmica que combine todas as vistas. A Seção 5.1 lista os procedimentos adotados para o cálculo da trajetória e montagem da panorâmica, enquanto a Seção 5.2 cita a base de dados utilizada.

5.1 Método Proposto

A trajetória de câmera foi obtida seguindo os procedimentos do algoritmo *Structure from Motion*. A partir do SURF, encontraram-se pontos correspondentes, que levaram ao cálculo da geometria epipolar entre os quadros. Com isso, foram obtidas as possíveis rotações e translações entre vistas sucessivas. Após compensar a ambiguidade existente, encontra-se a variação de posição e orientação para as vistas em relação a um mesmo eixo, que permitem a obtenção de uma imagem panorâmica. As subseções a seguir descrevem as principais etapas do algoritmo com as alterações efetuadas

5.1.1 Correspondência entre Pontos

Utilizou-se o SURF para determinar pontos salientes nas imagens, associados a descritores que permitem a obtenção de pontos correspondentes. Entretanto, o algoritmo obtém as melhores correspondências através da similaridade dos descritores, que não necessariamente representam correspondências reais da cena. Desse modo,

foram aplicados métodos de trabalhos anteriores para garantir uma melhor qualidade nos pontos, que são:

- Eliminação dos pares de pontos que possuem um ângulo com a horizontal acima de um limiar [19], visto que o movimento da câmera nos vídeos considerados é basicamente retilíneo e horizontal;
- Proibição do casamento de pontos entre escalas diferentes [20], visto que a câmera não efetua um movimento de se aproximar da cena, o que faz com que os objetos sejam sempre vistos com a mesma dimensão.

Para melhorar ainda mais a robustez, consideraram-se como pares válidos somente aqueles que mantivessem uma consistência temporal. Dadas as imagens A, B e C, para que a correspondência entre os pontos $\mathbf{x_A}$ e $\mathbf{x_B}$ nas imagens A e B seja válida, devemos ter que o ponto correspondente na imagem C para os pontos $\mathbf{x_A}$ e $\mathbf{x_B}$ deve ser o mesmo, como mostra a Figura 5.1

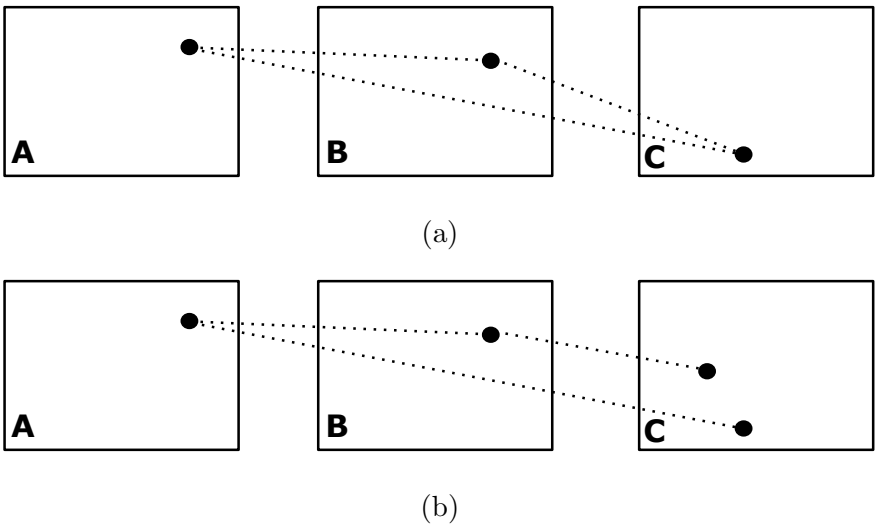


Figura 5.1: Correspondências válidas entre pontos das imagens. (a) Correspondência mantida. (b) Correspondência excluída.

5.1.2 Geometria Epipolar

Para cada par de quadros consecutivos, estima-se a geometria epipolar entre as vistas. Optou-se por utilizar o algoritmo de 5 pontos, visto que ele representa o estado-da-arte [56, 57] na obtenção da rotação e translação relativa entre as câmeras. Desse modo, também foi necessária a obtenção, de forma manual, da matriz de calibração das câmeras, através da marcação de um quadrilátero no espaço [15].

Uma vez que a câmera sofre um movimento suave, não deve haver grandes variações de orientação ou sentido do deslocamento entre uma vista e outra. Desse

modo, no processo de escolha da melhor solução para a matriz essencial, necessário ao algoritmo de 5 pontos, foi incorporada na função-custo uma parcela responsável por dar prioridade às soluções que apresentam uma nova matriz de câmera com movimento semelhante ao obtido anteriormente, através do seguinte cálculo:

$$C = ||\mathbf{R_{ant}} - \mathbf{R}||_{L_2} + \lambda_1 ||\mathbf{t_{ant}} - \mathbf{t}||_{L_2} + \lambda_2 [\%\text{ de pontos na frente das câmeras}], \quad (5.1)$$

em que $\mathbf{R_{ant}}$ e $\mathbf{t_{ant}}$ representam, respectivamente, as rotações e translações relativas obtidas pelo par de imagens anterior, $||\cdot||_{L_2}$ simboliza a norma L_2, e a porcentagem de pontos na frente das câmeras é a métrica definida em [53]. Os valores de λ_1 e λ_2 foram definidos empiricamente, de maneira a tornar a métrica de [53] responsável por obter uma solução inicial quando ainda não se tem conhecimento do movimento da cena, mas pouco influente nos outros casos.

5.1.3 Trajetória de Câmera

O cálculo da trajetória da câmera foi feito a partir de uma aproximação da segunda abordagem do SfM. dispensando a necessidade de realizar uma triangulação de pontos no espaço, que não é o foco do trabalho. A partir da matriz essencial, calculada para cada par de quadros consecutivos, extraem-se duas matrizes de câmera, dadas pela Equação (4.7). Como não se tem conhecimento sobre a posição real dos pontos no espaço, esta é uma solução relativa, onde a primeira câmera representa a origem do sistema de coordenadas e a segunda câmera é expressa por uma rotação e translação em relação à primeira.

O procedimento adotado é o seguinte. A partir do primeiro par de quadros, calcula-se a primeira câmera $\mathbf{P_1} = \begin{bmatrix} \mathbf{I} & | & \mathbf{0} \end{bmatrix}$, que passa a representar o sistema de coordenadas que vai ser utilizado na reconstrução, e uma câmera $\mathbf{P_2} = \begin{bmatrix} \mathbf{R_2} & | & \mathbf{t_2} \end{bmatrix}$, em que, nesse caso, $\mathbf{t_2}$ possui norma unitária. Os termos $\mathbf{R_2}$ e $\mathbf{t_2}$ representam, respectivamente, a rotação e a translação que levariam o eixo do sistema a ficar centrado em **P2**. Desse modo, a posição do centro de câmera da segunda vista será:

$$\mathbf{C_2} = -\mathbf{R_2 t_2}. \quad (5.2)$$

Para cada novo quadro, as matrizes de câmera obtidas devem ser escritas em função do sistema de coordenadas da primeira reconstrução. Na n-ésima vista, obtêm-se a câmera canônica $\hat{\mathbf{P}}_{\mathbf{n-1}}$ e a câmera $\hat{\mathbf{P}}_{\mathbf{n}} = \begin{bmatrix} \hat{\mathbf{R}}_{\mathbf{n}} & | & \lambda_n \hat{\mathbf{t}}_{\mathbf{n}} \end{bmatrix}$ em função da anterior. Como o procedimento é sequencial, a câmera anterior já foi obtida em função do eixo correto, da forma $\mathbf{P_{n-1}} = \begin{bmatrix} \mathbf{R_{n-1}} & | & \mathbf{t_{n-1}} \end{bmatrix}$. Temos então a seguinte

rotação para a n-ésima câmera:

$$\mathbf{R_n} = \mathbf{R_{n-1}}\hat{\mathbf{R}}_\mathbf{n}. \tag{5.3}$$

Já para a translação, também deve ser obtido o λ_n correto, visto que, com a decomposição da matriz essencial, só é possível obter um vetor de $\hat{\mathbf{t}}_\mathbf{n}$ de norma unitária. Para encontrar o valor de λ_n, considera-se a geometria da cena. Sabendo que a base de dados utilizada é composta de vídeos obtidos por uma câmera apontada perpendicularmente à direção do movimento, que por sua vez é retilíneo, a profundidade dos pontos no espaço deve ser sempre a mesma. Com isso, como visto na Figura 5.2, cria-se uma relação entre a profundidade de um ponto $\mathbf{X}$ no espaço e o deslocamento das projeções $\mathbf{x_n}$:

$$\mathbf{x_1} = \begin{bmatrix} fX/Z & fY/Z & 1 \end{bmatrix}^T, \tag{5.4}$$

$$\mathbf{x_2} = \begin{bmatrix} x_1 - d_2 & y_1 & 1 \end{bmatrix}^T = \begin{bmatrix} f(X - t_2)/Z & fY/Z & 1 \end{bmatrix}^T. \tag{5.5}$$

