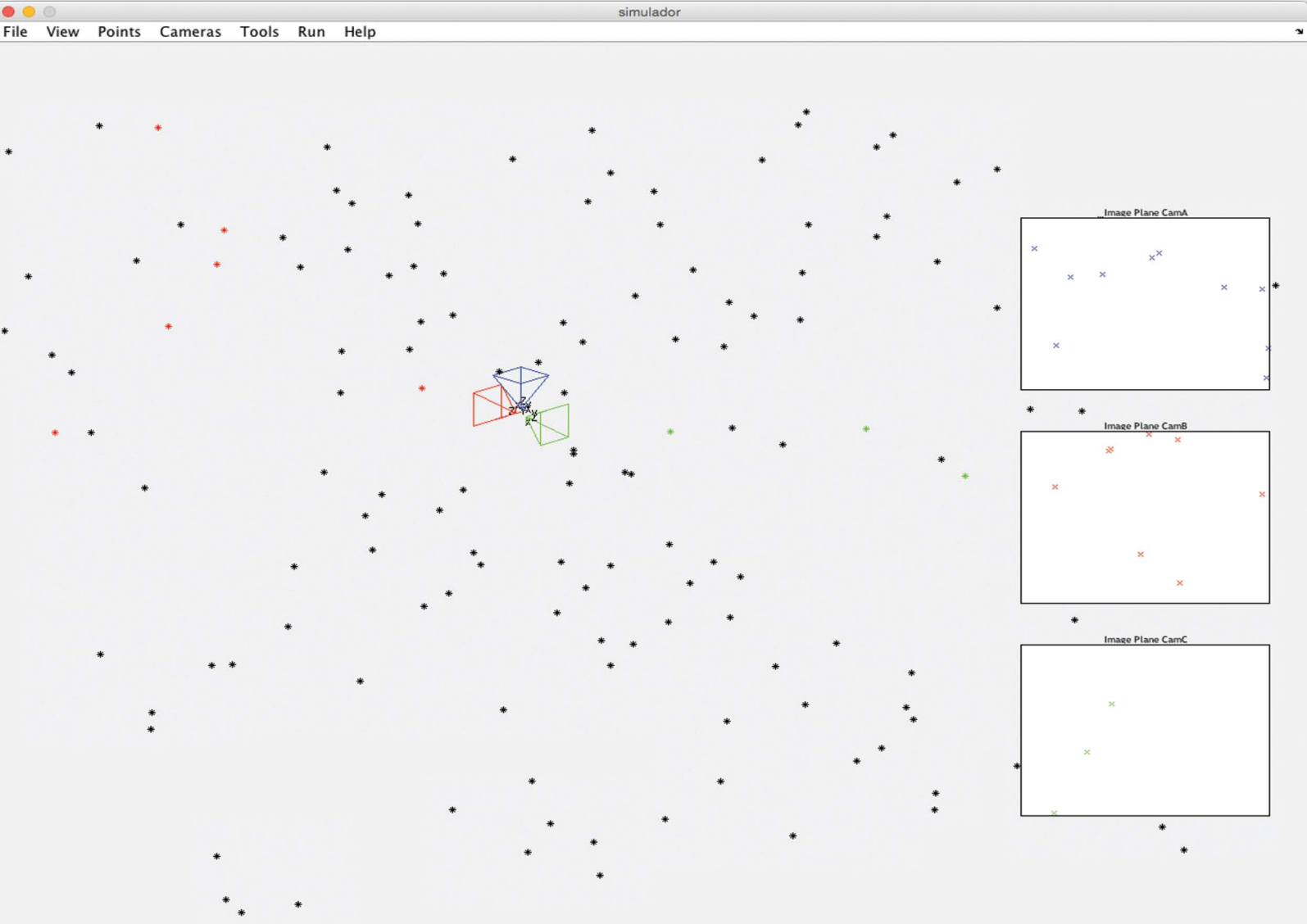


António Pedro Pinto Ribeiro

Odometria Visual
usando campos visuais não sobrepostos

Março 2015

· U C ·

Universidade de Coimbra

Departamento de Engenharia Electrotécnica e de Computadores
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

Dissertação apresentada à Faculdade de Ciências e Tecnologia da Universidade de Coimbra para obtenção do grau de Mestre em Engenharia Electrotécnica e de Computadores, especialidade Automação

Odometria Visual usando campos visuais não sobrepostos

António Pedro Pinto Ribeiro

Orientadores:
Professor Doutor Helder de Jesus Araújo
Doutor Pedro Daniel dos Santos Miraldo

Júri:
Professor Doutor João Pedro de Almeida Barreto
Professor Doutor Helder de Jesus Araújo
Professor Doutor Nuno Miguel Mendonça da Silva Gonçalves

Março 2015

Agradecimentos

Gostaria de começar por agradecer ao meu orientador Professor Doutor Helder de Jesus Araújo por todo conhecimento científico, apoio moral e orientação ao longo da realização deste trabalho. Ao meu co-orientador Doutor Pedro Miraldo por ter sido incansável no seu apoio, conhecimento disponibilizado e paciência constante desde o primeiro momento até à conclusão da dissertação.

Quero agradecer a toda a minha família, mas de uma forma muito especial e sentida aos meus pais e irmã, António Delfim Ribeiro, Maria de Fátima Ribeiro e Mariana Pinto Ribeiro, respectivamente, por desejarem sempre o melhor para mim e me darem todo o amor ao longo destes anos de distância, estando sempre ao meu lado nas alturas difíceis e porque sem eles nada disto teria sido possível.

Agradeço também a todos os meus colegas do Laboratório de Visão por Computador, em especial ao Tiago Dias, que me apoiou e esteve ao meu lado durante muitas horas de trabalho.

Aos Gato Fedorento, por existirem e proporcionarem pequenos momentos de boa disposição com os seus vídeos.

Agradeço ainda a todos os meus amigos, não mencionando nenhum em particular pois seria injusto esquecer-me de alguém, e todos de uma forma ou de outra foram importantes neste percurso académico.

Por fim, mas não menos importante, agradecer à Catarina Monteiro por ser a pessoa fantástica que é, pela companhia e por me ter dado uma apoio incondicional durante um percurso longo e difícil, que teria sido bem mais complicado sem o seu carinho.

A todos, o meu sincero obrigado,
Pedro.

Abstract

This Master Thesis is mainly focused the study of Visual Odometry using multi-camera systems with non-overlapping visual fields. The motion estimation plays a fundamental role in computer vision research.

During the last decade, there has been a considerable growing in the study of Visual Odometry. Initially only a combined camera with both sensors was used. Later, the classical stereo pair systems appeared, which made the motion estimation more robust. One limitation of these systems is field of view is the same, then came up multi-camera systems where the field of view became extended and non-overlapping.

In this dissertation were analyzed algorithms that solve the problem of motion estimation in 6 degrees of freedom by using multi-cameras rigidly coupled systems with non-overlapping fields of view. During the study of these methods was developed an application in `Matlab`, that allows recreate generics situations that can be resembled to real cases. This simulator has a wide range of options that can be predefined by the user. This is very useful for the implementation of algorithms in the area of computer vision and mobile robotics.

The methods described in this dissertation cover a broad knowledge of computer vision, in particular concepts of: rigid transformations; epipolar geometry; projection points; estimation of a fundamental and essential matrix, and others. All methods mentioned above were described in the thesis.

In the implementation of the algorithms in simulator, were considered generic movements in order to obtain a clear and precise study about the influence of noise in the pose estimation. Experiments show the performance study of the methods and suggest the most appropriated for use in real applications.

Keywords: Visual Odometry, Multi-Camera Systems, Non-Overlapping Fields of View, 6 Degrees of Freedom, Pose Estimation, Scale Factor, Projection Points.

Resumo

A presente dissertação tem como foco principal o estudo da Odometria Visual, utilizando sistemas de múltiplas câmaras com campos visuais não sobrepostos. A estimação do movimento é uma das tarefas mais importantes na investigação em visão por computador.

Durante a última década houve um grande desenvolvimento no estudo da Odometria Visual. Inicialmente, foram utilizados sistemas constituídos por apenas uma câmara combinada com outro tipo de sensores e só mais tarde, surgiram os clássicos sistemas de "*stereo pair*", que tornaram a estimativa de movimento mais robusta. Uma das limitações destes sistemas é o campo visual observado ser o mesmo, por isso, surgiram assim sistemas de múltiplas câmaras especificamente orientadas para que o campo visual seja mais alargado e não sobreposto.

No âmbito deste tema, foram analisados algoritmos que solucionam o problema da estimação do movimento em 6 graus de liberdade, utilizando sistemas de múltiplas câmaras rigidamente acopladas, com campos visuais não sobrepostos. No estudo deste métodos, foi desenvolvido uma aplicação em `Matlab` que permite recriar situações genéricas que possam ser equiparadas a situações reais. Este simulador possuí uma ampla gama de opções que podem ser pré-definidas pelo utilizador, sendo bastante útil para a implementação de algoritmos na área de visão por computador em conjunto com a robótica móvel.

Os métodos descritos na dissertação, envolvem um conhecimento alargado sobre visão por computador, nomeadamente, conceitos sobre transformações rigidas, geometria epipolar, projecção de pontos, estimação da matriz fundamental e essencial, entre outros. Todos os métodos referidos anteriormente foram abordados na tese.

Na implementação dos algoritmos no simulador, foram considerados vários movimentos genéricos de modo a obtermos um estudo claro e preciso sobre a influência do ruído na estimação da posição. As experiências realizadas traduzem uma análise da *performance* dos métodos e permite concluir qual o mais adequado para aplicação em situações reais.

Palavras-chave: Odometria Visual, Sistema de Múltiplas Câmaras, Campos Visuais Não Sobrepostos, 6 Graus de Liberdade, Estimação da Posição, Factor de Escala, Projecção de Pontos.

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

Nesta dissertação é analisado o tema de Odometria Visual inerente aos problemas de estimação de movimento de um sistema de múltiplas câmaras. Um sistema constituído por várias câmaras apresenta vários benefícios na estimação de movimentos com seis graus de liberdade e na reconstrução 3D do ambiente envolvente. A implementação destes sistemas está presente em várias aplicações reais, mas existem ainda poucos estudos sobre estes métodos, havendo um crescente interesse nos últimos anos.

O termo Odometria Visual surgiu em 2004 e foi introduzido por *Nister, D.* [18]. A expressão foi escolhida por ser semelhante à odometria utilizando as rodas, que se traduz na estimação incremental do movimento de um veículo, integrando o número de pulsos ao longo do tempo. Na Odometria Visual, a posição é estimada e incrementada ao longo de um percurso com base nas alterações entre as *frames* que as câmaras acopladas ao objecto em movimento capturam.

Durante a última década, a área da visão por computador relacionada com a estimação do movimento foi desenvolvendo novas técnicas, surgindo a necessidade de construção de sistemas visuais mais precisos na estimação. A grande vantagem da Odometria Visual em relação a outros sistemas de estimação de movimento, por exemplo, odometria através das rodas, é ser um método mais robusto, pois não existem problemas normalmente associados aos outros sistemas, como por exemplo o deslizamento das rodas, o tipo de terreno, entre outras condições adversas. Assim, a Odometria Visual torna-se um excelente complemento adicional a outros métodos de navegação já existentes, como o *Global Position System (GPS)*, o *Inertial Measurement Units (IMUs)*, e o *Laser Rangefinder Odometry*.

O problema de estimação do movimento utilizando informações visuais surgiu em meados de

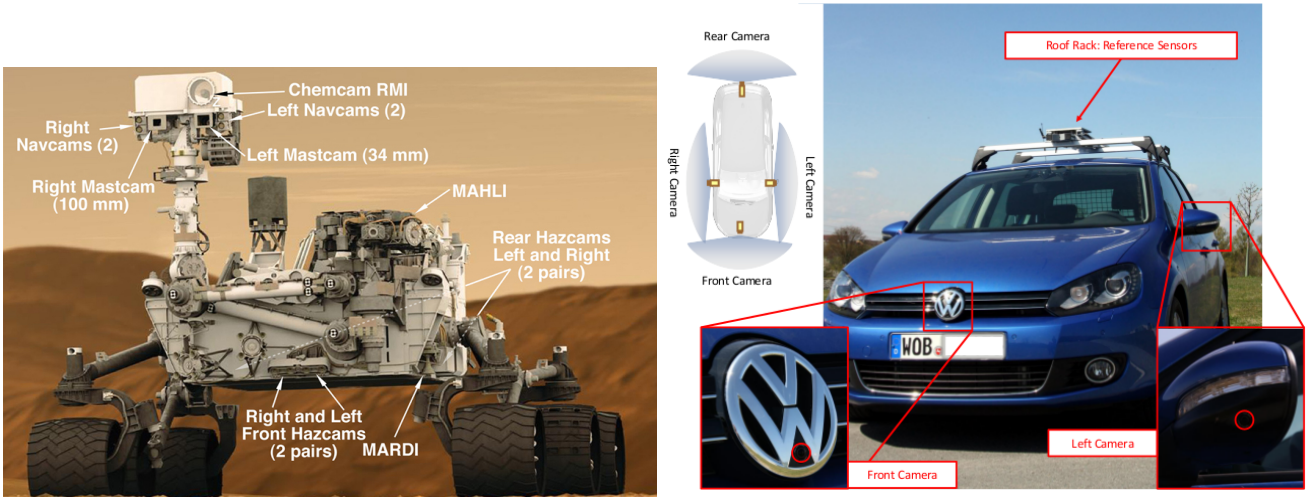
1980 e foi descrito por *Movarec, H.* [15]. A maioria das pesquisas em Odometria Visual [15, 11, 20] foram impulsionadas pelo programa da *NASA, Exploration Mars*, na tentativa de fornecer um outro método ao robô para estimar a sua posição em seis graus de liberdade. Uma vez que devido à presença de um terreno muito irregular sujeito a deslizamentos das rodas haveria certamente erros nas estimações, se fossem apenas utilizados os métodos tradicionais. A sonda *Curiosity (rover)*, que aterrou em Marte a 6 de Agosto de 2012, possui dezassete câmaras que têm sido usadas para estimar a sua posição, Fig. 1.1(a).

Mais recentemente, *Nister, D.* propôs a Odometria Visual através de sistemas constituídos por uma câmara e sistemas de "*stereo pair*" [18]. A configuração "*stereo pair*"apesar de tornar a estimação de movimento mais robusta, tem a desvantagem de estar limitada pelo campo visual das câmaras observarem a mesma cena. É um facto conhecido que através de uma só câmara é possível estimar o movimento, mas apenas a menos de um factor de escala [6]. Para determinar a escala utilizando uma câmara é necessário um sistema de informação adicional utilizando, por exemplo, um sensor inercial, como foi descrito por Nützi [19].

Nesta dissertação o foco de estudo centra-se na estimação métrica da escala utilizando câmaras adicionais, o que provoca um aumento do campo visual e da robustez do sistema. Várias configurações foram apresentadas ao longo dos anos, como o clássico sistema de "*stereo pair*", que usa a transformação conhecida entre as câmaras para estimar a escala [18, 10]. No artigo [2] *Clipp, B.*, introduziu um método que utilizava campos visuais muito amplos e com o mínimo de sobreposição possível. Hoje em dia, o estudo prossegue para sistemas constituídos por câmaras que não observam a mesma cena totalmente, ou seja, campos visuais não sobrepostos.

Nos últimos 20 anos, apareceram inúmeros estudos sobre a captura de imagens através de sistemas de múltiplas câmaras [6, 3]. No entanto, ainda há muito por explorar sobre estes sistemas pois para além de se usarem várias configurações, há tarefas desafiadoras como a obtenção da posição relativa das câmaras no meio em que se inserem. A estimação métrica através de câmaras com campos visuais não sobrepostos pode ser feita de duas maneiras: utilizando conjuntos de várias câmaras, mas considerando cada uma como individual [1, 8, 12], ou usando apenas uma câmara segundo o *Generalized Camera Model (GCM)*, [5, 16, 13, 9], onde existem vários centros de projeção para o mesmo ponto no mundo. Na Fig. 1.1(b), está representado um sistema de múltiplas câmaras utilizado em [12].

Uma vez que a escala métrica não pode ser determinada usando apenas uma câmara [6], são usados sistemas compostos por várias câmaras rigidamente acopladas e especificamente orien-



(a) **(b)**

Figura 1.1: Na Fig. (a) é representado o robô *Curiosity (rover)*, desenvolvido pela *NASA*, no âmbito do programa *Exploration Mars* onde podemos ver as dezassete câmaras que têm sido utilizadas na Odometria Visual. Na Fig. (b) é apresentado o sistema de múltiplas câmaras acopladas a um carro usadas para estimação da posição. As imagens foram retiradas da *NASA* e do artigo [12], respectivamente.

tadas criando campos visuais não sobrepostos, o que torna o problema mais complicado face aos sistemas clássicos de "*stereo pair*". Seguidamente será feito um estudo detalhado sobre este problema analisando os algoritmos desenvolvidos em [1, 8].

1.1 Breve resumo do trabalho desenvolvido

Nesta tese, foi estudado o problema de estimação da posição relativa, usando sistemas de múltiplas câmaras com campos visuais não sobrepostos.

Dados os parâmetros das câmaras, capturam-se sequências de imagens, neste caso pontos 3D, usando um sistema de múltiplas câmaras. Através dos pontos projectados no plano imagem de cada câmara e analisando as *frames* adquiridas, podemos estimar o movimento relativo do sistema de múltiplas câmaras para os seis graus de liberdade.

O trabalho desenvolvido, incidiu assim na criação de um ambiente de simulação, capaz de reproduzir um sistema real de navegação para múltiplas câmaras, onde podem ser implementados e validados vários algoritmos na área de visão por computador.