Substituindo (5.4) em (5.5):

$$Z/f = t_2/d_2. \tag{5.6}$$

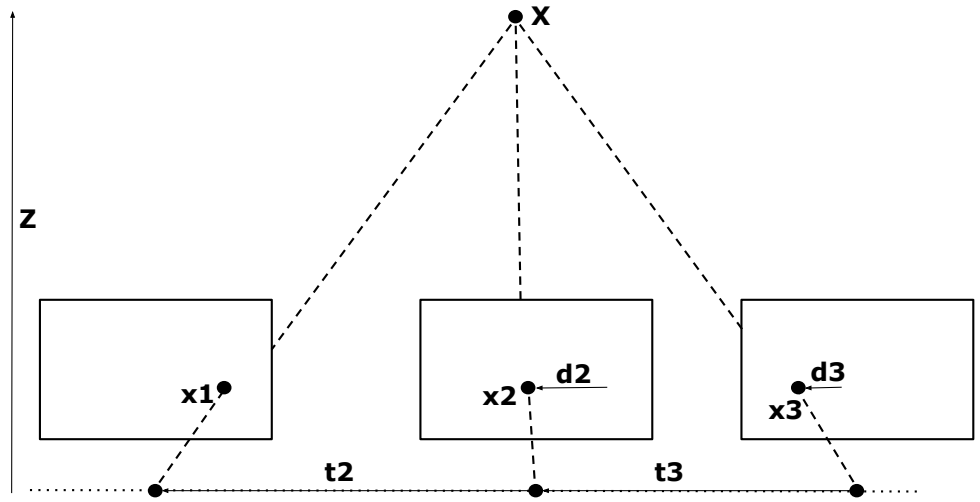


Figura 5.2: Projeções obtidas com uma câmera em movimento retilíneo, com orientação perpendiular ao movimento. O vetor $\mathbf{d_2}$ representa o deslocamento na imagem entre as posições dos píxels $\mathbf{x_1}$ e $\mathbf{x_2}$, de maneira análoga para $\mathbf{d_3}$.

Esta equação mostra que, caso um ponto se matenha sempre com a mesma profundidade em relação às vistas, a proporção t_2/d_2 deve ser constante. Durante a reconstrução de uma nova vista, obtém-se um vetor $\mathbf{t_n}$ com norma unitária mas, sabendo o deslocamento d_n em *pixels* sofrido pelos pontos na imagem entre uma

vista e outra, e de posse das mesmas informações para o par de vistas anterior, que já deve ter sido obtido, encontra-se a fórmula para o escalamento λ_n da nova vista:

$$\lambda_n = \lambda_{n-1} d_n / d_{n-1}. \tag{5.7}$$

Por fim, a posição do centro de câmera será a posição anterior somada ao novo deslocamento encontrado, convertido para o sistema de coordenadas da primeira câmera, ou seja:

$$\mathbf{C_n} = \mathbf{C_{n-1}} + \lambda_n \mathbf{R_n}^T \mathbf{t_n}. \tag{5.8}$$

5.1.4 Montagem da Imagem Panorâmica

Após a determinação da localização e orientação da câmera durante a captura dos quadros, realiza-se a criação de uma imagem panorâmica que combina os quadros. Uma superfície que vai conter a panorâmica é determinada a partir da trajetória da câmera. A imagem panorâmica poderia ser gerada a partir da retroprojeção de cada quadro considerado na superfície, com algum processamento nos trechos onde houver sobreposição. De maneira análoga, a própria superfície pode ser projetada nas imagens, para determinar onde alocar os *pixels* dos quadros na composição da imagem.

A superficie é amostrada em alguns pontos, que serão os *pixels* da panorâmica. Cada ponto na superfície é projetado em cada quadro utilizado do vídeo, de modo a encontrar quais *pixels* podem estar associados a esse ponto. A panorâmica é composta determinando, para cada ponto, um valor de luminância com base nas projeções encontradas. A Figura 5.3 ilustra esse procedimento.

5.2 Base de Dados

Foi utilizada a base de dados *Video Database of Abandoned Objects in a Cluttered Industrial Environment* (VDAO) [2] para todos os testes. Esta base consiste em vídeos em resolução *HD* gravados em um ambiente industrial, com uma câmera acoplada em um robô do tipo *Roomba* ©, realizando um movimento de ida e volta em um trilho, como ilustra a Figura 5.4.

A base possui 4 vídeos contendo somente o ambiente industrial, que são considerados os vídeos de referência, 56 vídeos com um objeto colocado em alguma posição do ambiente e 6 vídeos com múltiplos objetos distribuídos ao longo do campo de visão da câmera, além de apresentar dois tipos diferentes de iluminação do ambiente para cada vídeo. No total, soma-se mais de 8h de gravações com 24 objetos.

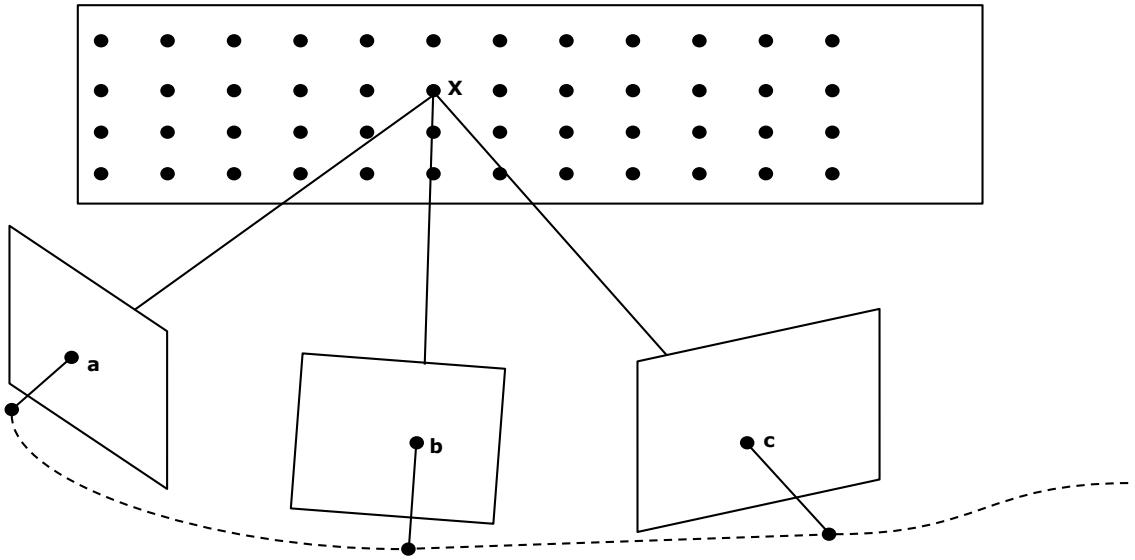


Figura 5.3: Montagem da imagem panorâmica. O ponto **X** se projeta nos pontos **a**, **b** e **c**. Na imagem panorâmica, o píxel equivalente ao ponto **X** será uma composição das projeções encontradas. A linha tracejada representa a trajetória do centro de câmera.

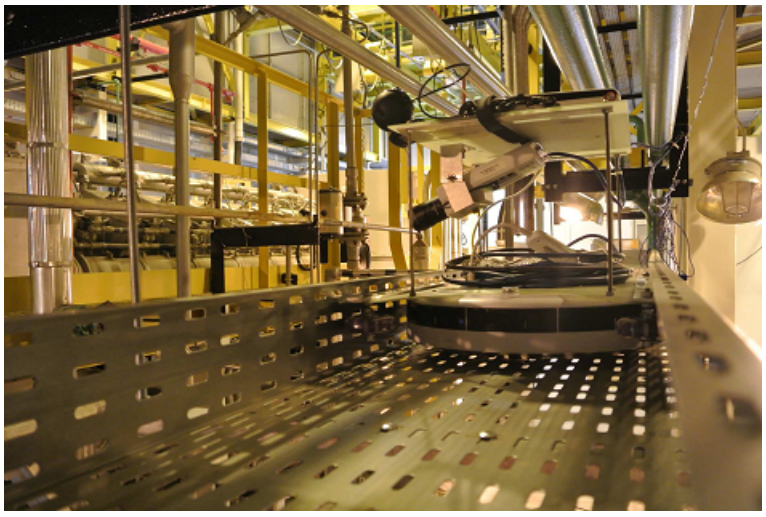


Figura 5.4: Sistema para monitoramento de um ambiente industrial com uma câmera montada em uma plataforma robotica.

Também está disponível uma marcação manual da posição dos objetos nos quadros, junto ao *software* responsável, de modo que esta base possa ser aplicada no teste e validação de algoritmos de detecção de objetos. As Figuras 5.5, 5.6 e 5.7 apresentam alguns trechos do ambiente obtidos através da câmera montada no robô.

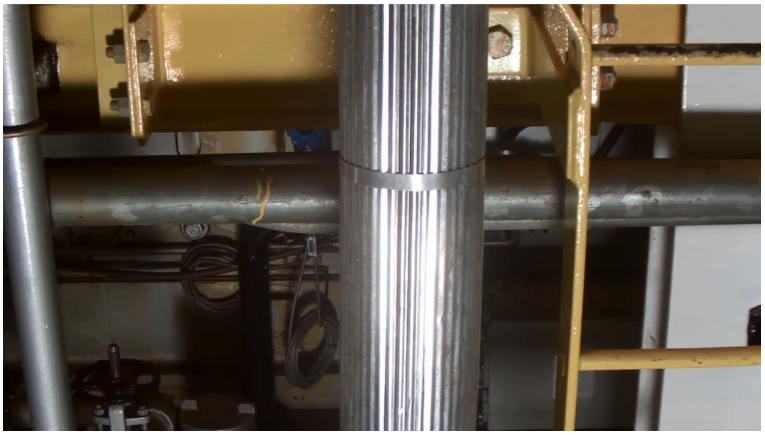


Figura 5.5: Trecho mais à direita do ambiente monitorado pelo sistema.

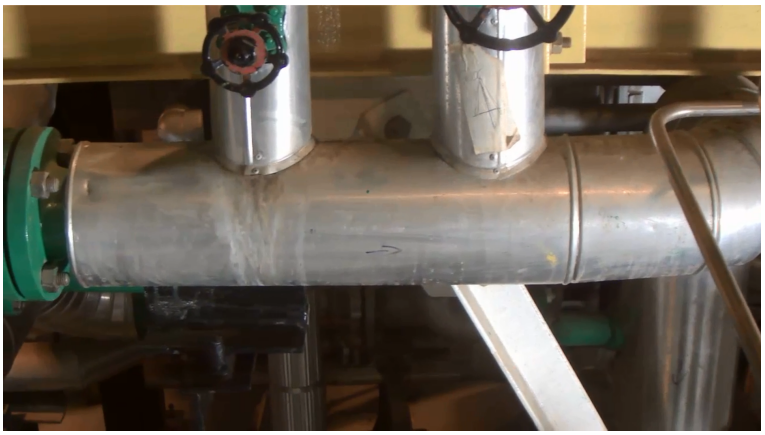


Figura 5.6: Trecho central do ambiente monitorado pelo sistema.

Figura 5.7: Trecho mais à esquerda do ambiente monitorado pelo sistema.

Capítulo 6

Resultados e Discussões

O algoritmo proposto no Capítulo 5 foi implementado em ambiente Linux com auxílio do OpenCV [58]. Foram utilizados para testes alguns trechos da base de dados referentes ao trajeto do robô de um extremo ao outro do trilho, e os gráficos para visualização dos resultados foram obtidos via MATLAB ©. A Seção 6.1 exibe as vantagens de utilizar um novo método para remoção de candidatos a pontos correspondentes. A Seção 6.2 expõe a melhoria na estimação ao utilizar a métrica para escolha da matriz essencial. Já a Seção 6.3 apresenta os problemas encontrados e os recursos utilizados para a obtenção da trajetória da câmera. Na Seção 6.4 está contido o processo de geração da imagem panorâmica, bem como alguns de seus resultados.

6.1 Correspondência entre Pontos

A proposta de eliminação dos pares de pontos que não se mantêm correspondentes entre pares diferentes, vista na Seção 5.1.1, foi analisada. Utilizando todos os pontos correspondentes obtidos através do SURF, aplicado nas duas vistas da Figura 6.1, obtém-se uma grande quantidade de pares, como pode ser visto na Figura 6.2.

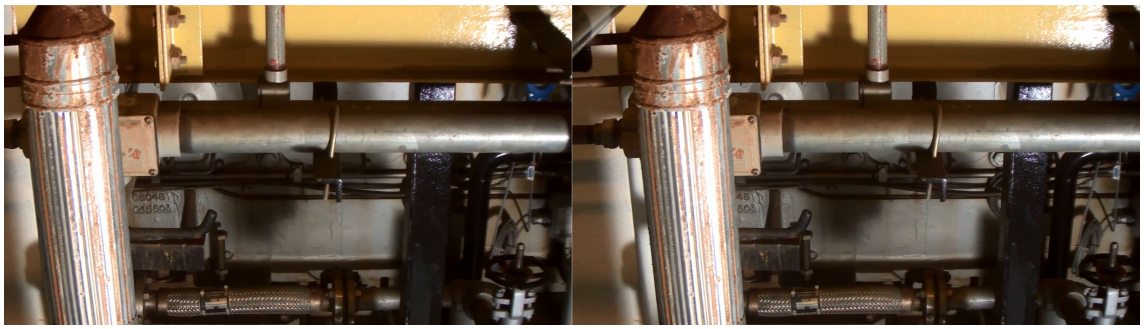


Figura 6.1: Par de vistas utilizadas para exemplificar a obtenção de pontos correspondentes.

Figura 6.2: Pares de pontos correspondentes obtidos pelo SURF.

Como o movimento entre os quadros é praticamente horizontal, espera-se que os pontos na imagem se desloquem horizontalmente, de forma que correspondências entre pontos que expressem um deslocamento oblíquo na imagem devem representar falsas correspondências de pontos. Além disso, uma vez que não há movimento da câmera se aproximar da cena, todos os objetos e estruturas não possuem muita variação de tamanho ou nível de detalhes. Assim, correspondências que apresentam pontos obtidos em escalas de detalhamento diferentes também não são confiáveis. Eliminando estes dois tipos de correspondências ruins, encontram-se os pontos apresentados na Figura 6.3. Neste caso, o movimento dos pontos parece representar o real movimento da câmera com maior exatidão.

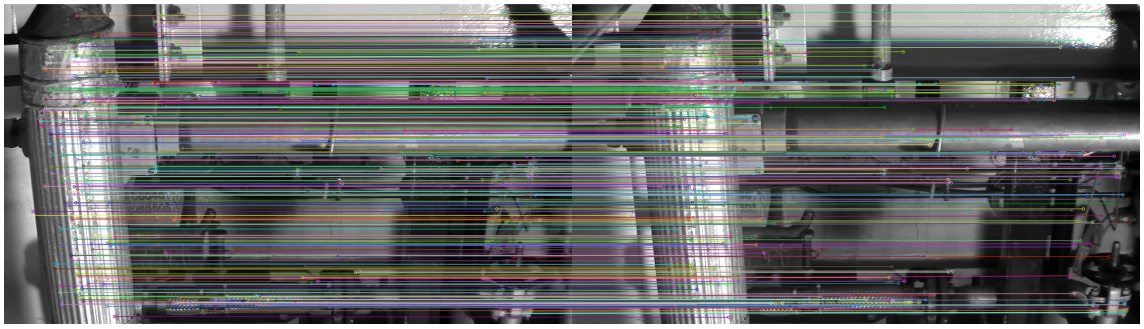


Figura 6.3: Pares de pontos correspondentes obtidos pelo SURF, após eliminação dos pares com ângulo grande e escala diferente.

De maneira a reduzir a quantidade de pontos, selecionando os melhores candidatos, foi aplicado também o critério exemplificado na Figura 5.1. Considera-se que os pontos salientes mais estáveis vão ser encontrados repetidamente ao longo dos quadros, de modo que deve ser possível encontrar as mesmas correspondências entre quadros diferentes. A Figura 6.4 mostra todos as correspondências mantidas por este critério, onde se nota uma redução considerável da quantidade de pontos, mantendo-se a qualidade das correspondências obtidas.

A Figura 6.5 compara o número de correspondências obtidas para vários pares

Figura 6.4: Pares de pontos correspondentes obtidos pelo SURF, após eliminação dos pares com ângulo grande, escala diferente e inconsistência entre quadros.

de quadros em um vídeo. Percebe-se que a adição do novo critério utilizado permite uma redução de até 50% do número de correspondências encontradas em relação ao obtido somente com os critérios anteriores, e uma redução de até 80% do número total de pontos. Com isso, diminui-se a quantidade de pontos que devem ser computados ou eliminados durante a execução do algoritmo, o que diminui o tempo de processamento.