Foram testados dois métodos na área da Odometria Visual, em que a característica intrínseca

ao sistema de múltiplas câmaras é os campos visuais não serem sobrepostos, o que assinala uma diferença face aos sistemas clássicos de "*stereo pair*". O problema da estimação métrica do factor de escala abordado nos dois algoritmos, e através das simulações realizadas na aplicação, torna possível apresentar conclusões quanto ao sucesso da implementação dos mesmos em situações reais.

1.2 Motivação, Objectivos e Contribuições

A Odometria Visual é uma área de estudo muito importante nos dias de hoje e tem-se observado um crescimento da investigação nesse sentido. A utilização de câmaras no mundo tecnológico é uma área que possui um conjunto muito diverso de aplicações e sem dúvida um campo científico em grande ascensão. A Odometria Visual tem assim uma diversidade de aplicações, procurando-se cada vez mais o desenvolvimento de métodos robustos de estimação para serem aplicados em ambientes reais, tornando assim a odometria uma fonte informação segura e fiável. Estando a ciência a evoluir no sentido do desenvolvimento de carros totalmente autónomos, a Odometria Visual encontra assim cada vez mais sistemas onde poderá ser directamente aplicada. A robótica móvel abre assim as portas a novos sistemas de estimação de posição, o que permite adicionar aos já existentes novas técnicas que quebrem as limitações impostas por certos tipos de tecnologia, como por exemplo, o *Global Position System (GPS)*, que apenas funciona em ambientes exteriores.

O objectivo principal desta dissertação incidiu assim em testar a funcionalidade de dois algoritmos de Odometria Visual já existentes, [1, 8], e desenvolver um estudo sobre uma possível aplicação dos mesmos em sistema reais. Esta tese reúne assim as seguintes contribuições:

- Criação e desenvolvimento de um simulador, que permite o teste de uma grande variedade de algoritmos, e que possui um conjunto de características adaptáveis à realização de simulações para a estimação de posição, na área de robótica móvel recorrendo a visão por computador, podendo sempre ser feita a comparação e validação com a informação *Ground Truth*;

- Análise da robustez de dois algoritmos de estimação de posição relativa, usando campos visuais não sobrepostos [1, 8] em ambientes controlados e avaliação do impacto da inserção de ruído na estimação da posição;

1.3 Estrutura da Dissertação

No Capítulo 1 está presente uma introdução ao tema de Odometria Visual e a sua importância na estimação do movimento quando usado em aplicações reais. No Capítulo 2 existe uma breve referência aos principais aspectos teóricos que servem como base para a compreensão da dissertação. No Capítulo 3 é apresentada a estrutura do simulador desenvolvido bem como as suas especificações funcionais. Os algoritmos de Odometria Visual alvos de estudo estão descritos detalhadamente no Capítulo 4. Os resultados experimentais e todos testes realizados encontram-se presentes no Capítulo 5. No Capítulo 6 são apresentadas as conclusões e o trabalho futuro.

Capítulo 2

Sistema de Câmaras

2.1 Modelo de Câmaras

Nesta secção revemos: a transformação rígida entre câmaras; a projecção directa de pontos 3D do mundo para o plano de imagem; e a geometria epipolar associada a duas imagens, capturadas em localizações diferentes. Serão ainda abordados os métodos para a estimação da matriz Essencial e Fundamental.

2.1.1 Transformação Rígida das Câmaras

Na Fig. 2.1, é ilustrado o referencial da câmara e a orientação do referencial mundo. Inicialmente, a câmara é posicionada de acordo com o sistema de coordenadas do mundo que corresponde à origem do referencial. A matriz de transformação da câmara na posição inicial corresponde assim a:

$$\mathbf{P} = [\mathbf{I} \,|\, \mathbf{0}], \tag{2.1}$$

onde $\mathbf{I} \in \mathbb{R}^{3\times3}$ representa a matriz identidade. Se a câmara se mover para um ponto $\mathbf{c}$ sem rotação em relação ao sistema do mundo, $\mathbf{c} = [c_x, c_y, c_z]^\top$ que corresponde ao centro da câmara, a nova transformação é dada por:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \end{bmatrix}. \tag{2.2}$$

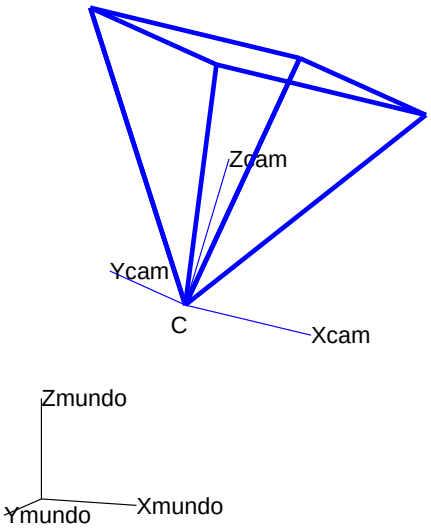


Figura 2.1: Sistema de coordenadas da câmara $\{x_{cam}, y_{cam}, z_{cam}\}$ e representação do sistema de coordenadas do mundo $\{x_{mundo}, y_{mundo}, z_{mundo}\}$.

Considerando agora que há rotação entre o sistema de coordenadas da câmara e do mundo, a transformação é representada por:

$$\mathbf{P} = \mathbf{R}[\mathbf{I}|-\mathbf{c}] = [\mathbf{R}|-\mathbf{Rc}] = [\mathbf{R}|\,\mathbf{t}], \tag{2.3}$$

onde $\mathbf{t} = -\mathbf{Rc}$, é um vector que representa a translação.

2.1.2 Projeção dos pontos na câmara

A projeção no plano imagem de um ponto $\mathbf{x} \in \mathbb{R}^3$ no espaço 3D pode ser descrita de várias formas. O modelo de perspectiva é o mais utilizado e que melhor descreve este processo de formação da imagem. Assumindo que o eixo z da câmara corresponde ao eixo óptico, o ponto $\mathbf{x}$ é projectado para um ponto $\mathbf{u} \in \mathbb{R}^3$ no plano imagem através da matriz de transformação da câmara $\mathbf{P} \in \mathbb{R}^{3\times4}$, usando:

$$\mathbf{u} \sim \underbrace{\mathbf{K}\,[\mathbf{R}|-\mathbf{Rc}]}_{\mathbf{P}} \begin{bmatrix} \mathbf{x} \\ \mathbf{1} \end{bmatrix}, \tag{2.4}$$

onde $\mathbf{K} \in \mathbb{R}^{3\times3}$ corresponde à matriz de parâmetros intrínsecos:

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.5}$$

f_x, f_y representam a distância focal expressa em *pixels*, s o *skew* (relacionado com o ângulo entre os eixos da imagem, que na grande maioria dos casos é igual a zero), e c_x, c_y o ponto principal da

imagem (o ponto correspondente ao *pixel* no plano imagem que é intersectado pelo eixo óptico da lente) [6]. Assim, **u** corresponde aos pontos 2D no plano imagem da câmara, expressos em *pixels*.

2.2 Geometria Epipolar

A geometria epipolar está associada aos sistemas de "*stereo pair*"onde, em posições distintas, as duas câmaras contêm campos de visão sobrepostos [23]. As relações geométricas que se podem obter entre os pontos 3D e as suas projeções representam um conjunto de propriedades que serão abordadas de seguida.

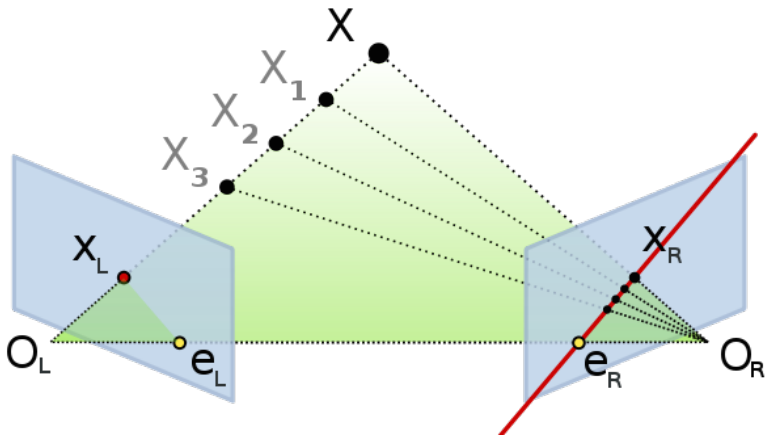


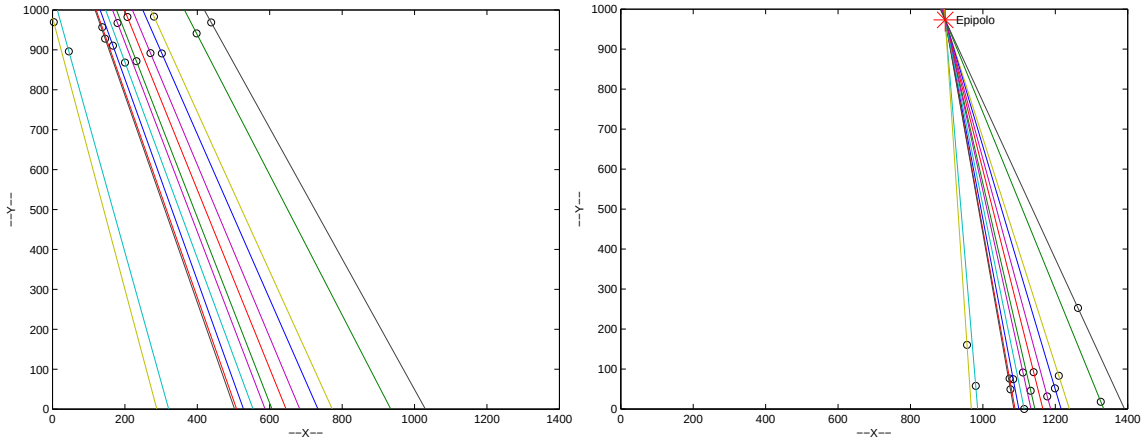
Figura 2.2: Representação da geometria epipolar [6]. O ponto X representa o ponto 3D, sendo x_L e x_R a sua projeção na câmara *left* e *right*, respectivamente. O_L e O_R, representam os centros das câmaras, e os pontos e_L e e_R são os epipolos associados a cada uma. A linha definida por x_R e e_R denomina-se como linha epipolar. O triângulo formado pelos centros das câmaras e pelo ponto 3D é chamado de plano epipolar.

Considerando-se duas câmaras colocadas em posições diferentes no espaço 3D representado, por exemplo Fig. 2.2, qualquer ponto **x** define um plano que contém esse mesmo ponto e os centros de projeção O_L e O_R de ambas as câmaras, este plano denomina-se como plano epipolar. Os pontos $\mathbf{u}_L$ e $\mathbf{u}_R$ representam a projecção do ponto **x**, nas câmaras *left* e *right*, respectivamente. Os sistemas de coordenadas referentes a cada câmara estão relacionados através de uma transformação de corpo rígido, a qual é definida por uma rotação **R** e translação **t**, tal que:

$$\mathbf{x}_R = \mathbf{R}\left(\mathbf{x}_L - \mathbf{t}\right), \tag{2.6}$$

onde $\mathbf{x}_L$ e $\mathbf{x}_R$ representam as coordenadas do ponto $\mathbf{x}$ nos referênciais da câmara *left* e *right*, respectivamente.

As linhas de interseção do plano epipolar com o plano imagem são definidas como linhas epipolares. A geometria epipolar é bastante útil para a computação de correspondências entre pontos nas imagens. Dado um ponto na imagem *left*, $\mathbf{u}_L$, o ponto na imagem *right*, $\mathbf{u}_R$, encontra-se na respectiva linha epipolar, não sendo necessário a pesquisa em toda a imagem.



(a) **(b)**

Figura 2.3: Nas Figs. (a) e (b) estão representadas as linhas epipolares, o epipolo e os pontos 2D correspondentes entre a (*frame 1* ↔ *frame 2*). As *frames* foram adquiridas utilizando apenas uma câmara, que ao sofrer um movimento rígido se comporta como um sistema "*stereo pair*", apresentado na Fig. 2.2.

Na Fig. 2.3, está representado o epipolo e as linhas epipolares referentes aos pontos da (*frame 1* ↔ *frame 2*). Neste caso, foi usada apenas uma câmara que sofreu uma transformação de corpo rígido, durante a aquisição de duas frames.

2.3 Estimação da Matriz Essencial

A matriz essencial $\mathbf{E}$ ($\mathbb{R}^{3\times3}$) é usada quando as câmaras se encontram calibradas, ou seja, quando são conhecidos os parâmetros intrínsecos das câmaras. Os vectores $\mathbf{x}_L$, $\mathbf{t}$ e $\mathbf{x}_L - \mathbf{t}$, que estão contidos no plano epipolar, são definidos no sistema da câmara *left*, que pertencem ao mesmo plano e satisfazem a seguinte equação:

$$(\mathbf{x}_L - \mathbf{t})^\top (\mathbf{t} \times \mathbf{x}_L) = 0 \tag{2.7}$$

Da equação 2.6 resulta,

$$\mathbf{x}_L - \mathbf{t} = R^\top \mathbf{x}_R \tag{2.8}$$

que substituindo na 2.7 obtêm-se,

$$(\mathbf{R}^\top \mathbf{x}_R)^\top (\mathbf{t} \times \mathbf{x}_L) = 0 \tag{2.9}$$

O produto vectorial pode ser expresso algebricamente por meio de um produto de uma matriz *Skew-Symmetric*,

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \tag{2.10}$$

tal que:

$$\mathbf{t} \times \mathbf{x}_L = [\mathbf{t}]_\times \mathbf{x}_L \tag{2.11}$$

Desta forma a equação 2.9, pode ser re-escrita:

$$\mathbf{x}_R^\top \mathbf{E} \mathbf{x}_L = 0 \tag{2.12}$$

onde resulta a matriz essencial $\mathbf{E}$ pode ser escrita como

$$\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times \tag{2.13}$$

2.3.1 Recuperar a Rotação R e a translação t através da Matriz Essencial

Através da matriz essencial $\mathbf{E}$ com característica 2, podemos recuperar a rotação $\mathbf{R}$, e a translação $\mathbf{t}$ (a menos de um factor de escala).

Definindo

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.14}$$

pelo teorema 1 [21], da decomposição em valores singulares *(SVD)* da matriz essencial resulta que $\mathbf{E} \sim \mathbf{U} diag(1,1,0) \mathbf{V}^\top$, onde $\mathbf{U}$ e $\mathbf{V}$ são escolhidos, tal que, $det(\mathbf{U}) > 0$ e $det(\mathbf{V}) > 0$. Assim $\mathbf{t} \sim \mathbf{t}_u \equiv [u_{13}\ u_{23}\ u_{33}]^\top$ e $\mathbf{R}$ é igual a $\mathbf{R}_a \equiv \mathbf{U}\mathbf{D}\mathbf{V}^\top$ ou $\mathbf{R}_b \equiv \mathbf{U}\mathbf{D}^\top\mathbf{V}^\top$. Qualquer combinação entre $\mathbf{R}$ e $\mathbf{t}$ resulta numa solução que satisfaz a restrição epipolar. Para resolver esta ambiguidade, onde temos quatro soluções possíveis para a posição da segunda câmara, é feita uma triangulação com um ponto $\mathbf{q}$ que nos ajuda a escolher a solução verdadeira, ver [17].

2.4 Estimação da Matriz Fundamental

A representação algébrica da geometria epipolar pode ser obtida a partir da matriz fundamental $\mathbf{F} \in \mathbb{R}^{3\times3}$. Considerando os pontos $\mathbf{v}_R$ e $\mathbf{v}_L$, representados em coordenadas homogéneas, correspondentes aos pontos num sistema "*stereo pair*", $\mathbf{F}\mathbf{v}_L$ representa a linha epipolar onde está a ponto $\mathbf{v}_R$. Dada a matriz $\mathbf{F}$,

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \tag{2.15}$$

e um par de pontos $\mathbf{v}_R = (v_1, v_2, v_3)^\top$ e $\mathbf{v}_L = (v'_1, v'_2, v'_3)^\top$, em coordenadas homogéneas, a partir da restrição epipolar obtemos a equação:

$$\mathbf{v}_R \mathbf{F} \ {\mathbf{v}'_L}^\top = 0 \tag{2.16}$$

As coordenadas dos pontos $\mathbf{v}_R = (\mathbf{u}_R, 1)$ e $\mathbf{v}_L = (\mathbf{u}_L, 1)$, em que $\mathbf{u}_R, \mathbf{u}_L$ são dados em *pixels* e podem ser calculados usando a equação 2.17, onde $\mathbf{K}_R$ e $\mathbf{K}_L$, correspondem às matrizes de parâmetros intrínsecos das câmaras *right* e *left*, respectivamente

$$\mathbf{v}_R \sim \mathbf{K_R}\mathbf{x}_R \ \text{ e } \ \mathbf{v}_L \sim \mathbf{K}_L\mathbf{x}_L \tag{2.17}$$

Através da equação 2.16 e 2.17, obtemos $\mathbf{F}$,

$$\mathbf{F} = \mathbf{K}_R^{-\top}\mathbf{E}\mathbf{K}_L^{-1}. \tag{2.18}$$

onde $\mathbf{E}$, representa a matriz essencial. A matriz $\mathbf{F}$, tem característica 2.

Para a estimação da matriz fundamental podem ser usados vários algoritmos. De seguida, serão apresentados os três algoritmos que foram usados nesta dissertação.

2.4.1 *Eight-Point Algorithm*

Para a estimação da matriz fundamental pode ser usado o algoritmo dos 8 pontos, introduzido por *Longuet-Higgins* [14]. Contudo, este método não é muito usado na prática por ser muito sensível ao ruído. Assim, *Hartley*, [7], apresentou um método robusto de estimação da matriz fundamental, que consiste em normalizar as coordenadas dos pontos da imagem antes de aplicar o método dos oito pontos.

Considere-se, assim, um conjunto n (com $n \geqslant 8$) de correspondências nas imagens *left* e *right*, onde cada par de pontos satisfaz a equação 2.16. Dessa equação resultam nove incógnitas. No entanto, como se trata de uma equação homogénea, esse grau de liberdade reduz o número de incógnitas para oito. Assim, para estimar $\mathbf{F}$ no sentido dos mínimos quadrados, podemos impor a restrição em que a norma do vector de incógnitas é unitária e usar a *Singular Value Decompositon (SVD)*.

2.4.2 *RANSAC Algorithm*

O algoritmo *RANdom SAmple Consensus (RANSAC)*, proposto por *Fischler e Bolles*, possibilita o cálculo robusto da matriz $\mathbf{F}$, permitindo distinguir os pontos correspondentes como *inliers* e *outliers* [4].

Partindo de um subconjunto de pontos $\mathbf{u}_R$ e $\mathbf{u}_L$, calcula-se uma primeira estimativa da matriz $\mathbf{F}$ usando o algoritmo dos 8 pontos e determinam-se os *inliers* e *outliers*, do restante conjunto de pontos. Quanto menor for o subconjunto de pontos escolhido, a probabilidade de estarem presentes *outliers* é menor. Repete-se todo o processo até se obter um número de *inliers* satisfatório, e refina-se a estimação da matriz $\mathbf{F}$ com base nesses pontos.

2.4.3 *LMedS Algorithm*

O algoritmo *Least Median of Squares (LMedS)* difere do anterior, na medida em que a melhor estimação é aquela que minimiza a média dos erros r_i, [22], definidos na equação:

$$r_i^2 \;=\; d^2(\mathbf{v}_R^i, \mathbf{F}\mathbf{v}_L^i) \;+\; d^2(\mathbf{v}_L^i, \mathbf{F}^\top \mathbf{v}_R^i) \tag{2.19}$$

onde a cada correspondência é atribuído o seguinte peso w_i:

$$w_i = \begin{cases} r_i^2 & : r_i^2 \leqslant t^2 \\ 0 & : outros\, casos \end{cases} \tag{2.20}$$

A matriz fundamental $\mathbf{F}$, pode ser calculada resolvendo o seguinte problema de minimização.

$$\text{argmin}_{\mathbf{F}} \;=\; \sum_i w_i r_i^2. \tag{2.21}$$

Capítulo 3

Simulador

Neste capítulo é apresentado o simulador desenvolvido em `Matlab`, assim como todas as suas características e aplicações. O objectivo deste simulador é ser uma ferramenta auxiliar para a realização de testes onde a sua dinâmica permite ao utilizador a implementação de vários algoritmos. Dada a sua versatilidade, as configurações são definidas sempre pelo utilizador que escolhe os vários parâmetros envolvidos nas simulações, podendo facilmente ser adaptado e expandido para outros casos. O simulador é composto por um sistema de múltiplas câmaras, que se movem rigidamente no espaço 3D, onde existe uma "nuvem"de pontos.

3.1 Design

Na Fig. 3.1, é representado o ambiente gráfico do simulador. A parte 1 corresponde ao mundo 3D, constituído por pontos 3D (gerados aleatoriamente) e câmaras. A parte 2 destaca o menu do simulador, com uma ampla quantidade de funções que permitem, entre muitas outras opções, por exemplo, carregar modelos de simulação pré-definidos. Na parte 3, temos a informação gráfica onde podemos observar o plano imagem de cada câmara durante o movimento, assim como a sua posição, velocidade e aceleração ao longo do tempo.

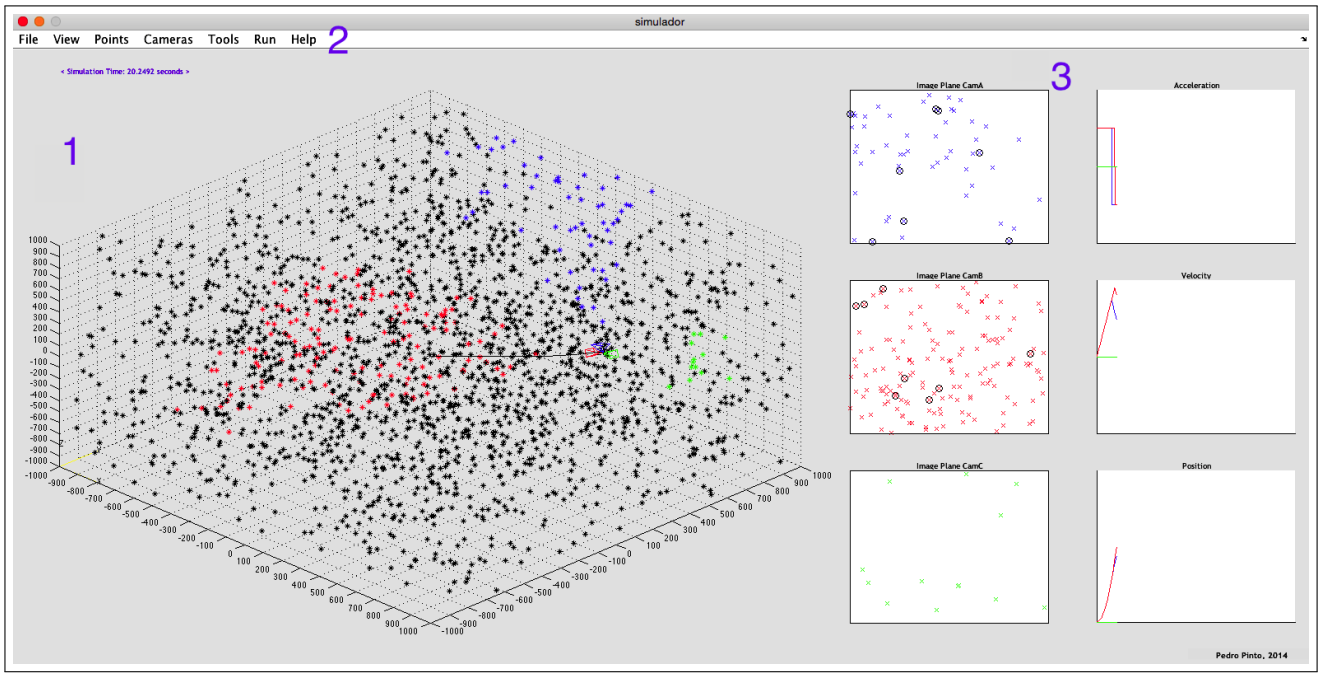


Figura 3.1: Design do Simulador. Na parte 1 está representada a zona de simulação e navegação das câmaras, correspondente ao mundo 3D. A parte 2 é referente ao menu, onde existe um conjunto de opções para o utilizador personalizar as suas simulações. Na parte 3 destaca-se a informação gráfica auxiliar, onde pode ser observado o plano imagem de cada câmara e os aspectos associados ao seu movimento.

3.2 Sistema de Câmaras

Para o estudo dos algoritmos, que vão ser analisados posteriormente, foi considerado o sistema de câmaras[1] *pinhole* representado na Fig. 3.2.

De acordo com a matriz de transformação genérica $\mathbf{T} \in \mathbb{R}^{4\times4}$:

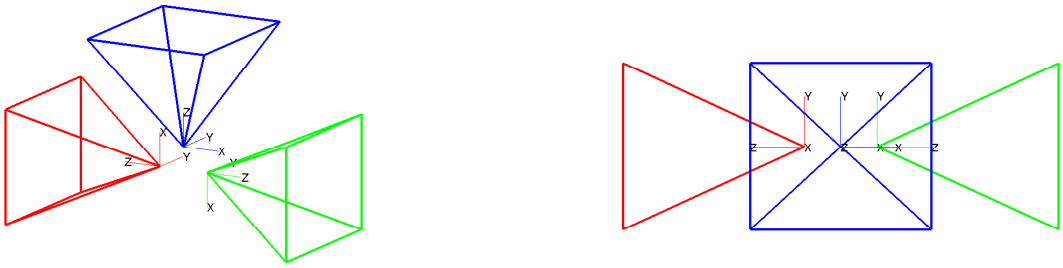
$$\mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.1}$$

onde r_{ij} são os elementos da matriz de rotação, e t_x, t_y, e t_z os parâmetros de translação.

As três câmaras estão acopladas rigidamente e têm as seguintes matrizes de transformação:

$$\mathbf{T}_{cam_A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{T}_{cam_B} = \begin{bmatrix} 0 & 0 & -1 & -10 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{T}_{cam_C} = \begin{bmatrix} 0 & 0 & 1 & 10 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

[1] A - Azul, B - Vermelha e C - Verde.



(a) **(b)**

Figura 3.2: Nas Figs. (a) e (b) está representado o sistema de câmaras em duas vista 3D e 2D, respectivamente. O sistema é constituído por três câmaras rigidamente acopladas e orientadas especificamente para que o campo visual não esteja sobreposto.

Como já foi referido anteriormente, podem introduzir-se alterações em várias características. Este sistema pode ser livremente adaptado, acrescentando-se novas câmaras e definindo novas posições, bem como a atribuição dos parâmetros intrínsecos, que está ao cuidado do utilizador. Os parâmetros intrínsecos usados foram (2.5):

$$\mathbf{K}_{cam_{A,B,C}} = \begin{bmatrix} 1729.8 & 0 & 703.6 \\ 0 & 1733.5 & 550.9 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.3}$$

3.3 Projeção dos Pontos 3D

O mundo é composto por pontos 3D, gerados aleatoriamente dentro de uma gama definida. Com as câmaras numa posição conhecida, através dos seus parâmetros intrínsecos e extrínsecos, é possível obter a projeção dos pontos 3D para cada um dos planos imagem. Em ambiente de programação, os pontos 3D gerados são incluídos numa *struct*, com o seguinte formato:

Código Matlab 3.1: *Struct* dos Pontos 3D.

```
st_points3D=struct('index',n',... %Indice
                   'point',[x' y' z'],... %Coordenadas do ponto 3D no mundo
                   'ca',a',... %Flag 1 se for visto na camera A
                   'point2Da',point2Da',... %Coordenadas do ponto 2D na camara A
                   'cb',b',...
                   'point2Db',point2Db',...
                   'cc',c',...
                   'point2Dc',point2Dc');
```

Deste modo e sabendo que

$$\mathbf{u}' \sim \mathbf{K}\,[\,\mathbf{R}\,|\,\mathbf{t}\,]\,\mathbf{x}', \tag{3.4}$$

temos sempre a informação das coordenadas dos pontos 3D e a projeção associada ao longo de todas as *frames* de movimento.

ou

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.5}$$

Na equação 3.4, os pontos 3D em coordenadas homogéneas representados no mundo, $\mathbf{x}$, são convertidos para as coordenadas da câmara, e seguidamente projectados na imagem 2D, segundo o modelo de perspectiva [6]. Para cada uma das câmaras foi definida uma janela de visualização de 1400 $\times$ 1000 *pixels*, como se pode observar na Fig. 3.3. Durante a simulação, pode ser observado o movimento dos pontos nos planos imagem, assim como a alteração de cor do ponto no plano mundo, respeitante à câmara que o está a visualizar naquele instante.

3.4 Gerador de Trajectórias

Na perspectiva de aumentar a interatividade e a utilização por outros utilizadores, foi projetado um gerador de trajetórias que permite definir movimentos específicos em 2D e 3D, onde intuitivamente se podem desenhar trajectos críticos, que ajudam a verificar a robustez de algoritmos que venham a ser implementados.

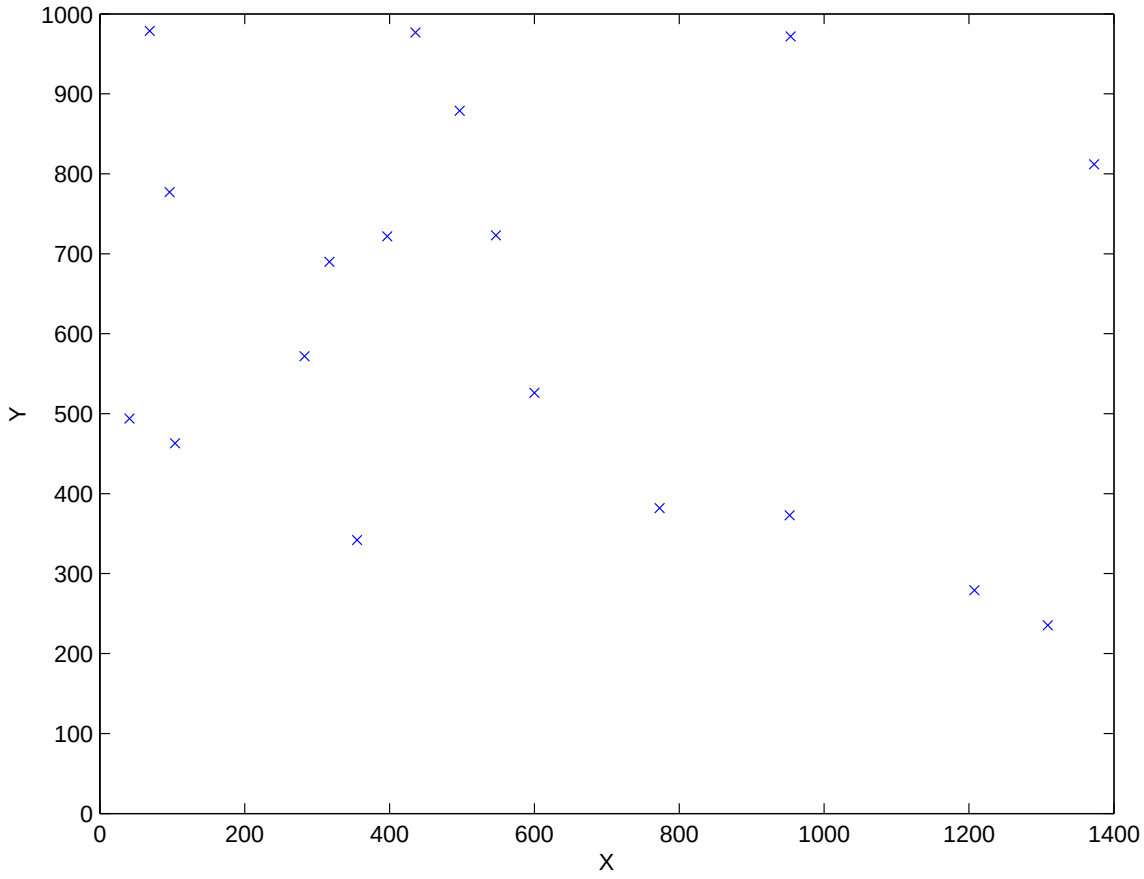
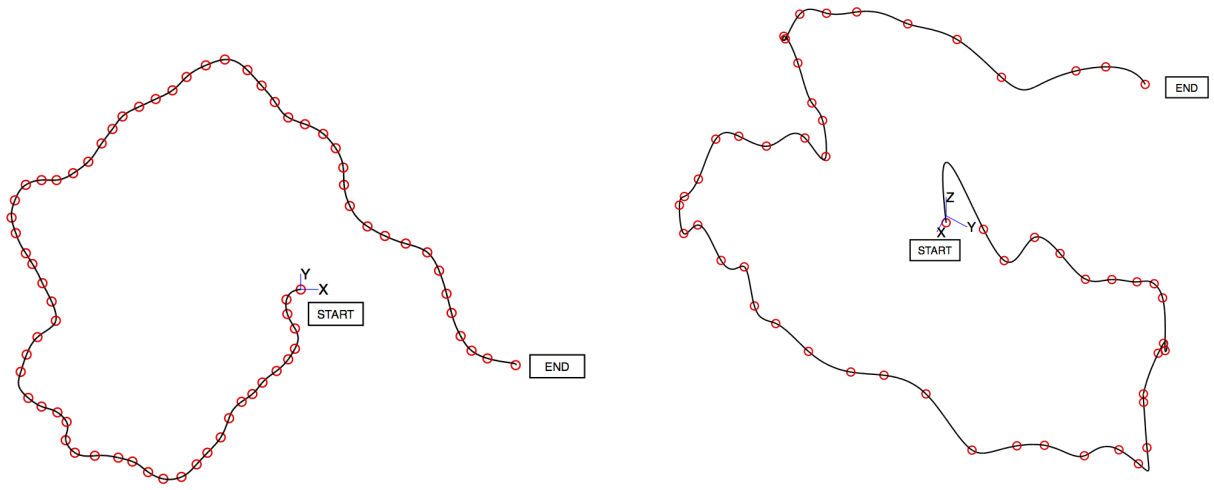


Figura 3.3: Representação do plano imagem associado à câmara A. Este plano imagem tem dimensão 1400 × 1000 *pixels*, e representa os pontos 2D vistos pela câmara A num determinado instante.

Outra alternativa é escolher uma posição final objectivo, e o sistema de câmaras deslocar-se-á até esse ponto segundo um movimento uniformemente acelerado, em que a velocidade de rotação e translação são um critério de escolha. Nesta aplicação são definidos pelo utilizador pontos de controlo com o *mouse*, onde este desenha o trajeto pretendido (para trajectórias 3D a coordenada z é gerada aleatoriamente dentro de uma gama). Depois de escolhidos os pontos, é gerada uma trajetória com recurso a *splines* de interpolação, que definem curvas suaves entre os pontos de controlo. Na Fig. 3.4 estão representadas duas trajetórias que podem ser integradas no simulador principal.