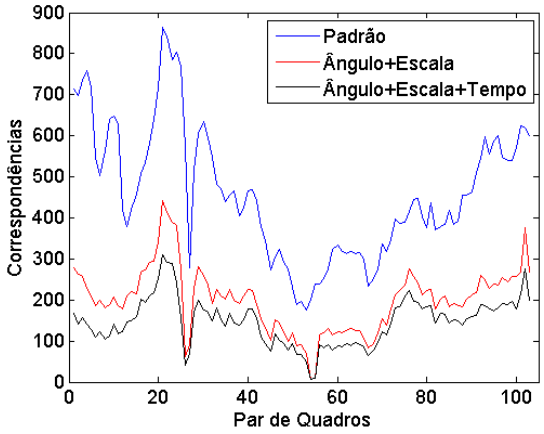


Figura 6.5: Número total de correspondências para cada par de quadros. A linha azul representa o número de correspondências obtidas pelo SURF. A linha vermelha apresenta o número de correspondências mantidas após a eliminação pelas métricas de ângulo e escala. A linha preta apresenta o número de correspondências mantidas após a eliminação pelas três métricas.

6.2 Geometria Epipolar

Testou-se o cálculo da matriz fundamental entre pares de quadros, utilizando o algoritmo de 5 pontos, disponível em [59]. Neste caso, comparou-se a versão original do algoritmo à alteração realizada, que leva em conta o movimento da câmera estimado

anteriormente como critério para seleção da melhor solução de matriz essencial.

Para o tipo de movimento apresentado pela câmera na base de vídeos utilizada, as duas vistas serão paralelas, e não haverá projeção do centro de uma câmera na outra, como pode ser visto na Figura 6.6. O epipolo neste caso estará no infinito e, como todas as linhas epipolares cruzam o epipolo, elas todas devem se encontrar no infinito, o que é a definição de retas paralelas. Desse modo, um critério indireto utilizado para medir a acurácia do cálculo da matriz fundamental é analisar as linhas epipolares obtidas.

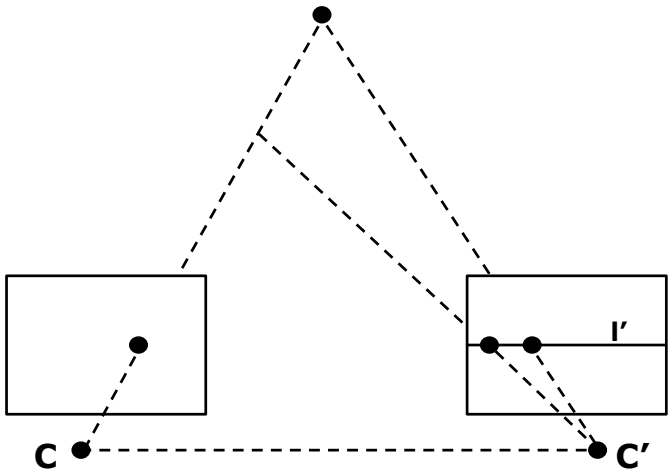


Figura 6.6: Geometria epipolar para uma câmera efetuando um movimento de translação pura.

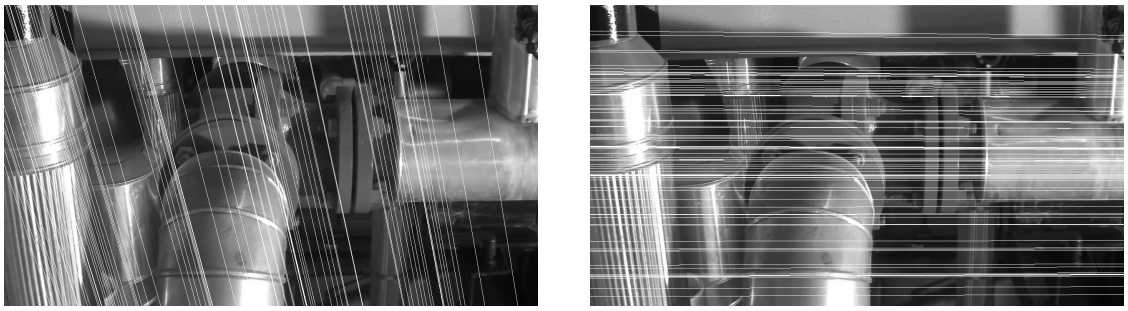
Encontram-se nas Figuras 6.7 e 6.8 alguns exemplos para comparação das linhas epipolares obtidas. Para o algoritmo original, visto nas Figuras 6.7(a) e 6.8(a), as linhas epipolares apresentaram um comportamento errático, com a posição do epipolo variando em praticamente todo o espaço. Já com a alteração proposta, as linhas epipolares se mantêm na maior parte do tempo com uma certa variação ao redor do eixo horizontal, que pode ser causada por alterações no movimento da própria câmera, devido a vibrações da mesma ao longo da sua trajetória. Percebe-se, deste modo, que a alteração do algoritmo de 5 pontos, para que a solução escolhida seja a que estima um movimento mais próximo do anterior, fornece um resultado mais condizente com o tipo de movimento esperado no vídeo.

6.3 Trajetória de Câmera

A posição do centro de câmera foi estimada para diversos quadros dos vídeos da base de dados, juntamente com sua respectiva orientação na cena. Devido à natureza do movimento, era esperada uma trajetória com comportamento retilíneo ao longo dos

(a) (b)

Figura 6.7: Conjunto de linhas epipolares para o par de quadros 14. (a) Algoritmo de 5 pontos original. (b) Algoritmo de 5 pontos com a modificação proposta.



(a) (b)

Figura 6.8: Conjunto de linhas epipolares para o par de quadros 59. (a) Algoritmo de 5 pontos original. (b) Algoritmo de 5 pontos com a modificação proposta.

quadros. Entretanto, o resultado obtido mostrou um comportamento anômalo para o deslocamento do centro de câmera. Analisando o problema, retratado na Figura 6.9, constatou-se que para cada par de vistas existe uma pequena rotação aleatória, o que era esperado já que existe a vibração do robô, mas também existe uma rotação média, que deveria ser nula, visto que a câmera está rígida no robô. As Figuras 6.10, 6.11 e 6.12 expõem o resultado irregular para a trajetória, com um movimento que, além de não ser uniforme, apresenta uma variação de profundidade da mesma ordem de grandeza do deslocamento horizontal.

Para compensar essa polarização nos resultados, foi calculada uma média entre as rotações relativas entre as vistas, que foi subtraída de cada rotação encontrada. Desse modo, obtiveram-se os resultados presentes também nas Figuras 6.10,6.11 e 6.12. Percebe-se que, em todos os casos, a componente X do movimento apresenta uma curva similar a uma reta indicando um movimento retilíneo e decrescente em relação ao eixo, que é centrado no primeiro quadro utilizado. Este resultado era esperado, visto que o trecho do vídeo considerado apresenta um movimento para a esquerda. Além disso, a amplitude total do movimento também é diferente para cada vídeo, o que também não é um problema, uma vez que cada reconstrução foi feita de maneira independente, e normalizada em relação à translação obtida no

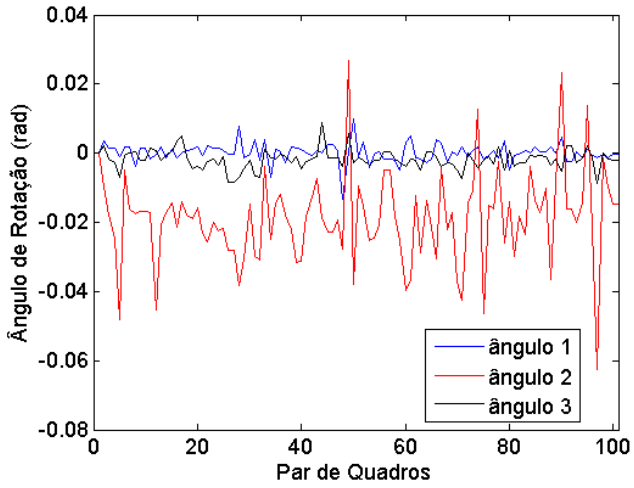
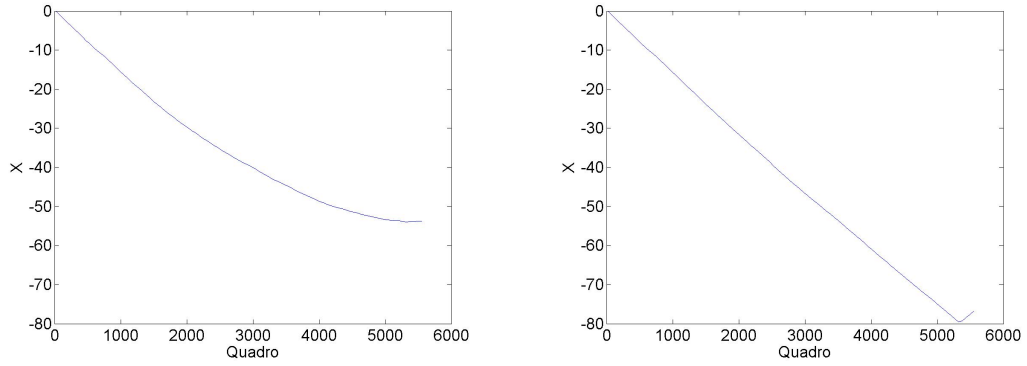


Figura 6.9: Rotação Relativa entre cada par de quadros. Os ângulos 1, 2 e 3 referem-se à transformação da matriz de rotação em ângulos de rotação [15].

primeiro par de câmeras.