3.5 Outras Funcionalidades

O simulador possui uma basta gama de funcionalidades e características que o tornam bastante robusto e adaptável à realização de diferentes testes. Todas essas funcionalidades foram implementadas com intuito de facilitar o uso da aplicação por qualquer utilizador, auxiliando na



(a) **(b)**

Figura 3.4: Nas Figs. (a) e (b), está representado o projecto de duas trajectórias 2D e 3D, respectivamente. As trajectórias são projectadas pelo utilizador e podem ser usadas na aplicação como movimento pré-definido.

exportação e análise de dados, proporcionando conjuntamente uma interactividade em tempo real.

Segue-se uma lista das principais funções:

- Carregar simulações pré-definidas;
- Adicionar pontos manualmente através do *mouse*;
- Definir trajectórias;
- Escolher a posição e orientação das câmaras;
- Definir os parâmetros associados a cada câmara;
- Visualização em 2D e 3D.
- Exportação de gráficos.
- Gráficos de Posição, Velocidade e Aceleração em tempo real;
- Visualização do plano imagem associado a cada câmara em tempo real;
- Guardar *datasets* de simulações.

- Permite ao utilizador observar o movimento das câmaras e aplicação de algoritmos em tempo real.

Capítulo 4

Algoritmos de Odometria Visual

Neste capítulo serão apresentados os algoritmos implementados no simulador, que permitem a estimação relativa do movimento das câmaras. Os métodos analisados funcionam em tempo real, usando um sistema de câmaras com campos visuais não sobrepostos [1, 8].

De seguida, será apresentado um estudo detalhado sobre estes dois métodos.

4.1 *Robust 6DOF Motion Estimation for Non-Overlapping, Multi-Camera Systems*

4.1.1 Descrição

Em 2008, Clipp et al. [1] definiram um método robusto para a estimação do movimento (seis graus de liberdade) de um sistema de múltiplas câmaras, rigidamente acopladas. A técnica consiste na escolha de cinco pontos observados pela câmara A e apenas um pela câmara B. Supondo que o sistema está calibrado, este move-se de uma posição para outra (*frame 1* $\leftrightarrow$ *frame 2*), onde através da geometria epipolar, a matriz essencial $\mathbf{E}$ pode ser estimada usando o algoritmo proposto por *Nistér*, [17]. A rotação e translação a menos de um factor de escala podem ser assim extraídas da matriz essencial, obtendo cinco graus de liberdade do movimento. Por fim, para completar estimação do movimento, falta apenas estimar o factor de escala associado à translação.

De seguida, são apresentadas as equações necessárias para estimar os seis graus de movimento.

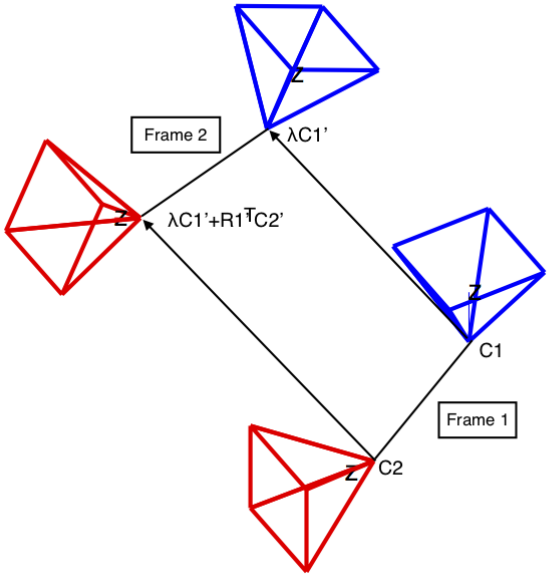


Figura 4.1: Representação dos centros das câmaras entre a (*frame 1* ↔ *frame 2*), onde existe um movimento rígido composto por uma rotação $\mathbf{R}$ e translação $\mathbf{t}$. Este sistema de duas câmaras com campos visuais não sobrepostos foi usado na implementação e simulações do algoritmo [1].

Na Fig. 4.1 está representado o sistema de câmaras utilizado para a aplicação do método.

4.1.2 Algoritmo

Sendo $\mathbf{P}_1$ e $\mathbf{P}_2$ a matriz de transformação das câmaras A e B, respectivamente. Pode ainda definir-se $\mathbf{P}_1 = [\mathbf{I}\,|\,\mathbf{0}]$ e $\mathbf{P}_2 = [\mathbf{R}_2\,|\,-\mathbf{R}_2\mathbf{c}_2]$. Movendo o sistema para outra posição, o centro da câmara A após o movimento é dado por $\lambda\mathbf{c}'_1$. Assim, o objectivo imediato é determinar a matriz de transformação para a câmara B, depois do movimento, $\mathbf{P}'_2$. As matrizes $\mathbf{P}'_1$ e $\mathbf{P}'_2$ representam as matrizes de transformação da câmara A e B, respectivamente, após a transformação euclidiana entre as *frames*. Considerando, $\mathbf{P}'_1$ tal que

$$\mathbf{P}'_1 = [\mathbf{I}\,|\,\mathbf{0}] \begin{bmatrix} \mathbf{R}'_1 & -\lambda\mathbf{R}'_1\mathbf{c}'_1 \\ \mathbf{0}^\top & \mathbf{1} \end{bmatrix} = \mathbf{P}_1\mathbf{T}, \tag{4.1}$$

onde $\mathbf{T}$ representa a transformação Euclidiana do movimento, é possível obter $\mathbf{P}'_2$:

$$\begin{aligned}\mathbf{P}'_2 &= \mathbf{P}_2\mathbf{T} \\ &= [\mathbf{R}_2 \,|\, -\mathbf{R}_2\mathbf{c}_2]\begin{bmatrix}\mathbf{R}'_1 & -\lambda\mathbf{R}'_1\mathbf{c}'_1 \\ \mathbf{0}^\top & \mathbf{1}\end{bmatrix} \\ &= [\mathbf{R}_2\mathbf{R}'_1 \,|\, -\lambda\mathbf{R}_2\mathbf{R}'_1\mathbf{c}'_1 - \mathbf{R}_2\mathbf{c}_2] \\ &= \mathbf{R}_2\mathbf{R}'_1[\mathbf{I} \,|\, -(\lambda\mathbf{c}'_1 + \mathbf{R}'^\top_1\mathbf{c}_2)].\end{aligned} \tag{4.2}$$

Conjugando $\mathbf{P}_2$ e $\mathbf{P}'_2$, calculamos a matriz essencial, $\mathbf{E}_2$, para a câmara B usando

$$\begin{aligned}\mathbf{E_2} &= \mathbf{R}_2\mathbf{R}'_1[\lambda\mathbf{c}'_1 + \mathbf{R}'^\top_1\mathbf{c}_2 - \mathbf{c}_2]_\times\mathbf{R}_2^\top \\ &= \mathbf{R}_2\mathbf{R}'_1[\mathbf{R}'^\top_1\mathbf{c}_2 - \mathbf{c}_2]_\times\mathbf{R}_2^\top + \lambda\mathbf{R}_2\mathbf{R}'_1[\mathbf{c}'_1]_\times\mathbf{R}_2^\top \\ &= \mathbf{A} + \lambda\mathbf{B}.\end{aligned} \tag{4.3}$$

Sendo assim, dada uma correspondência entre um único ponto visto na câmara B durante o movimento entre duas frames $\mathbf{x}' \leftrightarrow \mathbf{x}$, a seguinte restrição pode ser definida

$$\mathbf{x}'^\top\mathbf{A}\mathbf{x} + \lambda\mathbf{x}'^\top\mathbf{B}\mathbf{x} = 0, \tag{4.4}$$

tal que:

$$\mathbf{A} = \mathbf{R}_2\mathbf{R}'_1\,[(\mathbf{R}'^\top_1 - \mathbf{I})\mathbf{c}_2]_\times\mathbf{R}_2^\top, \tag{4.5}$$

e

$$\mathbf{B} = \mathbf{R}_2\mathbf{R}'_1\,[\mathbf{c}'_1]_\times\mathbf{R}_2^\top, \tag{4.6}$$

Na equação 4.5, 4.6 e 4.3, $[\mathbf{a}]_\times\mathbf{b}$ é o produto vectorial dado pela matriz *Skew-Symmetric*, 2.10. Através de $\mathbf{x}'^\top\mathbf{E}\mathbf{x} = 0$, obtemos a equação 4.7, que nos permite determinar o valor do factor de escala da translação, λ:

$$\lambda = -\frac{\mathbf{x}'^\top\mathbf{A}\mathbf{x}}{\mathbf{x}'^\top\mathbf{B}\mathbf{x}}. \tag{4.7}$$

O algoritmo 1 descreve detalhadamente a implementação do método, utilizando as equações anteriores.

Algorithm 1: *Robust 6DOF Motion Estimation Algorithm*

Inputs:

struct{*pontos*2*D*}, projectados na câmara A e câmara B (*frame 1* ↔ *frame 2*);

$\mathbf{K} \in \mathbb{R}^{3\times3}$ é a matriz de parâmetros intrínsecos das câmaras, ver 3.3.

Outputs:

Factor de escala λ.

Method:

1. Escolher no mínimo 5 correspondências da câmara A, utilizando o método *RANSAC* [4];
2. Calcular a Matriz Fundamental $\mathbf{F}$ para câmara A;
3. Obter a Matriz Essencial $\mathbf{E}$;
4. $SVD(E) = UDV^\top$, determinamos as 4 soluções para a transformação da câmara A, [17];
5. Selecionar a transformação correcta a menos de um factor de escala e calcular $\mathbf{P}'_1$ e $\mathbf{P}'_2$;
6. Calcular A e B com a equação 4.5 e 4.6;
7. Obter λ, através da equação 4.7.

4.2 *Real Time 6D Stereo Visual Odometry with Non-Overlapping Fields of View*

4.2.1 Descrição

Em 2012, Kazik et al. [8] propuseram um novo método de estimação do movimento das câmaras, usando um sistema composto por duas câmaras rigidamente acopladas, onde é conhecida a transformação entre as mesmas. A diferença deste método, para outros existentes na área de Odometria Visual, consiste na utilização de campos visuais não sobrepostos o que permite observar cenas diferentes em cada câmara, enquanto é estimada a escala absoluta. O algoritmo funciona em tempo real, onde as câmaras são colocadas nos espelhos laterais de um carro, fornecendo informação essencial para estimar a deslocação do veículo ao longo do tempo.

Assumindo um sistema rígido de duas câmaras calibradas (a transformação euclidiana entre a câmara *right* e *left* é conhecida), através da estimação individual do movimento realizado entre duas *frames* é possível obter os factores de escala associados *frame 1* ↔ *frame 2*. A análise aplicada individualmente para estimar o movimento é igual à de um sistema "*stereo pair*". Assim, sabendo $\mathbf{T}_{LR}$ e com base na estimação de $\mathbf{T}_{R2R1}$ e $\mathbf{T}_{L2L1}$ a menos de um factor de escala, temos

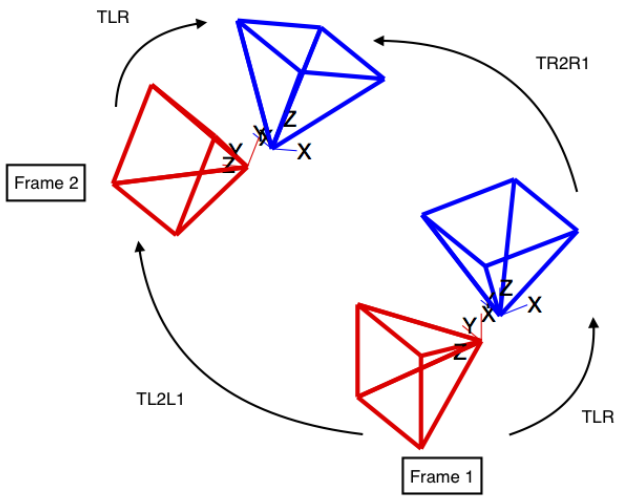


Figura 4.2: Representação das transformações euclidianas entre a (*frame 1* ↔ *frame 2*), ou seja, $\mathbf{T}_{R2R1}$ e $\mathbf{T}_{L2L1}$, e transformação rígida entre as duas câmaras dada por $\mathbf{T}_{LR}$. Este sistema de duas câmaras com campos visuais não sobrepostos foi usado na implementação e simulações do algoritmo [8].

os dados necessários para aplicação do algoritmo. Na Fig. 4.2 é possível visualizarmos o sistema de câmaras, considerado na implementação do algoritmo [8].

4.2.2 Algoritmo

Através da restrição rígida entre as duas câmaras é possível escrever:

$$\mathbf{T}_{L2L1}\,\mathbf{T}_{LR} \;=\; \mathbf{T}_{LR}\,\mathbf{T}_{R2R1}. \tag{4.8}$$

Expandindo as transformações euclidianas e introduzindo os factores de escala desconhecidos, λ e μ na câmara *right* e *left*, respectivamente, resulta a equação 4.9:

$$\begin{bmatrix} \mathbf{R}_{L2L1} & \mu_{L2}\mathbf{t}_{L2L1} \\ \mathbf{0}_{1\times 3} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{LR} & {}_{L}\mathbf{t}_{LR} \\ \mathbf{0}_{1\times 3} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{LR} & {}_{L}\mathbf{t}_{LR} \\ \mathbf{0}_{1\times 3} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{R2R1} & \lambda_{R2}\mathbf{t}_{R2R1} \\ \mathbf{0}_{1\times 3} & \mathbf{1} \end{bmatrix}. \tag{4.9}$$

As matrizes $\mathbf{R}_{R2R1}$ e $\mathbf{R}_{L2L1}$, e os vectores $\mathbf{t}_{R2R1}$ e $\mathbf{t}_{L2L1}$, correspondem às matrizes de rotação e vectores de translação da câmara A e B, respectivamente, estimadas entre a *frame 1* ↔ *frame 2*. Estas matrizes são obtidas após a estimação da Matriz Fundamental $\mathbf{F}$, utilizando o método *RANSAC* [4], e Matriz Essencial $\mathbf{E}$, onde recuperamos a rotação e a translação a menos de um factor de escala. Assim, decompondo as partes em rotação e translação obtemos a seguinte equação:

$$\begin{aligned} \mathbf{R}_{L2L1}\,\mathbf{R}_{LR} \;&=\; \mathbf{R}_{LR}\,\mathbf{R}_{R2R1}, \\ \mathbf{R}_{L2L1}\;{}_{L}\mathbf{t}_{LR} \;+\; \mu_{L2}\mathbf{t}_{L2L1} \;&=\; \mathbf{R}_{LR}\,\lambda_{R2}\mathbf{t}_{R2R1} \;+\; {}_{L}\mathbf{t}_{LR} \end{aligned} \tag{4.10}$$

que pode ser reescrita (isolando as incognitas λ e μ) como

$$\underbrace{\begin{bmatrix} \mathbf{R}_{LR}\,{}_{R2}\mathbf{t}_{R2R1} & -{}_{L2}\mathbf{t}_{L2L1} \end{bmatrix}}_{:=\mathbf{A_i}} \underbrace{\begin{bmatrix} \lambda \\ \mu \end{bmatrix}}_{:=\mathbf{x_i}} = \underbrace{(\mathbf{R}_{L2L1} - \mathbf{I}_{3\times 3})\;{}_{L}\mathbf{t}_{LR}}_{:=\mathbf{b_i}}. \tag{4.11}$$

Este sistema pode ser resolvido usando a pseudo-inversa de *Moore-Penrose*, determinando os fatores de escala relativos à transformação euclidiana das câmaras entre duas frames:

$$\mathbf{x}_i \;=\; \mathbf{A}_i^{\dagger}\mathbf{b}_i. \tag{4.12}$$

Com base nas equações anteriores, podemos assim desenvolver o algoritmo 2, que nos permite obter a estimação do movimento do sistema de câmaras, usando campos visuais não sobrepostos.

Algorithm 2: *Real Time 6D Stereo Visual Odometry Algorithm*

Inputs:

struct{*pontos*2*D*}, projectados na câmara A e câmara B (*frame 1* $\leftrightarrow$ *frame 2*);

$\mathbf{T}_{LR} \in \mathbb{R}^{4\times 4}$, representa a transformação rígida entre as câmaras;

$\mathbf{K} \in \mathbb{R}^{3\times 3}$ é a matriz de parâmetros intrínsecos das câmaras, ver 3.3.

Outputs:

Factor de escala $\lambda\, e\, \mu$.

Method:

1. Escolher no mínimo 5 correspondências da câmara A, e 5 correspondências da câmara B, utilizando o método *RANSAC* [4];
2. Calcular a Matriz Fundamental **F** para a câmara A e para a câmara B;
3. Calcular a Matriz Essencial **E** da câmara A e câmara B;
4. $SVD(E) = UDV^{\top}$, e obter as 4 soluções para a transformação das câmaras, [17];
5. Determinar as transformações correctas a menos de um factor de escala $\mathbf{T}_{R2R1}$ e $\mathbf{T}_{L2L1}$;
6. Aplicar as equações 4.8, 4.9, 4.10 e 4.11;
7. Obter $\lambda\, e\, \mu$, equação 4.12, resolvendo o sistema 4.11 utilizando a pseudo-inversa de *Moore-Penrose*.

Capítulo 5

Resultados Experimentais

Neste capítulo é feita uma análise detalhada aos algoritmos anteriormente mencionados, devidamente implementados no ambiente de simulação virtual desenvolvido e descrito na secção 3. O objectivo principal é estimar a posição relativa do sistema de câmaras no mundo, determinando os factores de escala associados a cada *frame* de movimento (com base nas matrizes essencial e fundamental só é possível obter as translações a menos de um factor de escala). O estudo destes algoritmos permitiu assim avaliar a potencialidade dos métodos em determinar a posição relativa, para vários tipos de trajectórias e movimentos.

Um dos pontos fundamentais visa compreender o efeito do ruído e quantificar o seu impacto na estimação do factor de escala. A inserção de ruído permite testar a robustez dos algoritmos e prever o seu comportamento quando aplicados em situações de simulação reais.

De seguida, serão apresentados os resultados obtidos para dois movimentos genéricos do sistema de câmaras, mais especificamente: um movimento planar (2D) e uma trajectória 3D, projectados no simulador. Para ambos os testes, foi gerado um *dataset* de 10000 pontos 3D distribuídos aleatoriamente no mundo. É assumido que o sistema se encontra correctamente calibrado, onde as câmaras rigidamente acopladas são orientadas segundo as transformações representadas nas matrizes da equação 3.2. Iniciando-se o movimento do sistema segundo as trajectórias definidas, a posição relativa é estimada *frame* a *frame*, aplicando-se diretamente os algoritmos descritos na secção 4. O sistema de câmaras usado nas simulações é ilustrado na Fig. 3.2.

A determinação dos factores de escala é condicionada pelo nível de ruído. O ruído Gaussiano é

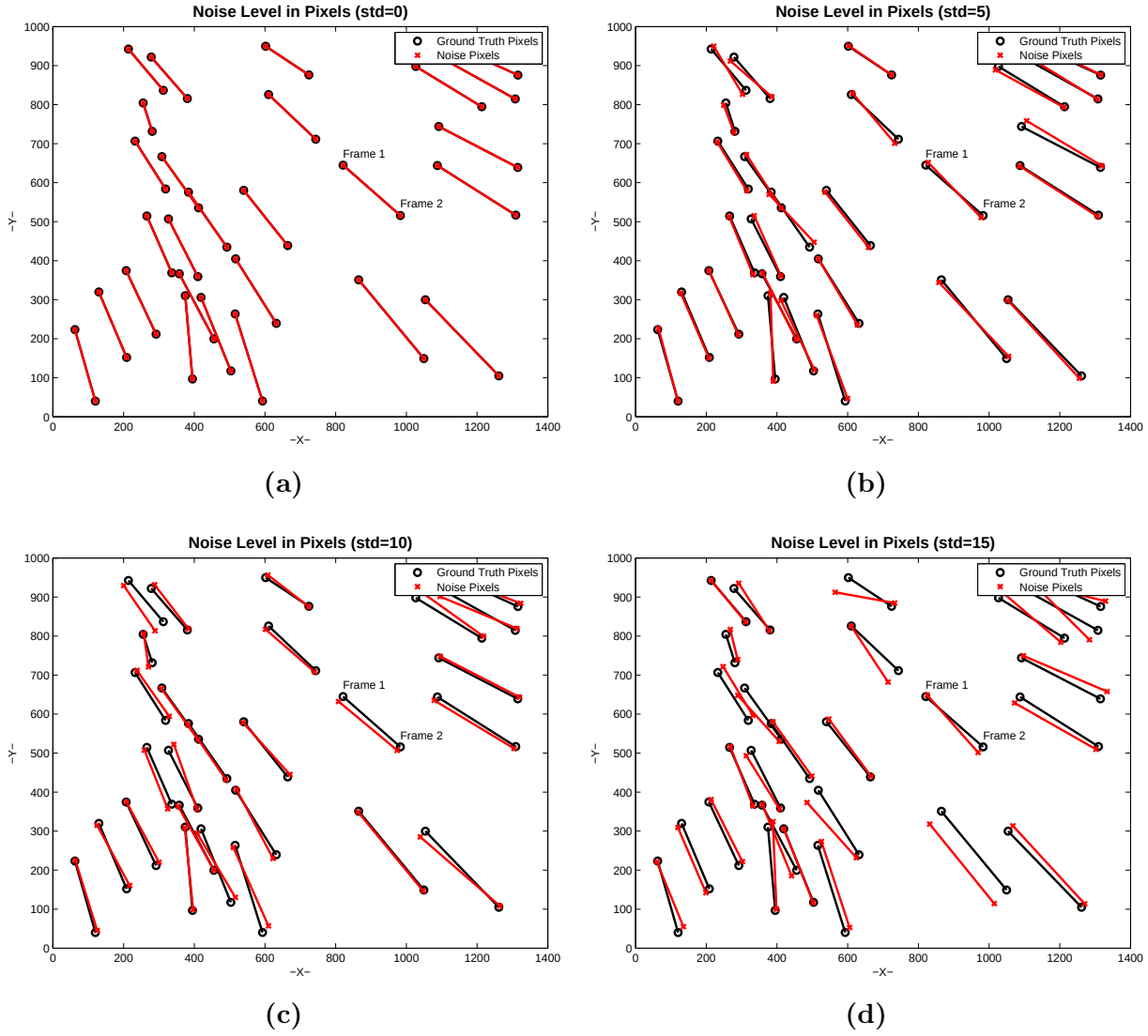


Figura 5.1: Nas Fig. (a), (b), (c) e (d) estão representados níveis de ruído Gaussianos com um desvio padrão diferente, *std(0)*, *std(5)*, *std(10)* e *std(15)*, aplicados aos *pixels* originais entre duas *frames*. Os níveis de ruído escolhidos demonstram o desvio necessário para o teste dos algoritmos.

um ruído estatístico cuja função densidade de probabilidade $\mathcal{P}$, é igual ao da distribuição normal:

$$\mathcal{P}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(z-\mu)^2}{2\sigma^2}}, \tag{5.1}$$

sendo μ a média e σ o desvio padrão. Em imagens reais, as principais fontes de ruído Gaussiano devem-se a problemas de iluminação durante a aquisição da imagem. Neste caso, visto que no simulador usamos directamente os pontos 3D, o ruído branco Gaussiano é aplicado na projecção dos pontos 3D associados a cada câmara. Na Fig. 5.1 é ilustrado o efeito dos vários níveis de ruído nesses *pixels*.

Os resultados obtidos na análise dos algoritmos foram bastante satisfatórios e permitiram

observar detalhadamente o impacto do ruído na estimação do factor de escala. Para a trajectória 2D, representada na Fig. 5.9, e para a trajectória 3D, apresentada na Fig. 5.10, foram escolhidos movimentos entre duas *frames* aos quais se aplicaram diferentes níveis de ruído, realizando-se para cada nível 200 experiências.

A *performance* dos algoritmos está dependente do nível de ruído injectado em cada experiência. Neste caso, e uma vez que os algoritmos implementados têm o objectivo de recuperar a escala correcta da translação efectuada pelo sistema de câmaras entre duas *frames*, foi feita uma análise ao módulo e fase do vector de translação. O erro no módulo do vector de translação é calculado entre o vector de translação de *Ground Truth*, $\mathbf{t}_{GT}$ que é conhecido, e o vector de translação estimado pelos algoritmos, $\mathbf{t}_{EST}$. Aplicando a fórmula $|\mathbf{t}_{EST}-\mathbf{t}_{GT}|/|\mathbf{t}_{GT}|$, resulta o erro relativo no módulo associado à translação efectuada entre as duas *frames* em análise. Repete-se a experiência 200 vezes para o mesmo nível de ruído e obtemos uma média e desvio padrão para essa intensidade. O erro relativo do módulo é representado nas Figs. 5.2(a), 5.2(c), 5.3(a) e 5.3(c). Sendo $\mathbf{t}_{GT} \in \mathbb{R}^{3\times1}$ e $\mathbf{t}_{EST} \in \mathbb{R}^{3\times1}$, os dois vectores translação, podemos fazer uma análise detalhada do erro associado a cada coordenada, o que tem bastante interesse pois permite-nos tirar conclusões se um movimento num determinado sentido afecta mais o erro e porque razão isso acontece. Nas figuras mencionadas anteriormente, é visível que quando o ruído é nulo o erro no módulo é zero, o que significa que os vectores de translação $\mathbf{t}_{GT}$ e $\mathbf{t}_{EST}$ são iguais. Na Fig. 5.5, é apresentado o mesmo tipo de erro relativo, mas para os vectores devidamente normalizados, o que dá uma informação ao nível global do erro. Por outro lado, o erro associado à fase permite-nos avaliar a direcção do vector de translação, representado nas Figs. 5.2(b), 5.2(d), 5.3(b) e 5.3(d). Assim, o erro na fase ao fim de 200 amostras é calculado pelo ângulo entre o vector, $\mathbf{t}_{GT}$ e os vectores estimados pelo algoritmo, $\mathbf{t}_{EST}$, que mais uma vez é zero na ausência de ruído.

Nos resultados seguintes são apresentados também os erros relativos à rotação, apesar da sua estimação não depender dos dois métodos implementados, mas sim, dos métodos usados para a estimação da matriz fundamental e essencial, que não foram objecto de estudo nos resultados experimentais. A estimação da matriz fundamental pode ser feita através dos métodos descritos na secção 2. Para as simulações usadas de seguida foi escolhido o método *RANSAC*, por ser rápido em termos de computação e por ter sido o método usado pelos autores dos artigos. O método *RANSAC*, apresenta um tempo médio de computação de 0.0035 segundos, enquanto que o algoritmo dos 8 pontos apresenta um tempo médio de 0.0032 segundos e o método *Least Median of Squares* demora em média 0.4081 segundos. Para a estimação da matriz fundamental é usado

o máximo número de correspondências de pontos disponíveis entre as duas *frames*. Os resultados da matriz fundamental em termos numéricos são exactamente iguais aplicando qualquer um dos três algoritmos de estimação. Assim, como é necessário uma estimação da matriz fundamental e essencial *frame* a *frame*, é importante que o algoritmo seja rápido computacionalmente para poder ser aplicado em tempo real. A recuperação da rotação como é visível nos gráficos da Fig. 5.4, é bastante robusta, o que está de acordo e é sabido, que através da matriz de essencial existe uma óptima recuperação da matriz de rotação. Para calcular o erro na rotação, ilustrado na Fig. 5.4, sabendo $\mathbf{R}_{GT} \in \mathbb{R}^{3\times3}$, rotação de *Ground Truth*, e $\mathbf{R}_{EST} \in \mathbb{R}^{3\times3}$ rotação estimada, retirando os ângulos de *Euler* associados a cada uma das matrizes $(\theta_{GT_x}, \theta_{GT_y}, \theta_{GT_z})$ e $(\theta_{EST_x}, \theta_{EST_y}, \theta_{EST_z})$, podemos aplicar a fórmula $|\epsilon| = \sqrt{(\theta_{EST_x} - \theta_{GT_x})^2 + (\theta_{EST_y} - \theta_{GT_y})^2 + (\theta_{EST_z} - \theta_{GT_z})^2}$, onde ϵ corresponde ao erro de rotação em graus. Quanto ao próprio factor de escala não é efectuada nenhuma análise em particular, pois o erro no seu cálculo influencia directamente o módulo do vector de translação estimado, $\mathbf{t}_{EST}$ estando directamente implícito nesse vector.

O algoritmo descrito na secção 4.1, revelou uma grande sensibilidade ao ruído, tanto na trajectória 2D como na 3D, visto que para estimar o factor de escala é apenas usada uma correspondência da câmara B. Aplicando o algoritmo entre duas *frames* e realizando 200 amostras para cada nível de ruído, podemos então concluir que uma pequena variação nesse ponto provoca grandes desvios na estimação do factor de escala correcto e, consequentemente, na estimação da posição relativa das câmaras. De acordo com a análise apresentada nas Figs. 5.2 e 5.3, podemos verificar que o erro relativo no módulo do vector de translação cresce rapidamente à medida que se aumenta o ruído. Em relação à direcção do vector de translação, apesar de haver um aumento acompanhado do aumento do nível de ruído, mantém-se sempre por valores muito baixos.

Quanto ao algoritmo descrito na secção 4.2, foi comprovado que é bastante robusto para níveis normais de ruído. Os resultados observados na Fig. 5.2 e Fig. 5.3 comprovam isso mesmo, o que seria de esperar, e está de acordo com os resultados demonstrados pelos autores do artigo. De notar que, para níveis muito superiores de ruído aplicados ao algoritmo anterior, este mantém uma boa estabilidade e permite determinar a posição relativa com bastante exactidão. O erro na direcção do vector de translação é igualmente muito baixo nos dois algoritmos (a sua estimação não depende dos métodos). Verifica-se ainda que o erro no módulo cresce para valores de ruído muito altos, contrariamente ao que acontecia com o algoritmo anterior.

Para uma análise mais detalhada e complementar aos resultados anteriormente expostos, são apresentadas as Figs. 5.6 e 5.7, onde é apresentado o erro absoluto do módulo relativamente ao

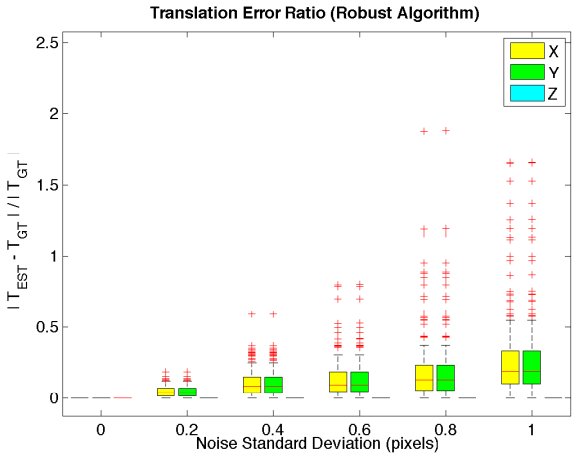
vector de translação. Para a Fig. 5.6, o erro é calculado através da expressão $|\mathbf{t}_{GT} - \mathbf{t}_{EST}|$, e apresentado detalhadamente para as três coordenadas. Na Fig. 5.7, o erro é calculado para a globalidade do vector de translação. Por fim, para a Fig. 5.8, é apresentado o rácio de erro na escala dado pela fórmula $|\mathbf{t}_{EST}|/|\mathbf{t}_{GT}|$. Uma das conclusões que estes gráficos permitem tirar, é que o erro na coordenada Z, para os movimentos 3D, aumenta muito mais rapidamente que em X e Y. Isto acontece quando existe uma variação brusca da translação na direcção de uma das coordenadas. Foram efectuados testes para outros tipos de trajectórias e os resultados foram semelhantes para variações bruscas em diferentes orientações. Como já foi várias vezes referido, todos estes erros foram calculados tendo por base a mesma trajectória 2D e 3D entre duas *frames*, sendo que para cada nível de ruído foram realizadas 200 experiências.

Nas Figs. 5.9 e 5.10, podemos visualizar as trajectórias 2D e 3D projectadas, a posição da câmara segundo *Ground Truth*, e as posições relativas estimadas através dos algoritmos. É possível observar que os resultados são exactamente sobrepostos à trajectória *Ground Truth* na ausência de ruído, o que seria de esperar.

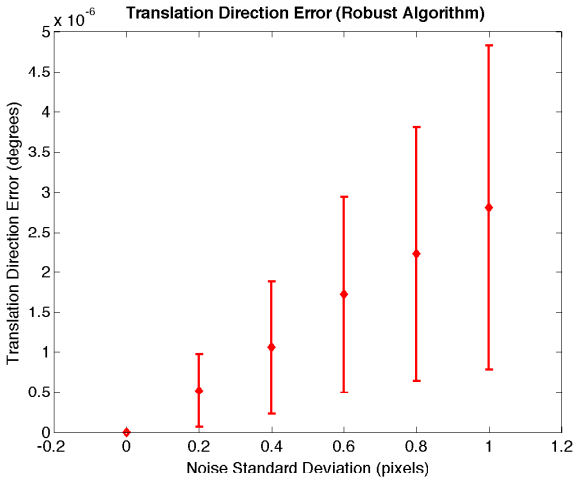
Nas Figs. 5.11 e 5.12 são apresentados os resultados após a inserção de ruído e o seu impacto na posição relativa estimada. É observável pelos dados anteriores que um pequeno ruído (desvio padrão 0.1 *pixels*) provoca grandes desvios na estimação da posição segundo o algoritmo da secção 4.1, enquanto que para o algoritmo 4.2 é necessário um nível de ruído muito superior (desvio padrão 15 *pixels*) para haver grandes discrepâncias na estimação da posição. Um ruído com desvio padrão inferior a 0.5 *pixels* é quase insignificante, como pode ser observado na Fig. 5.1 e não deveria provocar um desvio tão acentuado na estimação da posição. Por outro lado, um desvio padrão superior a 15 *pixels* está a um nível suficiente elevado para ilustrar o efeito do ruído em situações de aquisição de imagens reais.

As trajectórias 2D e 3D representadas nas Figs. 5.11 e 5.12, ilustram o desvio da estimação da posição face à trajectória *Ground Truth*. Estas trajectórias, como já foi referido anteriormente, são estimadas *frame* a *frame*, isto é, o movimento relativo é estimado entre duas *frames* consecutivas, por exemplo, entre a *frame* 10 e *frame* 11. O ruído aplicado é constante ao longo de todas as frames de movimento, ou seja, desvio padrão 0.1 *pixels* utilizando o algoritmo da secção 4.1, e desvio padrão 15 *pixels* para o algoritmo da secção 4.2. Somando iterativamente todas as estimações relativas, obtemos o gráfico da posição do sistema de câmaras ao longo tempo.