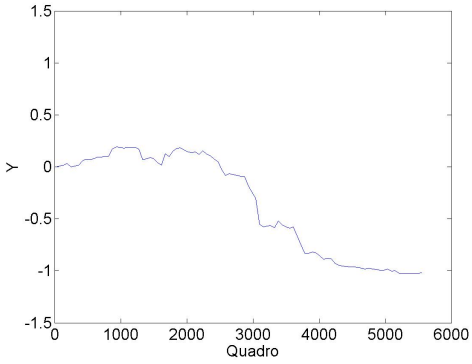
Já a componente Y apresenta uma trajetória um pouco menos linear, mas ainda assim com um bom indicativo do percurso realizado. A amplitude total do movimento também se manteve aproximadamente a mesma, considerando a normalização aplicada em cada cálculo. Intui-se, neste caso, que o trilho não estava perfeitamente alinhado com a horizontal, e o robô realiza um pequeno movimento na vertical, em relação ao primeiro quadro.

Por fim, a componente Z não apresenta um comportamento regular para todos os vídeos. Uma vez que o movimento foi obtido através de imagens não é trivial recuperar a profundidade dos pontos, que não é bem representada. Deve-se ressaltar, porém, que o movimento é bastante limitado se comparado ao obtido antes do ajuste, e possui amplitude muito menor que o movimento total.

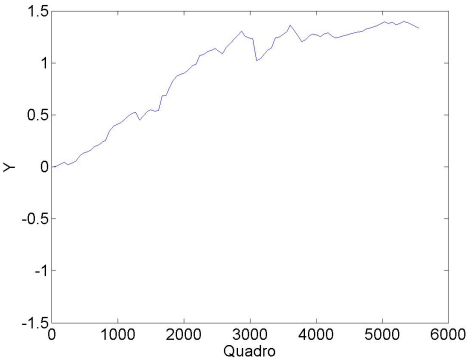


(a)

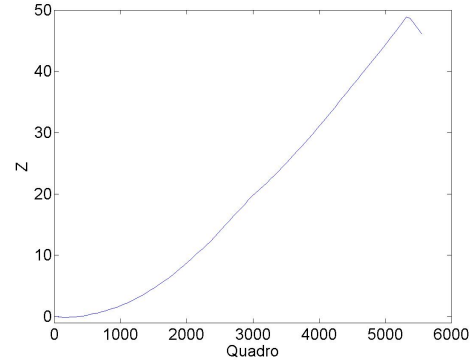
(b)



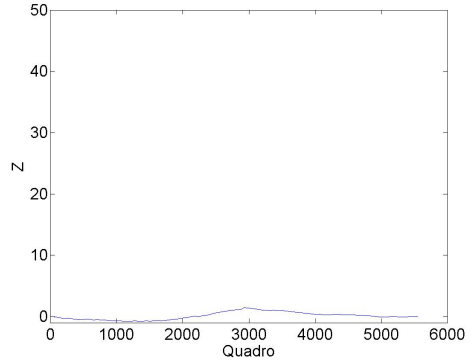
(c)



(d)



(e)



(f)

Figura 6.10: Trajetória de câmera para o vídeo ext-part02-video01. A unidade de medida considera unitário o deslocamento entre o primeiro par de câmeras. (a) x. (b) x com compensação. (c) y. (d) y com compensação. (e) z. (f) z com compensação.

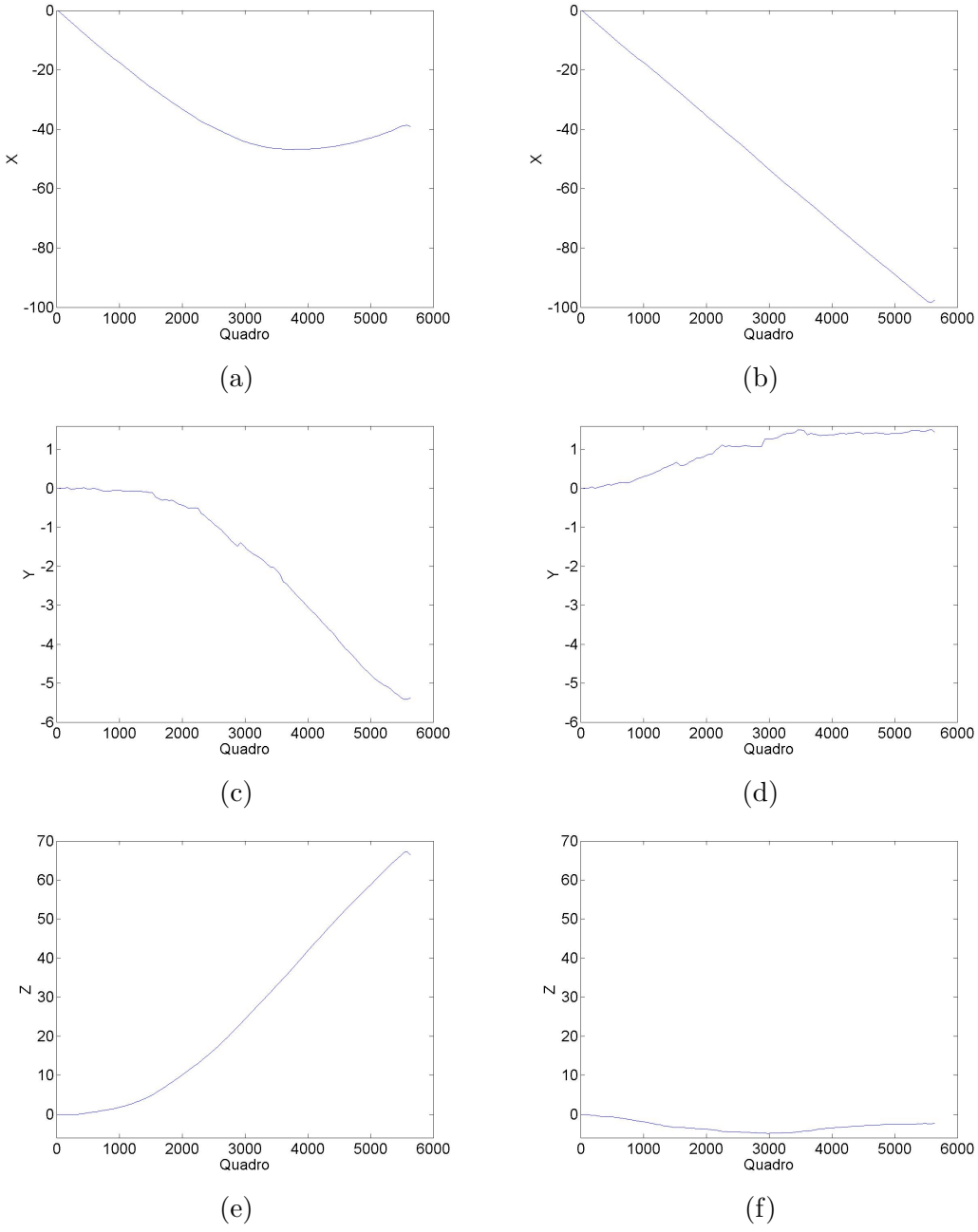


Figura 6.11: Trajetória de câmera para o vídeo ext-part03-video13. A unidade de medida considera unitário o deslocamento entre o primeiro par de câmeras. (a) x. (b) x com compensação. (c) y. (d) y com compensação. (e) z. (f) z com compensação.