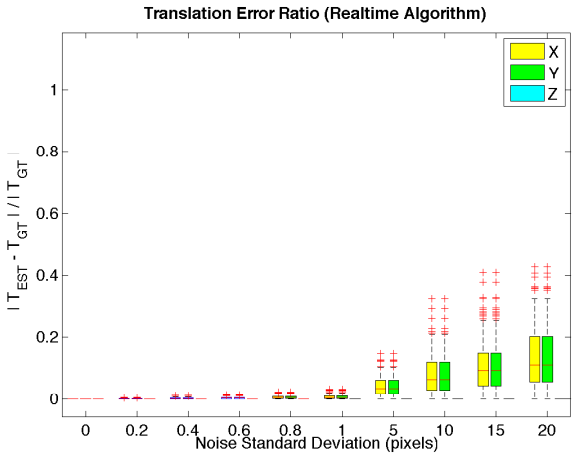
Na Fig. 5.13 está representada uma ampliação dos mini-referenciais associados a cada algoritmo, desenhados nos gráficos de movimento.



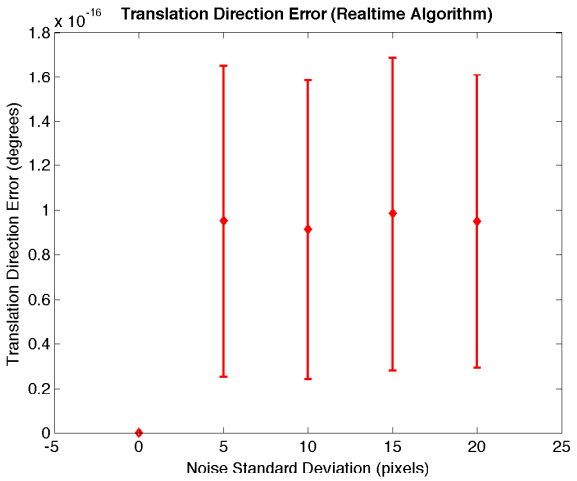
(a)



(b)



(c)



(d)

Figura 5.2: Na Fig. (a), está ilustrado o erro relativo em módulo obtido entre o vector de translação *Ground Truth*, $\mathbf{t}_{GT}$, e o vector de translação estimado, $\mathbf{t}_{EST}$, para diferentes níveis de ruído utilizando o algoritmo *Robust 6DOF Motion Estimation for Non-Overlapping, Multi-Camera Systems*, [1]. O erro na direcção do vector de translação é apresentado na Fig. (b). Analogamente, na Fig. (c), está ilustrado o erro em módulo, para diferentes níveis de ruído utilizando o algoritmo *Real Time 6D Stereo Visual Odometry with Non-Overlapping Fields of View*, [8], e o erro na direcção representado na Fig. (d). Para cada grau de ruído são realizadas 200 experiências, o que permite uma análise detalhada do efeito dos vários níveis de ruído na média e desvio padrão do erro, associados ao módulo e fase do vector de translação. Esta análise foi construída com base numa trajectória 2D entre duas *frames*, considerando duas câmaras e existindo apenas um movimento genérico em X e Y, como se pode verificar nos gráficos.

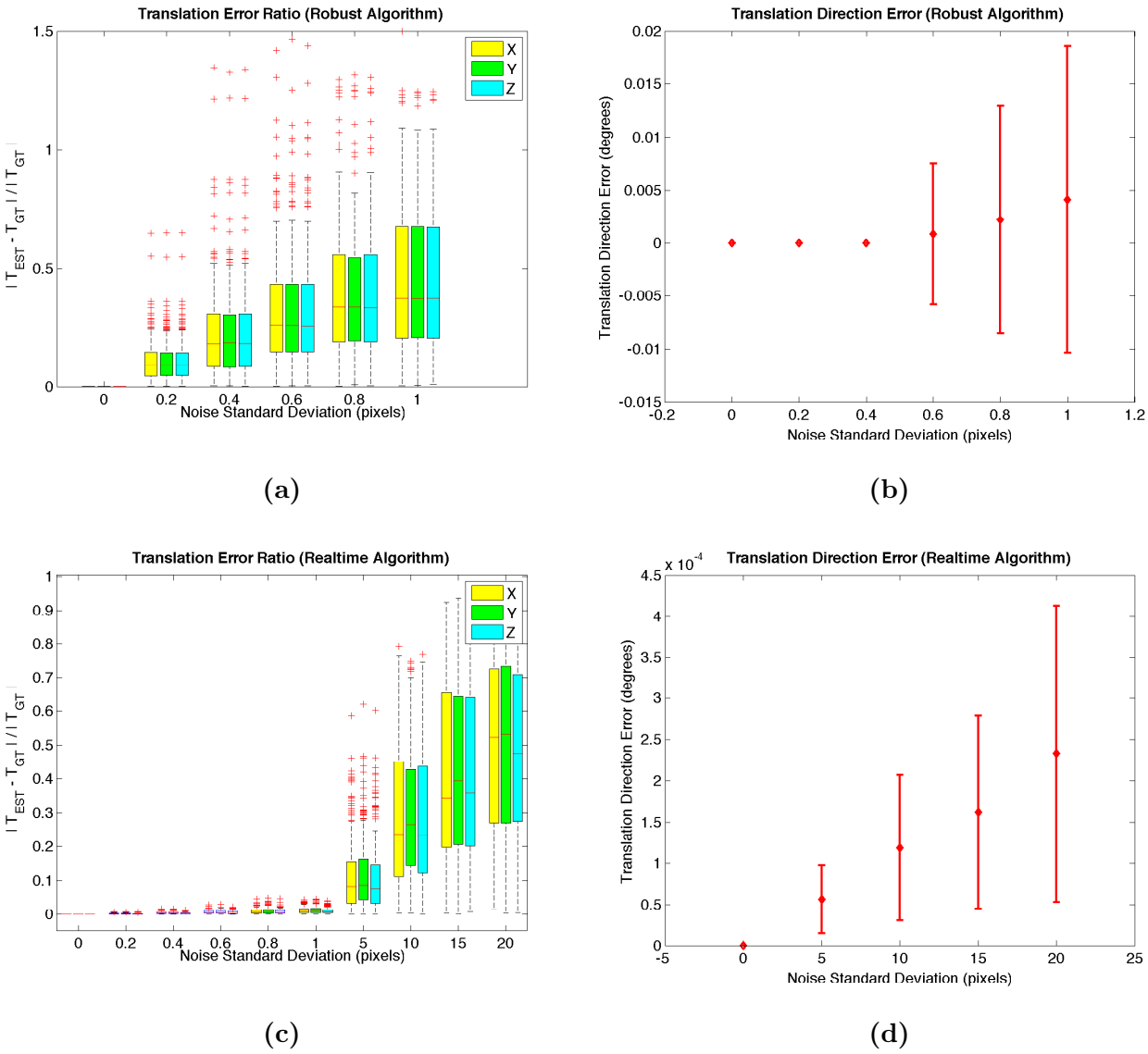


Figura 5.3: Na Fig. (a), está ilustrado o erro relativo em módulo obtido entre o vector de translação *Ground Truth*, $\mathbf{t}_{GT}$, e o vector de translação estimado, $\mathbf{t}_{EST}$, para diferentes níveis de ruído utilizando o algoritmo [1]. O erro na direcção do vector de translação é apresentado na Fig. (b). Analogamente, na Fig. (c), está ilustrado o erro em módulo, para diferentes níveis de ruído utilizando o algoritmo [8], e o erro na direcção representado na Fig. (d). Para cada grau de ruído são realizadas 200 experiências, o que permite uma análise detalhada do efeito dos vários níveis de ruído na média e desvio padrão do erro, associados ao módulo e fase do vector de translação. Esta análise foi construída com base numa trajectória 3D entre duas *frames*, considerando duas câmaras e existindo um movimento genérico em X, Y, e Z como é demonstrado no gráfico.

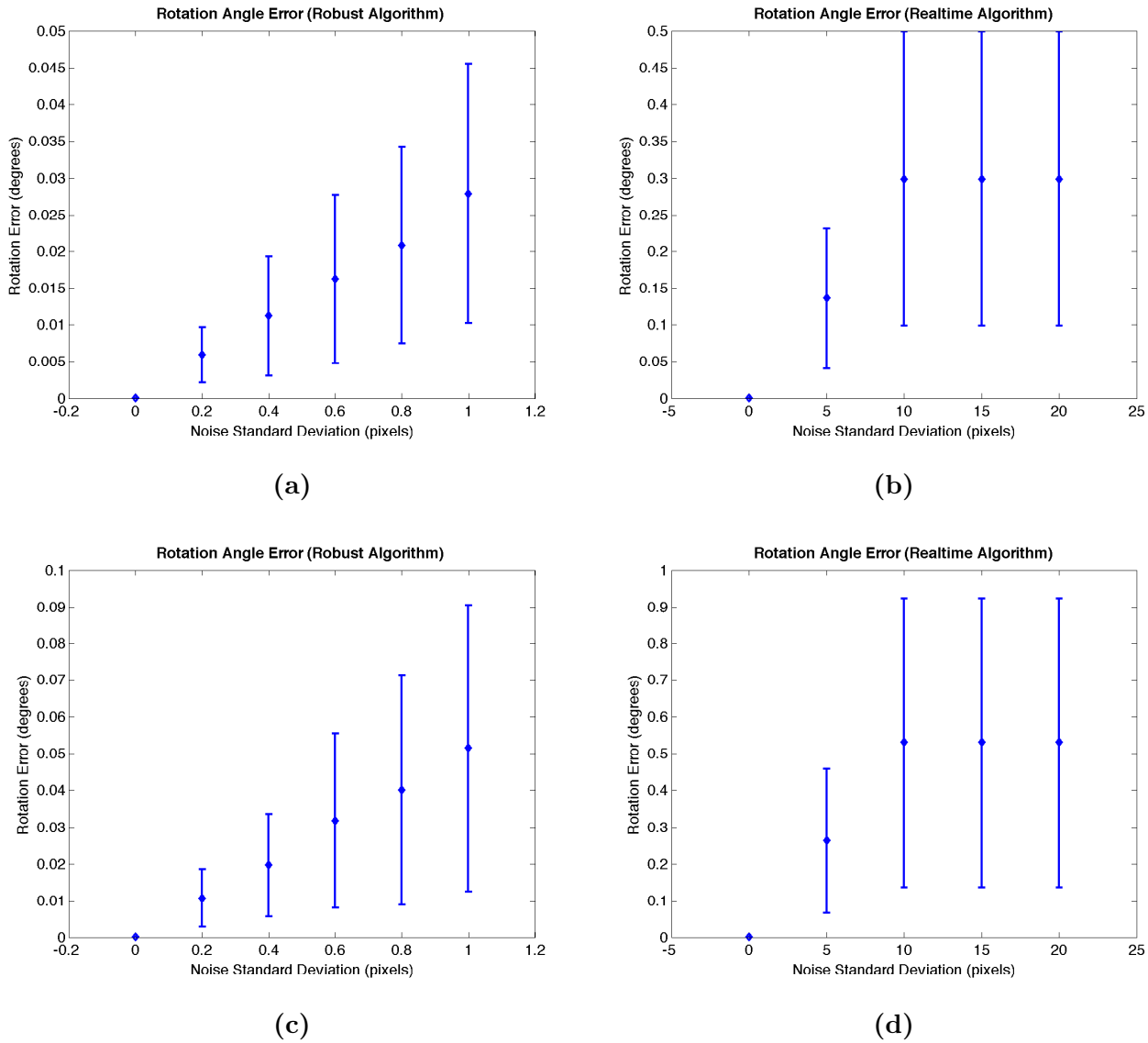


Figura 5.4: Na Fig. (a) e Fig. (b), está ilustrado o erro em graus associado à estimação da rotação face à rotação conhecida, *Ground Truth*, respectivamente aos algoritmos [1, 8], para o movimento 2D. Na Fig. (c) e Fig. (d), analogamente à descrição anterior, estão representados os erros na rotação para o movimento 3D. Para cada grau de ruído são realizadas 200 experiências, o que permite uma análise detalhada do efeito dos vários níveis de ruído na média e desvio padrão do erro, associados à rotação. Esta análise foi construída com base numa trajectória genérica em 2D e 3D entre duas *frames*.

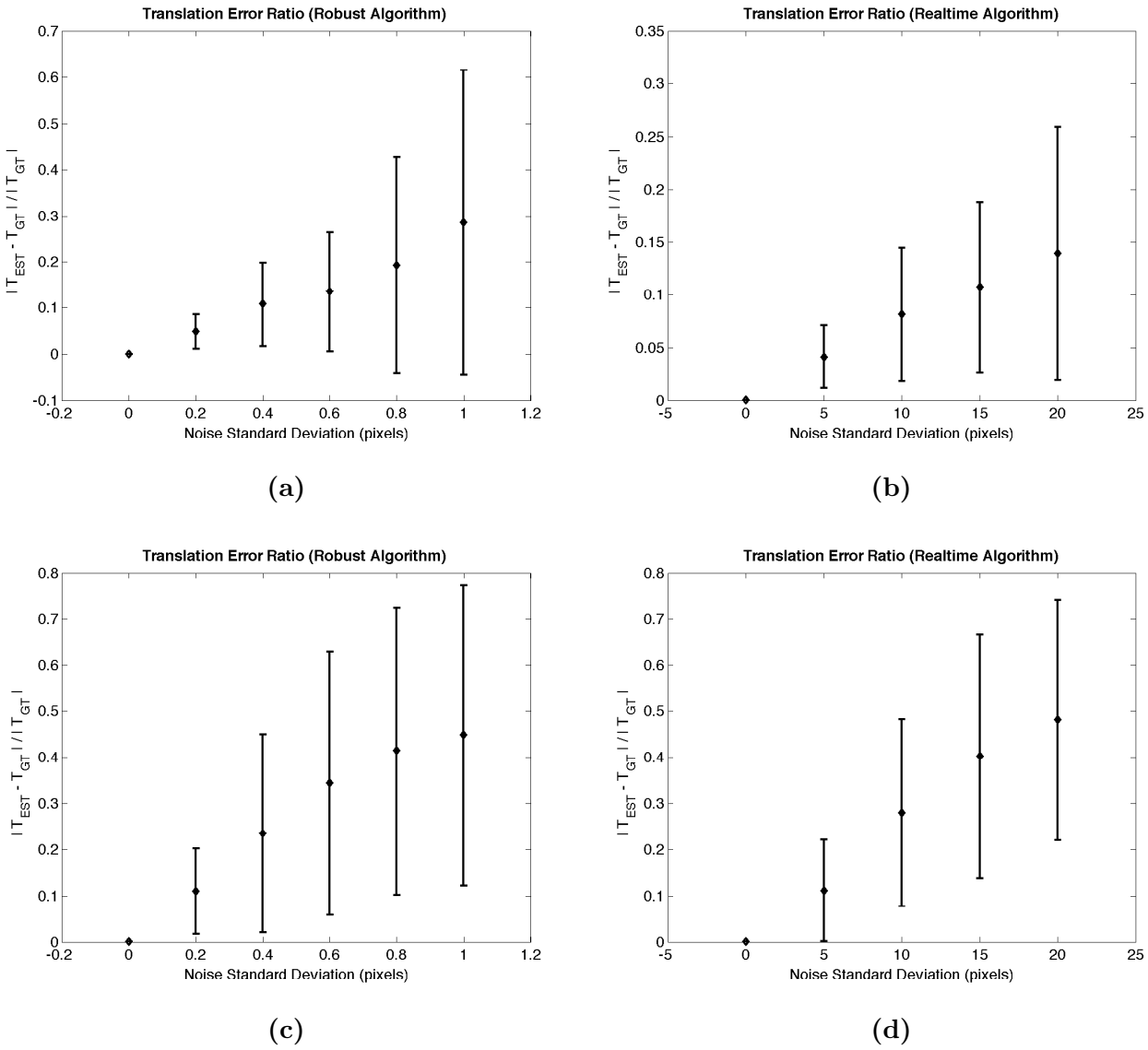


Figura 5.5: Na Fig. (a) e Fig. (b), está ilustrado o erro relativo em módulo obtido entre o vector de translação *Ground Truth*, $\mathbf{t}_{GT}$, e o vector de translação estimado, $\mathbf{t}_{EST}$, para diferentes níveis de ruído utilizando os algoritmos [1, 8], respectivamente, e associados ao movimento 2D. Analogamente na Fig. (c) e Fig. (d), está representado o erro para o movimento 3D. Para cada grau de ruído são realizadas 200 experiências, o que permite uma análise detalhada do efeito dos vários níveis de ruído na média e desvio padrão do erro, associados ao módulo do vector de translação. Esta análise foi construída com base numa trajectória genérica em 2D e 3D entre duas *frames*.

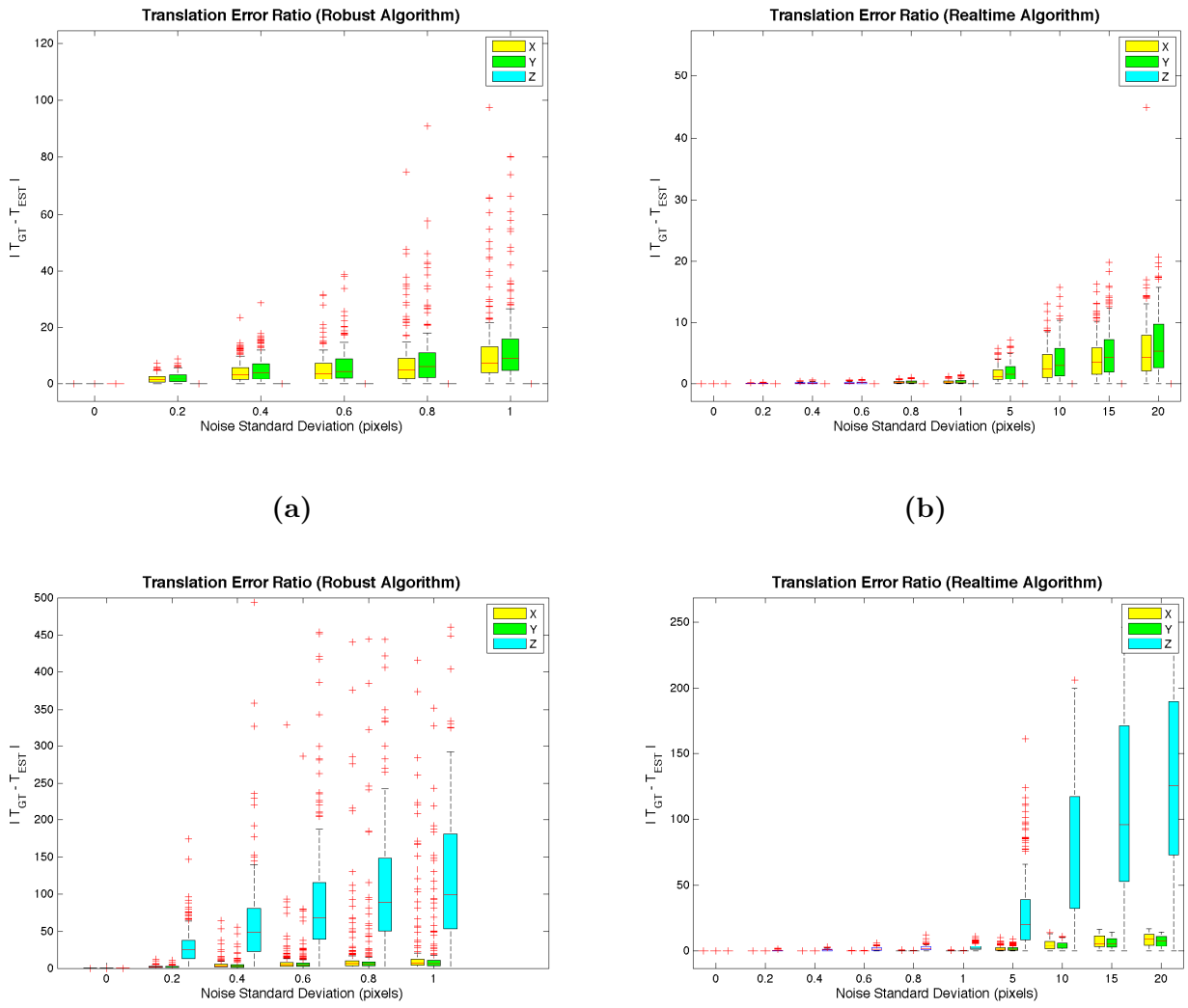


Figura 5.6: Na Fig. (a) e Fig. (b), está ilustrado o erro absoluto do módulo obtido entre o vector de translação *Ground Truth*, $\mathbf{t}_{GT}$, e o vector de translação estimado, $\mathbf{t}_{EST}$, para diferentes níveis de ruído utilizando os algoritmos [1, 8], respectivamente, e associados ao movimento 2D. Analogamente na Fig. (c) e Fig. (d), está representado o erro para o movimento 3D. Para cada grau de ruído são realizadas 200 experiências, o que permite uma análise detalhada do efeito dos vários níveis de ruído na média e desvio padrão do erro, associados ao módulo do vector de translação. Esta análise foi construída com base numa trajectória genérica em 2D e 3D entre duas *frames*.

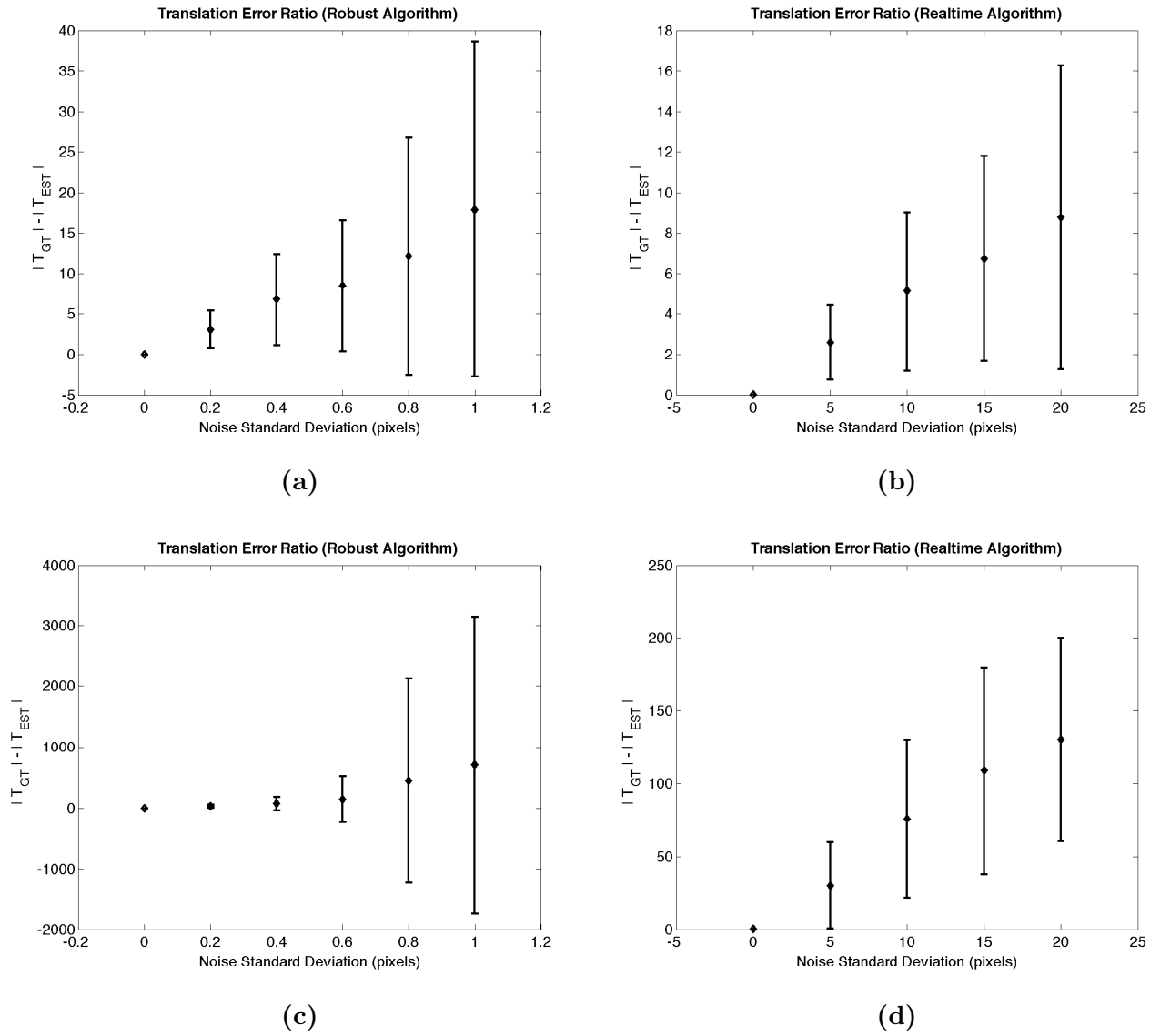


Figura 5.7: Na Fig. (a) e Fig. (b), está ilustrado o erro absoluto do módulo obtido entre o vector de translação *Ground Truth*, $\mathbf{t}_{GT}$, e o vector de translação estimado, $\mathbf{t}_{EST}$, para diferentes níveis de ruído utilizando os algoritmos [1, 8], respectivamente, e associados ao movimento 2D. Analogamente na Fig. (c) e Fig. (d), está representado o erro para o movimento 3D. Em contraste com a Fig. 5.6, o erro aqui é apresentado na globalidade. Para cada grau de ruído são realizadas 200 experiências, o que permite uma análise detalhada do efeito dos vários níveis de ruído na média e desvio padrão do erro, associados ao módulo do vector de translação. Esta análise foi construída com base numa trajectória genérica em 2D e 3D entre duas *frames*.

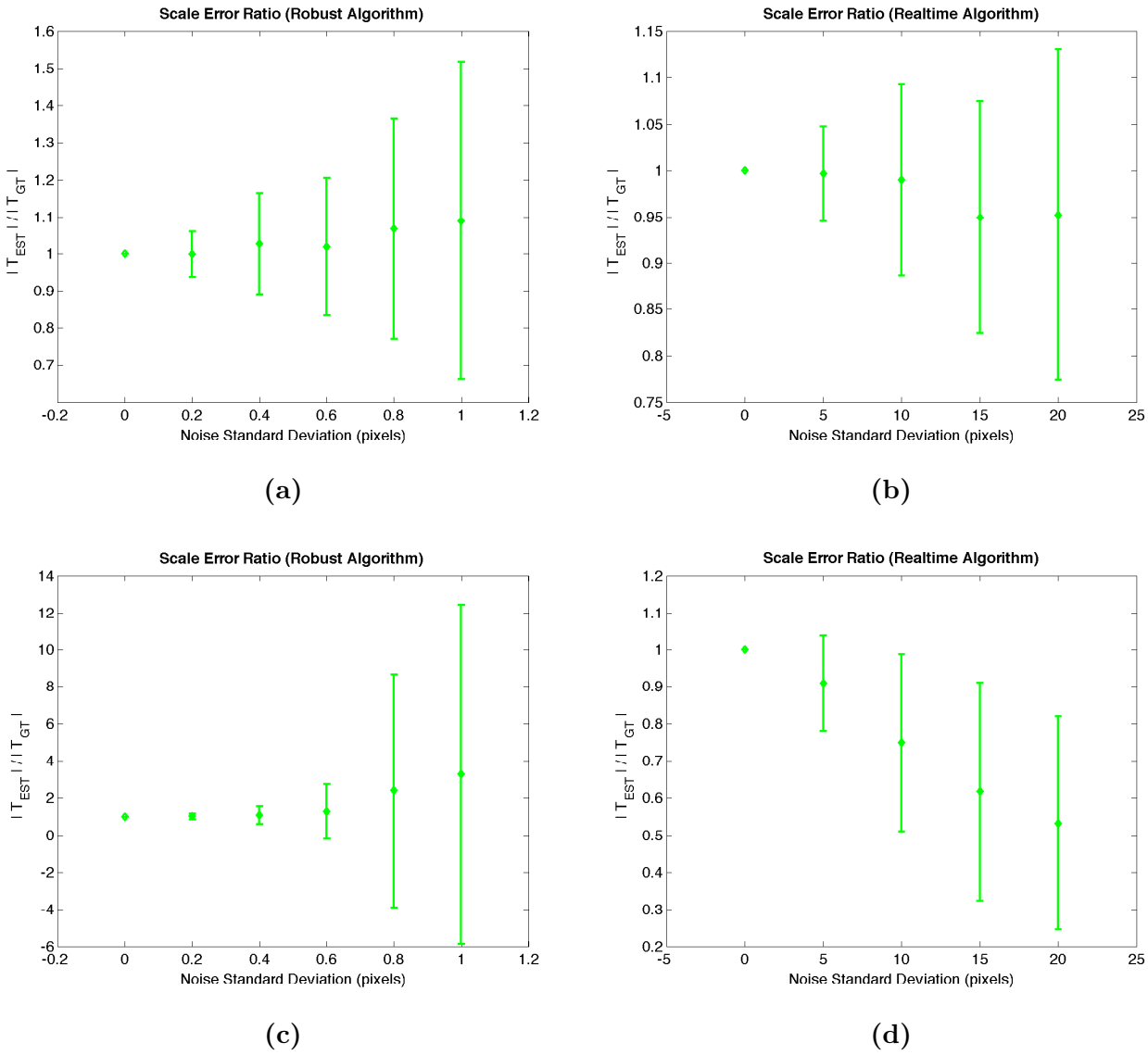
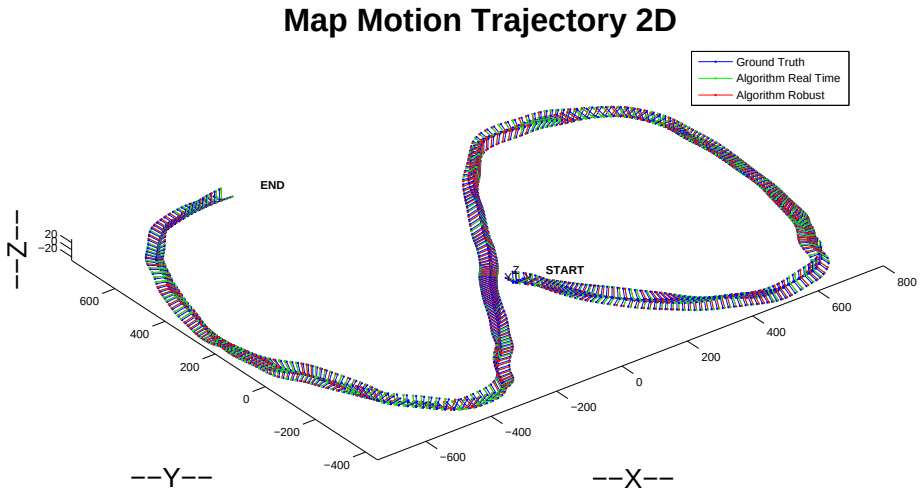
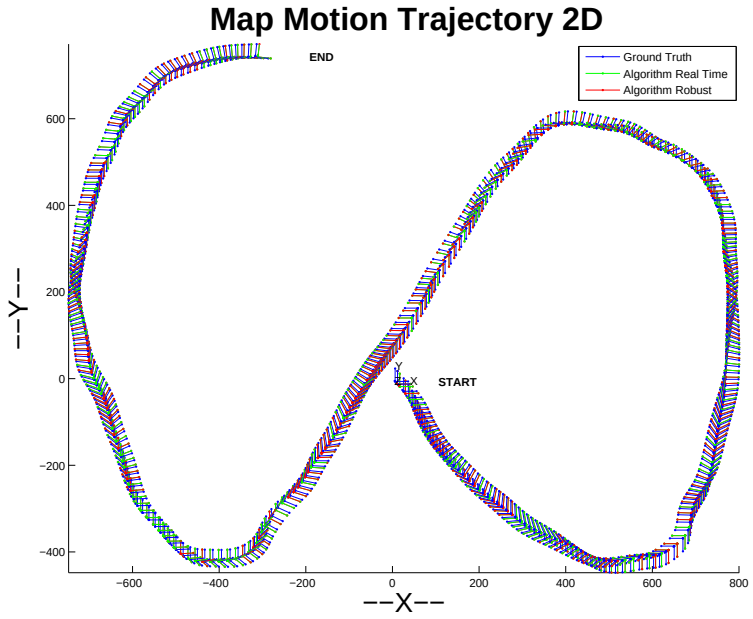


Figura 5.8: Na Fig. (a) e Fig. (b), está ilustrado o rácio entre o vector de translação *Ground Truth*, $\mathbf{t}_{GT}$, e o vector de translação estimado, $\mathbf{t}_{EST}$, para diferentes níveis de ruído utilizando os algoritmos [1, 8], respectivamente, e associados ao movimento 2D. Analogamente na Fig. (c) e Fig. (d), está representado o rácio para o movimento 3D. Esta análise foi desenvolvida com base numa trajectória genérica em 2D e 3D entre duas *frames*.

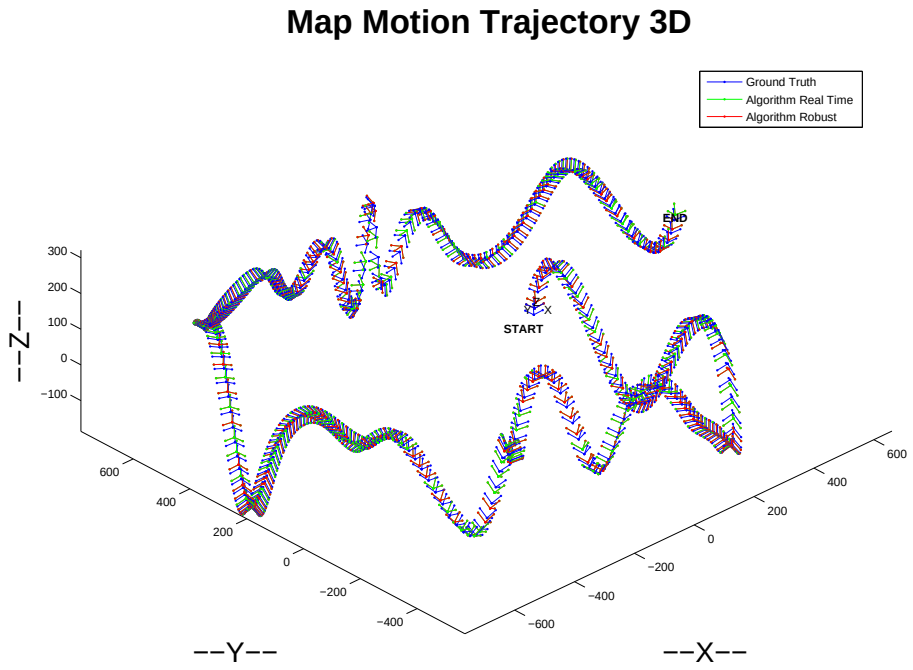


(a)

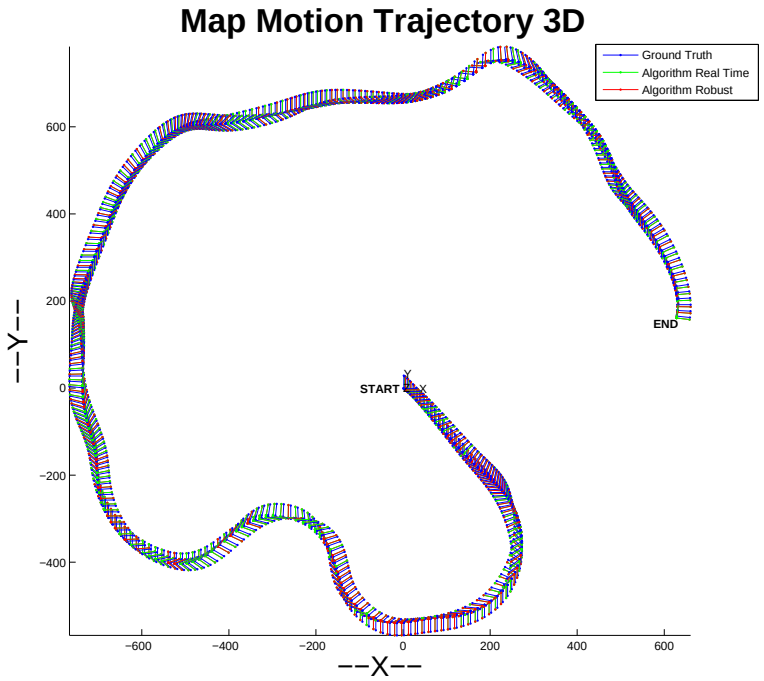


(b)

Figura 5.9: Na Fig. (a), é representado o trajecto 2D realizado pela câmara A. A cor azul é apresentada a trajectória *Ground Truth*, onde pode ser visto um mini-referencial da posição da câmara ao longo de todo o movimento. A cor verde está associada à posição relativa estimada, utilizando o algoritmo *Real Time 6D Stereo Visual Odometry with Non-Overlapping Fields of View*, [8], e cor vermelha à posição relativa ao longo da trajectória, estimada pelo algoritmo *Robust 6DOF Motion Estimation for Non-Overlapping, Multi-Camera Systems*, [1]. Como é possível visualizar, os movimentos são exactamente iguais ao *Ground Truth* devido à ausência de qualquer tipo de ruído. A Fig. (b), apresenta a mesma trajectória, mas vista de outra perspectiva.