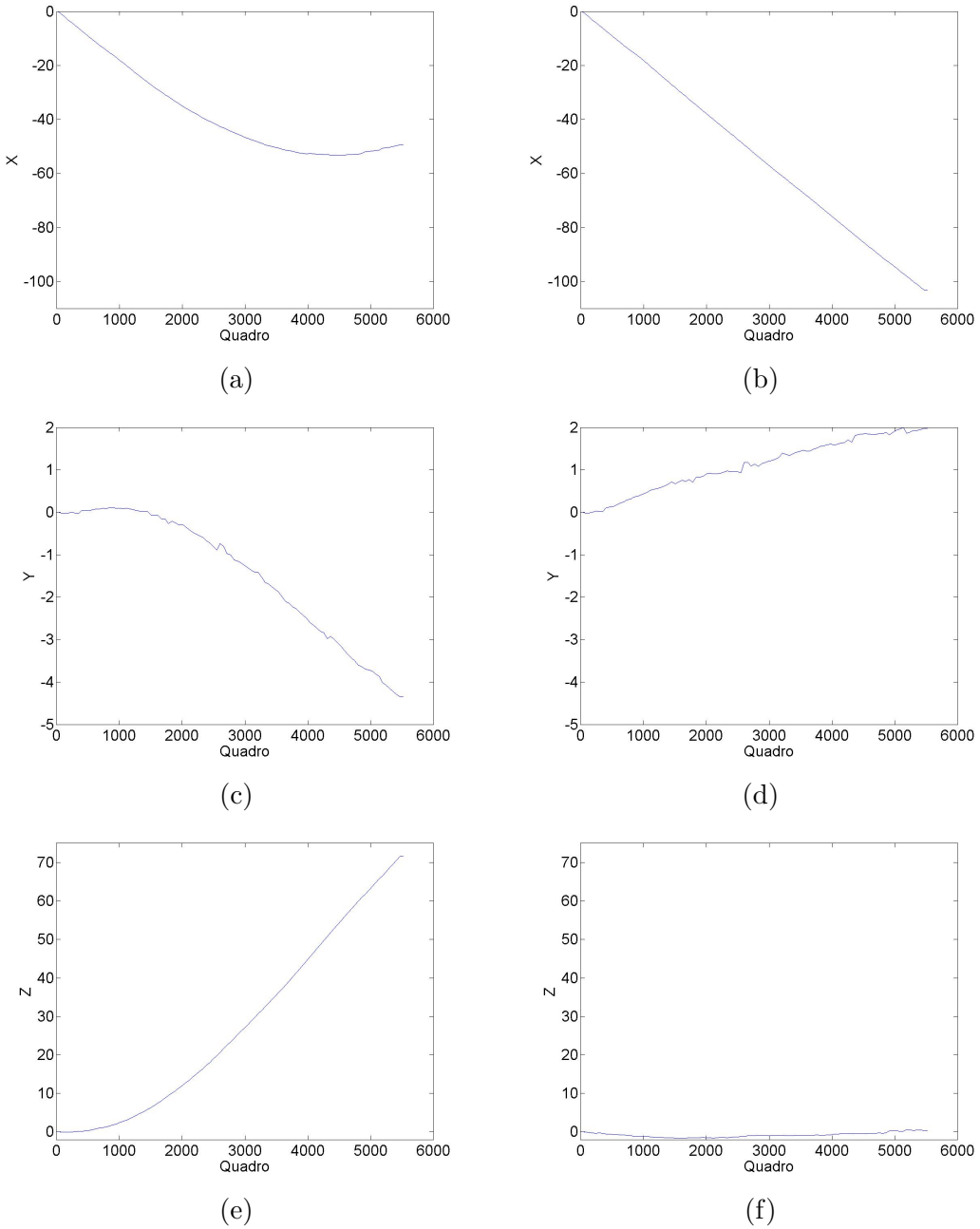


Figura 6.12: Trajetória de câmera para o vídeo de referência. A unidade de medida considera unitário o deslocamento entre o primeiro par de câmeras. (a) x. (b) x com compensação. (c) y. (d) y com compensação. (e) z. (f) z com compensação.

6.4 Montagem da Imagem Panorâmica

De posse de trajetória e orientação da câmera em cada vista, montou-se uma imagem do cenário como um todo. Para isto, definiu-se um plano em uma profundidade proporcional ao deslocamento total da câmera, que foi amostrado com base nas posição e resolução das imagens. As amostras são projetadas em cada imagem para determinar a retroprojeção de cada vista no plano, que serão combinadas em uma imagem única, formando a imagem-panorama. A Figura 6.13 mostra exemplos de quadros retroprojetados.

(a)

(b)

(c)

Figura 6.13: Quadros reprojetados no plano considerado. (a) Quadro 1. (b) Quadro 51. (c) Quadro 101.

As imagens retroprojetadas, como visto na Figura 6.13, são combinadas utilizando média ponderada para as regiões de interseção. Espera-se que uma região do plano seja melhor representada pela imagem mais próxima. Dessa forma, a distância entre o píxel do plano e o centro de câmera é utilizada como o inverso do peso.

As Figuras 6.14, 6.15 apresentam exemplos de imagens panôramicas compostas com alguns vídeos. Percebe-se um borramento geral na cena, causado pelo cálculo da média entre elementos que não estão exatamente sobrepostos, além de haver grande influência da vibração da câmera, que gera diversos objetos fantasmas.

As regiões com maior nitidez são aquelas onde o plano da panorâmica está próximo dos objetos da cena porque, neste caso, a projeção efetuada para encontrar os pontos na imagem que poderiam compor a panorâmica é a mesma projeção sofrida pelos objetos reais do espaço durante a captura dos quadros, não havendo problemas de disparidade. Por essa razão, os canos verticais, que estão mais próximos da câmera, foram melhor retratados com um plano de projeção de menor profundi-

dade. Em todos os casos, a região da esquerda não foi bem representada, visto que ela possui um trecho com muita variação de profundidade.

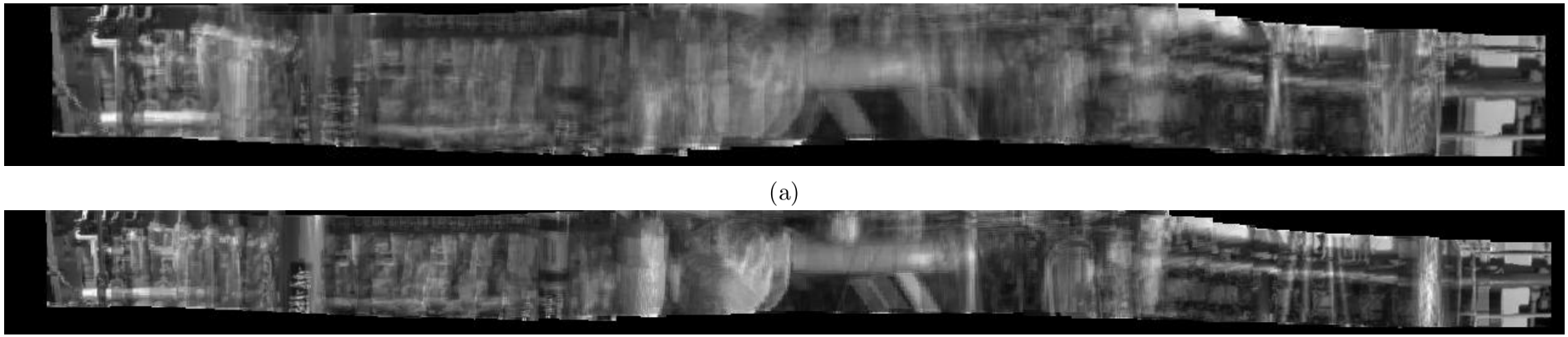
Optou-se por utilizar a média aritmética para compor a imagem panorâmica, visto que é um método de fácil implementação. Para que as imagens sejam visualmente mais agradáveis a um usuário, deveriam ser utilizadas técnicas tradicionais de panorama para costura dos quadros. Entretanto, estes métodos são em geral não lineares, consistindo de algum procedimento de decisão, que poderia ser responsável por eliminar algum trecho contendo o objeto que deve ser detectado da imagem final.

Apesar do borramento, ainda é possível reconhecer objetos anômalos no ambiente. A Figura 6.17(a) mostra um trecho da Figura 6.15(b), onde é possível identificar um objeto, no caso a toalha presente na Figura 6.17(b).