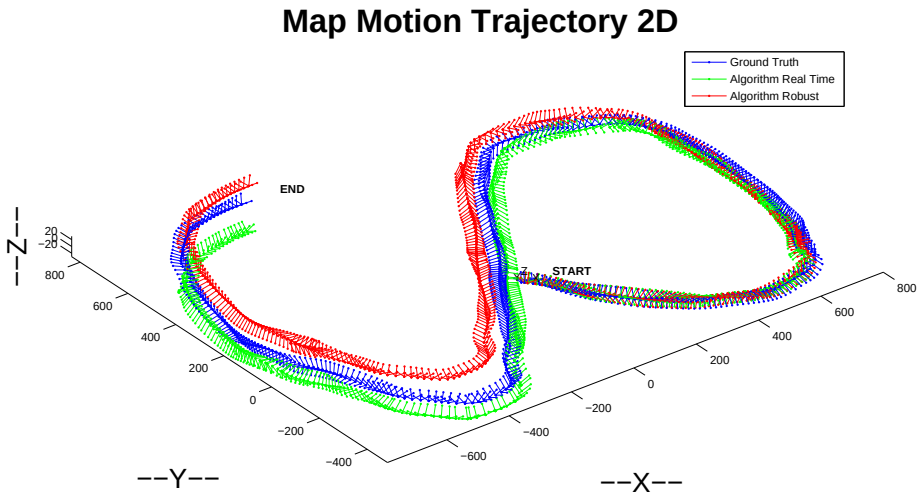


(a)

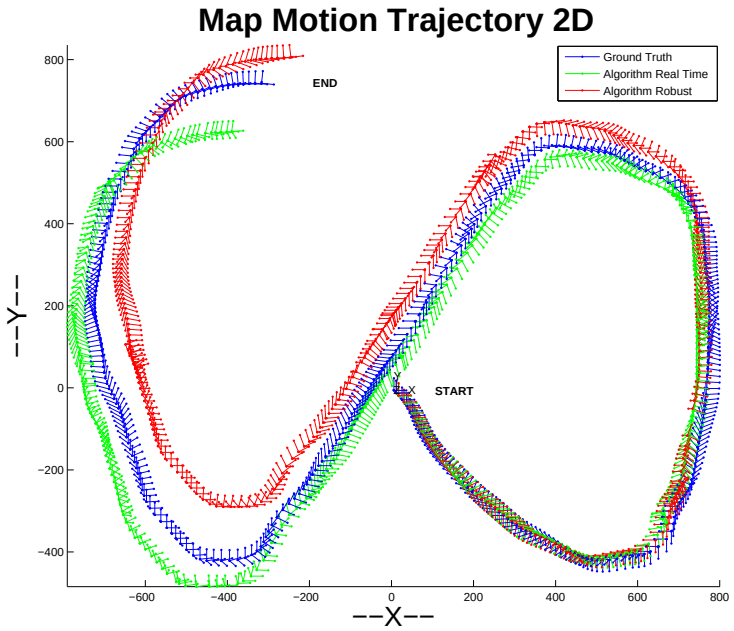


(b)

Figura 5.10: Na Fig. (a), está representado o trajecto 3D realizado pela câmara A, de igual modo ao apresentado para a trajectória 2D na Fig. 5.9. Os movimentos estimados são exactamente iguais ao *Ground Truth* devido à ausência de qualquer tipo de ruído. A Fig. (b), apresenta uma vista superior do trajecto 3D.

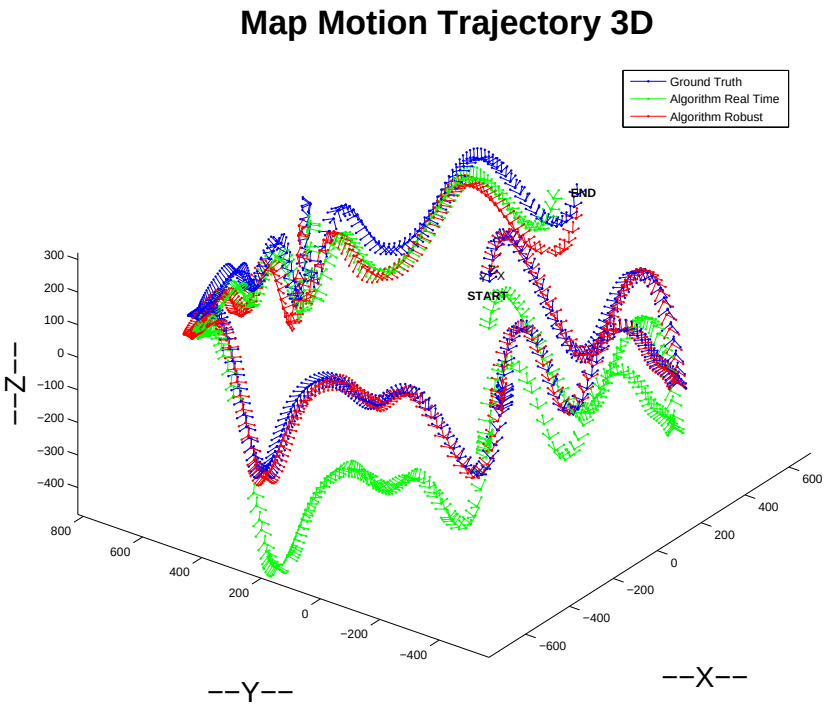


(a)

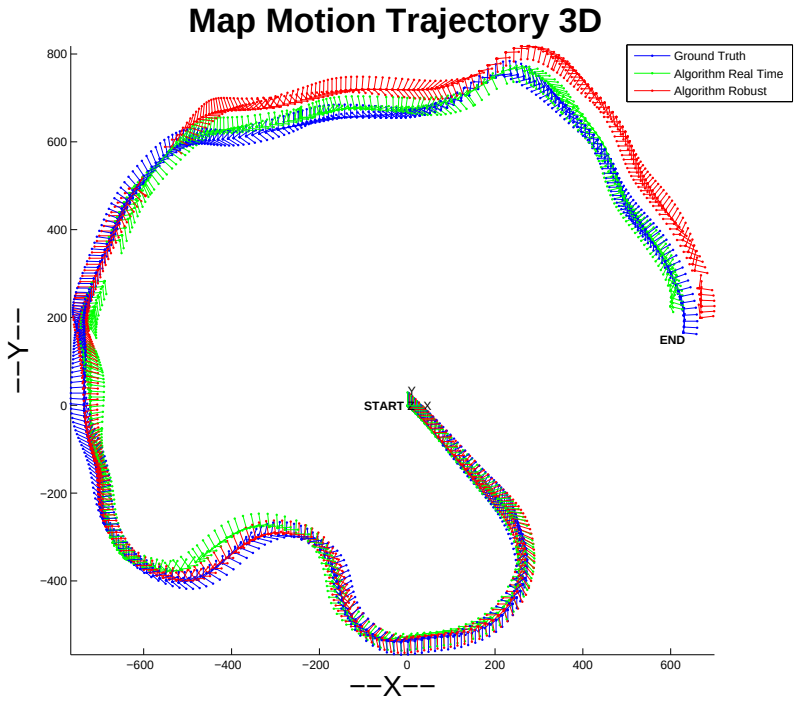


(b)

Figura 5.11: Na Fig. (a), é representado o trajecto 2D realizado pela câmara A. Analogamente ao que foi representado na Fig. 5.9, está ilustrado agora o efeito do ruído na estimação relativa da posição das câmaras, utilizando os dois algoritmos [1, 8]. Em comparação com a trajectória *Ground Truth*, há um desvio em relação às trajectórias estimadas como se pode comprovar visualmente. O ruído aplicado não foi o mesmo nos dois algoritmos, devido às diferenças de sensibilidade. Para o algoritmo [1], foi utilizado um ruído Gaussiano com desvio padrão de 0.1 *pixels*, e para o algoritmo [8], um ruído Gaussiano com desvio padrão de 15 *pixels*. Na Fig. (b), pode observar-se o efeito do ruído na estimação da posição relativa da câmara A, com vista superior.



(a)



(b)

Figura 5.12: Na Fig. (a), é mostrado o resultado da estimação relativa da trajectória 3D sujeita a ruído, face à trajectória *Ground Truth*. Como já tinha sido mencionado para a trajectória 2D na Fig. 5.11, foi aplicado o mesmo ruído, sendo que para o algoritmo [1], foi utilizado um ruído Gaussiano com desvio padrão de 0.1 *pixels*, e para o algoritmo [8], um ruído Gaussiano com desvio padrão de 15 *pixels*. O impacto dos níveis de ruído provocam diferenças visíveis na estimação da posição ao longo do movimento. Na Fig. (b) está ilustrada a perspectiva superior do trajecto 3D.

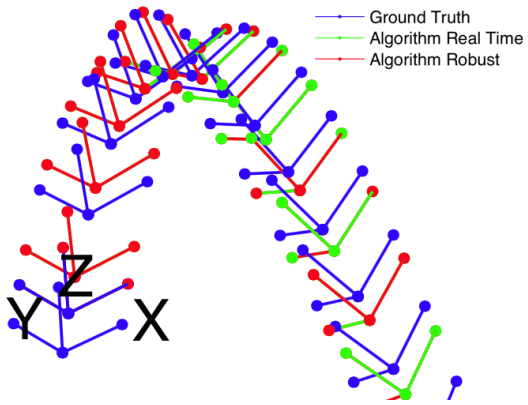


Figura 5.13: Representação ampliada dos mini-referenciais que são apresentados nas figuras anteriores. O referencial azul corresponde ao *Ground Truth*, o verde está associado ao algoritmo *Real Time 6D Stereo Visual Odometry with Non-Overlapping Fields of View*, [8], e o vermelho ao algoritmo *Robust 6DOF Motion Estimation for Non-Overlapping, Multi-Camera Systems*, [1].

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Conclusões

A estimação do movimento da câmara em seis graus de liberdade (rotação e translação), que definem a posição da câmara no mundo foi objecto de estudo nesta dissertação, para sistemas de múltiplas câmaras rigidamente acopladas e com campos visuais não sobrepostos. A estimação métrica do factor de escala foi feita com base na implementação dos algoritmos apresentados na secção 4. O simulador apresentado na secção 3 é um peça fundamental para a aplicação de algoritmos e estudo cientifico na área de visão, visto tratar-se de um ambiente controlado pelo utilizador, onde podem ser feitos vários testes de simulação para uma diversidade de situações.

A análise dos algoritmos apresentados na secção 4 permitiu concluir que a estimação métrica do factor de escala pode ser feita robustamente utilizando sistemas de múltiplas câmaras com campos visuais não sobrepostos. Este estudo, permitiu identificar a melhor solução a ser aplicada num sistema móvel com múltiplas câmaras e quais os limites impostos teoricamente, que se podem vir a verificar em situações de aplicação real.

O algoritmo descrito na secção 4.1 revelou-se bastante ineficaz para aplicação em sistemas reais visto ser bastante sensível ao ruído, facto que está inerente a qualquer tipo de experiência. Por outro lado, o algoritmo apresentado na secção 4.2, demonstrou ser uma boa solução para estimar a escala métrica do movimento, e a sua aplicação em sistemas reais será um método de precisão adicional face a outros sistemas de estimação já existentes. Este algoritmo também surgiu como melhoria do algoritmo anterior, o que comprova que houve uma evolução no sentido de descobrir um método robusto para aplicação em situações reais.

Os objectivos inicialmente propostos para a realização da dissertação foram cumpridos, e os resultados da análise efectuada na secção 5, demonstram as conclusões principais expostas anteriormente e dão um contributo adicional na validação dos métodos.

6.2 Trabalho Futuro

Os trabalhos futuros nesta área podem ser amplamente alargados, pois há sempre novas situações por investigar e a evolução tecnológica no campo da visão por computador está em actual crescimento.

Existem vários problemas por resolver e poderão ser feitos mais estudos, sobre sistemas de múltiplas câmaras, como por exemplo, o uso de três ou mais câmaras para a estimação de movimento.

O simulador desenvolvido na secção 3, poderia ser adaptado para o uso de imagens reais ou até mesmo ser implementado num sistema que use *Graphic Processor Unit (GPU)*, o que diminuiria os tempos de computação e permitiria trabalhar com *datasets* reais. A validação dos algoritmos recorrendo a esses *datasets*, iria proporcionar a aplicação dos métodos em tempo real a sistemas de robótica móvel e aprofundar o estudo de eventuais problemas que daí pudessem advir.

Bibliografia

[1] B. Clipp, J.-H. Kim, J.-M. Frahm, M. Pollefeys, and R. Hartley. Robust 6DOF Motion Estimation for Non-Overlapping, Multi-Camera Systems. In *Proceedings of the 2008 IEEE Workshop on Applications of Computer Vision*, pages 1–8, 2008.

[2] Brian Clipp, Christopher Zach, J-M Frahm, and Marc Pollefeys. A New Minimal Solution to the Relative Pose of a Calibrated Stereo Camera with Small Field of View Overlap. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1725–1732. IEEE, 2009.

[3] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images: the laws that govern the formation of multiple images of a scene and some of their applications*. MIT press, 2004.

[4] Martin A Fischler and Robert C Bolles. Random Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[5] Michael D Grossberg and Shree K Nayar. A General Imaging Model and a Method for Finding its Parameters. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 108–115. IEEE, 2001.

[6] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[7] Richard I Hartley. In Defense of the Eight-Point Algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580–593, 1997.

[8] T. Kazik, L. Kneip, J. Nikolic, M. Pollefeys, and R. Siegwart. Real-Time 6D Stereo Visual Odometry with Non-Overlapping Fields of View. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1529–1536, 2012.

[9] Jae-Hak Kim, Hongdong Li, and Richard Hartley. Motion Estimation for Nonoverlapping Multicamera Rigs: Linear algebraic and l∞ geometric solutions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1044–1059, 2010.

[10] Kurt Konolige, Motilal Agrawal, and Joan Sola. Large-Scale Visual Odometry for Rough Terrain. In *Robotics Research*, pages 201–212. Springer, 2011.

[11] Simon Lacroix, Anthony Mallet, Raja Chatila, and Laurent Gallo. Rover Self Localization in Planetary-Like Environments. In *Artificial Intelligence, Robotics and Automation in Space*, volume 440, page 433, 1999.

[12] Gim Hee Lee, Marc Pollefeys, and Friedrich Fraundorfer. Relative Pose Estimation for a Multi-Camera System with Known Vertical Direction. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 540–547. IEEE, 2014.

[13] Hongdong Li, Richard Hartley, and Jae-hak Kim. A Linear Approach to Motion Estimation Using Generalized Camera Models. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[14] H Christopher Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, MA Fischler and O. Firschein, eds*, pages 61–62, 1987.

[15] Hans P. Moravec. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. Technical report, DTIC Document, 1980.

[16] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Generic and Real-Time Structure from Motion Using Local Bundle Adjustment. *Image and Vision Computing*, 27(8):1178–1193, 2009.

[17] David Nister. An Efficient Solution to the Five-Point Relative Pose Problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):756–770, 2004.

[18] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652, 2004.

[19] Gabriel Nützi, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Fusion of imu and Vision for Absolute Scale Estimation in Monocular slam. *Journal of intelligent & robotic systems*, 61(1-4):287–299, 2011.

[20] Clark F Olson, Larry H Matthies, Marcel Schoppers, and Mark W Maimone. Robust Stereo Ego-Motion for Long Distance Navigation. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 453–458, 2000.

[21] Roger Y Tsai and Thomas S Huang. Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):13–27, 1984.

[22] Shinji Umeyama. Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):376–380, 1991.

[23] Gang Xu and Zhengyou Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*, volume 6. Springer Science & Business Media, 1996.