(a)

(b)

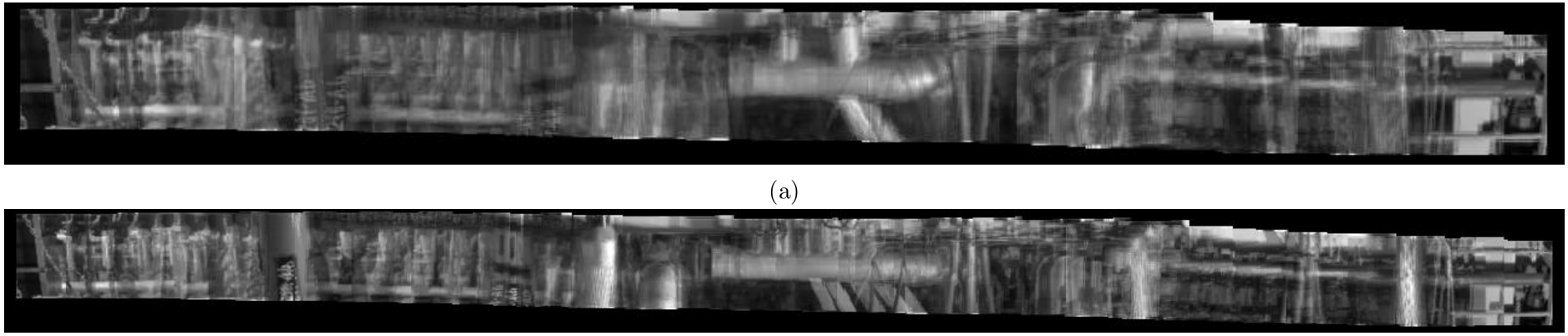
Figura 6.14: Imagens panorâmicas compostas a partir do vídeo ext-part02-video01. (a) Com um plano mais distante. (b) Com um plano mais próximo.



(a)

(b)

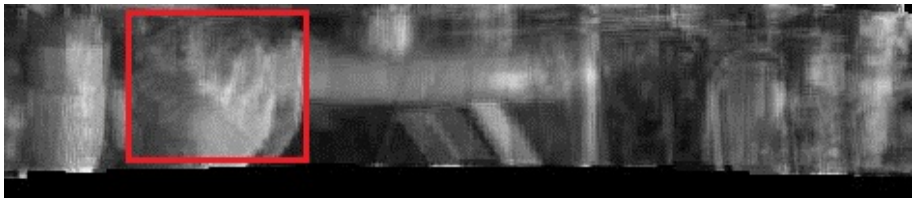
Figura 6.15: Imagens panorâmicas compostas a partir do vídeo ext-part03-video06. (a) Com um plano mais distante. (b) Com um plano mais próximo.



(a)

(b)

Figura 6.16: Imagens panorâmicas compostas a partir do vídeo de referência. (a) Com um plano mais distante. (b) Com um plano mais próximo.



(a)

(b)

Figura 6.17: Identificação de um objeto na imagem-panorama. (a) Fragmento da imagem-panorama que contém um objeto. (b) Objeto localizado.

Capítulo 7

Conclusões

7.1 Trabalho Realizado

Apresentou-se neste trabalho, a implementação de um sistema para detectar a trajetória de uma câmera em movimento que monitora um ambiente carregado. Este algoritmo pretende auxiliar um sistema para detecção de objetos abandonados.

A trajetória da câmera foi obtida a partir de um algoritmo de *Structure from Motion* adaptado para operar com imagens adquiridas por uma câmera que percorre um movimento retilíneo. Algumas melhorias foram implementadas, que funcionam bem para o tipo de movimento proposto. Além disso, foi constatado um erro sistemático na estimação da rotação, que teve que ser corrigido. Constatou-se que este método não é recomendado para a estimação de um movimento em que haja muita variação de profundidade no trajeto da câmera.

Também foi gerada uma imagem estilo panorâmica a partir da trajetória obtida. Esta imagem apresenta uma visão geral de toda a cena, que pode ser apresentada a um operador ou vir a ser utilizada no processo de deteção de objetos. No geral, conseguiram-se bons resultados, considerando que não houve um processamento mais complexo para a composição do panorama. Com a escolha correta de uma superfície de projeção adequada, a imagem pode ser nítida o suficiente para ser utilizada em uma inspeção visual.

7.2 Próximos Passos

Este trabalho possui algumas ramificações que podem ser seguidas. O algoritmo utilizado funciona somente para uma passada de um percurso retilíneo. Uma continuação imediata é tornar o algoritmo compatível com um percurso de ida e volta, detectando início e fim do movimento e reiniciando o cálculo, uma vez que a câmera parada pode gerar um par de vistas com uma configuração degenerada para a ge-

ometria epipolar. Uma futura contribuição será a implementação do mesmo em percursos com curvas e trajetórias fechadas. Neste caso, também é necessário um algoritmo para detecção de *loop*, de forma a garantir que a trajetória sempre retorne a um ponto inicial.

A trajetória obtida pode ser testada para utilização em outros fins. Através do cálculo da trajetória para dois vídeos distintos, pode ser possível realizar um alinhamento temporal entre eles.

Para a imagem panorâmica, são necessários testes complementares com outros métodos para a junção das imagens e a definição de outros planos para a projeção. Além disso, a imagem panorâmica obtida não foi testada para utilização em qualquer processamento, sendo atualmente somente para visualização. No domínio da panorâmica, toda a informação da cena pode estar presente em uma única imagem, de modo que a detecção de objetos pode ser feita através da mesma, substituindo a necessidade de comparar dois vídeos inteiros.

Apêndice A

Lista de Artigos Derivados deste Trabalho

DA SILVA, A. F., THOMAZ, L. A., CARVALHO, G., et al. "An annotated video database for abandoned-object detection in a cluttered environment". In: International Telecommunications Symposium, pp. 1-5, 2014.

KUCHARCZAK, F., DA SILVA, A. F., THOMAZ, L. A., et al. "Comparison and Optimization of Image Descriptors for Real-Time Detection of Abandoned Objects". In: Anais do Simpósio de Processamento de Sinais da UNICAMP, v. 1, 2014. Disponível em: `http://www.sps.fee.unicamp.br/anais/`.

THOMAZ, L. A., DA SILVA, A. F., DA SILVA, E. A. B., et al. "Abandoned Object Detection Using Operator-Space Pursuit". In: Submited to International Conference on Image Processing, 2015.

Referências Bibliográficas

[1] TOMIOKA, Y., TAKARA, A., KITAZAWA, H. "Generation of an Optimum Patrol Course for Mobile Surveillance Camera". In: *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, pp. 216–224, fev. 2012.

[2] DA SILVA, A., THOMAZ, L., CARVALHO, G., et al. "An Annotated Video Database for Abandoned-Object Detection in a Cluttered Environment". In: *International Telecommunications Symposium (ITS)*, pp. 1–5, São Paulo, SP, BR, ago. 2014.

[3] STAUFFER, C., GRIMSON, W. "Adaptive background mixture models for real-time tracking". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, v. 2, p. 246–252, 1999.

[4] MUKHERJEE, D., WU, Q. M. J., NGUYEN, T. M. "Multiresolution Based Gaussian Mixture Model for Background Suppression", *IEEE Transactions on Image Processing*, v. 22, n. 12, pp. 5022–5035, 2013.

[5] BEHRAD, A., SHAHROKNI, A., MOTAMEDI, S. A. "A Robust Vision-based Moving Target Detection and Tracking System", 2001.

[6] JUNG, B., SUKHATME, G. S. "Detecting moving objects using a single camera on a mobile robot in an outdoor environment". In: *International Conference on Intelligent Autonomous Systems*, pp. 980–987, 2004.

[7] TSINKO, E. *Background Subtraction with a Pan/Tilt Camera*. B. Sc. dissertation, University Of British Columbia, Vancouver, dez. 2010.

[8] ELGAMMA, A., HARWOOD, D., DAVIS, L. "Non-parametric model for background subtraction". In: *2000 IEEE International Conference on Computer Vision (ICCV)*.

[9] KIM, K., CHALIDABHONGSE, T. H., HARWOOD, D., et al. "Real-time Foreground-background Segmentation Using Codebook Model". In: *6th*

IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), v. 11, pp. 172–185, London, UK, UK, jun. 2005. Academic Press Ltd.

[10] HAYMAN, E., OLOF EKLUNDH, J. "Statistical Background Subtraction for a Mobile Observer". In: *Proceedings ICCV*, pp. 67–74, 2003.

[11] ZHAO, Y., CASARES, M., VELIPASALAR, S. "Continuous Background Update and Object Detection with Non-static Cameras." In: *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 309–316, 2008.

[12] FARIN, D., DE WITH, P. H. N., DE, P. H. N., et al. "Video-Object Segmentation Using Multi-Sprite Background Subtraction". In: *Proceedings IEEE International Conference on Multimedia and Expo*, pp. 343–346, 2004.

[13] RAO, N. I., DI, H., XU, G. "Joint Correspondence and Background Modeling Based on Tree Dynamic Programming". In: *18th International Conference on Pattern Recognition*, v. 2, pp. 425–428, 2006.

[14] JIN, Y., TAO, L., DI, H., et al. "Background modeling from a free-moving camera by Multi-Layer Homography Algorithm." In: *15th IEEE International Conference on Image Processing*, pp. 1572–1575, 2008.

[15] HARTLEY, R., ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[16] SUN, S.-W., WANG, Y.-C. F., HUANG, F., et al. "Moving foreground object detection via robust SIFT trajectories." *Journal of Visual Communication and Image Representation*, v. 24, n. 3, pp. 232–243, 2013.

[17] LOWE, D. G. "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, 2004.

[18] KONG, H., AUDIBERT, J.-Y., PONCE, J. "Detecting Abandoned Objects with a Moving Camera", v. 19, n. 8, pp. 2201–2210, ago. 2010.

[19] CARVALHO, G., DE OLIVEIRA, J. F. L., DA SILVA, E. A. B., et al. "Um Sistema de Monitoramento para Detecção de Objetos em Tempo Real empregando Camera em Movimento". In: *XXXI Simpósio Brasileiro de Telecomunicações*, Fortaleza, set. 2013.

[20] KUCHARCZAK, F., DA SILVA, A. F., THOMAZ, L. A., et al. "Comparison and Optimization of Image Descriptors for Real-Time Detection of Abandoned Objects". In: *SPS-UNICAMP*, Campinas, 2014.

[21] BAY, H., ESS, A., TUYTELAARS, T., et al. "Speeded-Up Robust Features (SURF)", *Computer Vision and Image Understanding (CVIU)*, v. 110, n. 3, pp. 346–359, jun. 2008.

[22] LEUTENEGGER, S., CHLI, M., SIEGWART, Y. "Brisk: Binary robust invariant scalable keypoints". In: *2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 2548–2555, 2011.

[23] ALAHI, A., ORTIZ, R., VANDERGHEYNST, P. "FREAK: Fast Retina Keypoint". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 510 – 517.

[24] SERRAT, J., DIEGO, F., LUMBRERAS, F. "Alignment of videos recorded from moving vehicles". In: *International Conference on Image Analysis and Patterns (ICIAP)*, 2007.

[25] EVANGELIDIS, G., BAUCKHAGE, C. "Efficient and Robust Alignment of Unsynchronized Video Sequences". In: *DAGM 2011 - 33rd Annual Symposium of the German Association for Pattern Recognition*, p. 286–295, set. 2011.

[26] EVANGELIDIS, G., BAUCKHAGE, C. "Efficient Subframe Video Alignment Using Short Descriptors". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, p. 2371–2386, 2013.

[27] EVANGELIDIS, G., PSARAKIS, E. "Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 30, p. 1858–1865, 2008.

[28] DIEGO, F., PONSA, D., SERRAT, J., et al. "Video Alignment for Change Detection", v. 20, n. 7, pp. 1858–1869, jul. 2011.

[29] DIEGO, F., S. J. L. A. "Joint Spatio-Temporal Alignment of Sequences". In: *IEEE Transaction on Multimedia*, v. 15, p. 1520–9210, 2013.

[30] ROZANOV, Y. *Markov Random Fields*. Springer New York, 1982.

[31] CORNELIS, N., CORNELIS, K.AND VAN GOOL, L. "Fast Compact City Modeling for Navigation Pre-Visualization". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1339 – 1344, 2006.

[32] SNAVELY, N., SEITZ, S. M., SZELISKI, R. "Photo Tourism: Exploring Photo Collections in 3D". In: *ACM SIGGRAPH*, pp. 835–846, New York, NY, USA, 2006. ACM.

[33] WENG, J., HUANG, T. S., AHUJA, N. *Motion and Structure from Image Sequences*. Springer Publishing Company, Incorporated, 2012.

[34] TRIGGS, B., MCLAUCHLAN, P. F., HARTLEY, R. I., et al. "Bundle Adjustment - A Modern Synthesis". In: *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pp. 298–372, London, UK, 2000. Springer-Verlag.

[35] CORNELIS, N., LEIBE, B., CORNELIS, K., et al. "3D City Modeling Using Cognitive Loops". In: *3rd International Symposium on 3D Data Processing, Visualization, and Transmission*, jun. 2006.

[36] KLOPSCHITZ, M., ZACH, C., IRSCHARA, A., et al. "Generalized detection and merging of loop closures for video sequences". In: *4th International Symposium on 3D Data Processing, Visualization and Transmission*, jun. 2008.

[37] SCARAMUZZA, D., FRAUNDORFER, F., SIEGWART, R., et al. "Closing the loop in appearance guided SfM for omnidirectional cameras". In: *8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, pp. 1–14, 2008.

[38] AGARWALA, A., AGRAWALA, M., COHEN, M., et al. "Photographing Long Scenes with Multi-viewpoint Panoramas". In: *ACM SIGGRAPH*, pp. 853–861, New York, NY, USA, 2006. ACM.

[39] OKATANI, T., YANAGISAWA, J., TETSUKA, D., et al. "Creating Multi-Viewpoint Panoramas of Streets with Sparsely Located Buildings". In: *Field and Service Robotics*, v. 92, pp. 65–79, 2014.

[40] CASTELLANOS, J. A., MONTIEL, J. M. M., NEIRA, J., et al. "The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building", v. 15, pp. 948–953, 1999.

[41] DISSANAYAKE, M., NEWMAN, P., CLARK, S., et al. "Photographing Long Scenes with Multi-viewpoint Panoramas". In: *IEEE Transactions on Robotics and Automation*, v. 17, pp. 229–241, 2001.

[42] ENDRES, F., HESS, J., ENGELHARD, N., et al. "An evaluation of the RGB-D SLAM system". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1691 – 1696, 2012.

[43] STEDER, B., GRISETTI, G., STACHNISS, C., et al. "Visual SLAM for Flying Vehicles". In: *IEEE Transactions on Robotics*, v. 24, pp. 1088–1093, out. 2008.

[44] LEE, S., LEE, S. "Embedded Visual SLAM: Applications for Low-Cost Consumer Robots", *IEEE Robotics & Automation Magazine*, v. 20, pp. 83–95, dez. 2013.

[45] MEILLAND, M., COMPORT, A. "On unifying key-frame and voxel-based dense visual SLAM at large scales". In: *International Conference on Intelligent Robots and Systems*, Tokyo, Japan, nov. 2013.

[46] DAVISON, A. J. "SLAM with a Single Camera". In: *SLAM/CML Workshop at ICRA*, 2002.

[47] DAVISON, A. J. "Real-Time Simultaneous Localisation and Mapping with a Single Camera". In: *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, pp. 1403–, Washington, DC, USA, 2003.

[48] SALAS-MORENO, R. F., NEWCOMBE, R. A., STRASDAT, H., et al. "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1352–1359, 2013.

[49] FAUGERAS, O., LUONG, Q.-T., PAPADOPOULO, T. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, 2001.

[50] XU, G., ZHANG, Z. *Epipolar Geometry in Stereo, Motion and Object Recognition*. Kluwer Academic Publishers, 1996.

[51] MAYBANK, S. *Theory of Reconstruction from Image Motion*. Springer Verlag, 1993.

[52] LONGUET-HIGGINS, H. C. "A computer algorithm for reconstructing a scene from two projections", *Nature*, v. 293, n. 5828, pp. 133–135, set. 1981.

[53] NISTER, D. "An efficient solution to the five-point relative pose problem", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 6, pp. 756–770, jun. 2004.

[54] LI, H., HARTLEY, R. "Five-Point Motion Estimation Made Easy". In: *Proceedings of the 18th International Conference on Pattern Recognition - Volume 01*, pp. 630–633, Washington, DC, USA, 2006.

[55] LUONG, Q.-T., VIÉVILLE, T. "Canonical Representations for the Geometries of Multiple Projective Views", *Computer Vision and Image Understanding*, v. 64, n. 2, pp. 193–229, set. 1996.

[56] NISTÉR, D. "An Efficient Solution to the Five-Point Relative Pose Problem", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 6, pp. 756–777, jun. 2004.

[57] LUI, V., DRUMMOND, T. "An Iterative 5-pt Algorithm for Fast and Robust Essential Matrix Estimation". In: *Proceedings of the British Machine Vision Conference*, Bristol, UK, set. 2013.

[58] "OpenCV". `http://opencv.org/`. Acessado: 26-02-2015.

[59] "Algoritmo de 5 pontos". `http://nghiaho.com/?p=1675s`. Acessado: 26-02-2015.