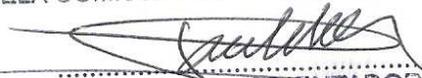


ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA
TESE DEFENDIDA POR Carlos Andrés Díaz
Espinosa E APROVADA
PELA COMISSÃO JULGADORA EM 28/07/2010


.....
ORIENTADOR

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

Autor: Carlos Andrés Díaz Espinosa

**Uma Aplicação de Navegação Robótica
Autônoma Através de Visão Computacional
Estéreo**

Campinas, 2010.

Autor: Carlos Andrés Diaz Espinosa

Uma Aplicação de Navegação Robótica Autônoma Através de Visão Computacional Estéreo

Dissertação apresentada ao Curso de Mestrado da Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas, como requisito para a obtenção do título de Mestre em Engenharia Mecânica.

Área de Concentração: Mecânica dos sólidos e projeto mecânico

Orientador: Prof. Dr. Paulo Roberto Gardel Kurka

Campinas, 2010
S.P. – Brasil

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

D543a Diaz Espinosa, Carlos Andrés
Uma aplicação de navegação robótica autônoma
através de visão computacional estéreo / Carlos Andrés
Diaz Espinosa. --Campinas, SP: [s.n.], 2010.

Orientador: Paulo Roberto Gardel Kurka.
Dissertação de Mestrado - Universidade Estadual de
Campinas, Faculdade de Engenharia Mecânica.

1. Navegação de robôs móveis - Localização. 2.
Visão de robô. 3. Processamento de imagens. 4.
Mapeamento (Matemática). 5. Simulação por
computador. I. Kurka, Paulo Roberto Gardel. II.
Universidade Estadual de Campinas. Faculdade de
Engenharia Mecânica. III. Título.

Título em Inglês: Autonomous application of robotic navigation using computer
stereo vision

Palavras-chave em Inglês: Mobile robot navigation - Location, Robot vision,
Image processing, Mapping (Mathematics), Computer
simulation

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Mestre em Engenharia Mecânica

Banca examinadora: Eleri Cardozo, Ely Carneiro de Paiva

Data da defesa: 28/07/2010

Programa de Pós Graduação: Engenharia Mecânica

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE POS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO

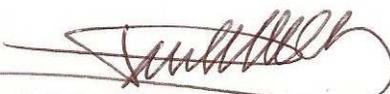
DISSERTAÇÃO DE MESTRADO ACADÊMICO

Uma Aplicação de Navegação Robótica Autônoma através de Visão Computacional Estéreo

Autor: Carlos Andrés Diaz Espinosa

Orientador: Prof. Dr. Paulo Roberto Gardel Kurka

A banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:



Prof. Dr Paulo Roberto Gardel Kurka, Presidente
DPM/FEM/UNICAMP



Prof. Dr Eleri Cardozo
DCA/FEEC/UNICAMP



Prof. Dr Ely Carneiro de Paiva
DPM/FEM/UNICAMP

Campinas, 28 de julho de 2010

Dedicatória

Quero dedicar este trabalho primeiro a Deus, por me dar a força necessária a cada dia, disciplina, conhecimento e especialmente por haver colocado pessoas de grande apoio e companhia durante este período da minha vida.

A meus pais Ruth Espinosa, Elvis Diaz e irmãos, porque apesar de não estarem presentes, sempre me incentivam, apoiaram e deram muito bem-estar emocional nos momentos mais difíceis, desde meu país Colômbia.

Agradecimentos

A realização deste trabalho exigiu todo meu esforço, estudo, empenho e dedicação, mas não poderia ser terminado sem a ajuda de diversas pessoas, instituições e especialmente este bonito país, que me acolheu como um outro filho.

Devo agradecer profundamente a uma pessoa muito especial e sincera, ao Prof. Dr. Paulo Kurka, pela oportunidade que me deu para vincular-me a esta prestigiosa Universidade. Além disso, pela orientação, amizade, apoio, confiança e por fornecer os meios e ferramentas necessárias para concluir este projeto com sucesso.

A meus colegas do laboratório de processamento de sinais e análise de sistemas dinâmicos, Carlos Mingoto, Jaime Izuka, Victor Bini, Vinícius de Souza, Jaime delgado e Elvira Rafikova, pelas sugestões, troca de idéias, experiências e ajuda durante toda minha estadia no Brasil.

Ao Prof. Eleri Cardozo, por me permitir realizar os testes no Robô Pioneer 3.

À minha família que me infundiram à ética e rigor necessário para guiar minha vida. Hoje quero agradecer-lhes pelo apoio incondicional, amor e compreensão, permitindo-me cumprir com meus sonhos.

A uma pessoa muito especial em minha vida, minha namorada Maria Fernanda, fonte de alegria e motivação. A ela quero agradecer por sua tolerância, amor, apoio e suas palavras de alento nos momentos mais difíceis, fazendo minha estadia mais agradável.

A minha família em Brasil Jeffrey Leon, Julian Prada, Ana Maria, Maria Isabel, Rodrigo Martinez e Luis Fuentes, que me acompanharam, apoiaram, aconselharam e motivaram nos momentos mais difíceis, fazendo mais agradável está historia de minha vida, de coração muito obrigado pela amizade e todas as lembranças que levo em minha mala.

A meu grande amigo Oscar Cárdenas, colega e irmão de grandes batalhas, me acompanhando desde a graduação na Colômbia. Muito obrigado pelas ensinias, conselhos, sorrisos e todas as histórias vividas. Embora estivesse na gloria de Deus, sempre viverá em meu coração.

Agradeço a CAPES pelo financiamento deste trabalho, fundamental para realizar meu mestrado.

La vida es cuestión de aguante
† *Oscar Leonardo Cárdenas*

Resumo

DIAZ, Carlos Andrés Espinosa, *Uma Aplicação de Navegação Robótica Autônoma através de Visão Computacional Estéreo*, Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2010. 106 p. Dissertação de Mestrado.

O presente trabalho descreve uma técnica de navegação autônoma, utilizando imagens estereoscópicas de câmeras para estimar o movimento de um robô em um ambiente desconhecido. Um método de correlação de pontos em imagens unidimensionais é desenvolvido para a identificação de pontos homólogos de duas imagens em uma cena. Utilizam-se métodos de segmentação de bordas ou contornos para extrair as principais características inerentes nas imagens. Constrói-se um mapa de profundidade dos pontos da imagem com maior similitude dentre os objetos visíveis no ambiente, utilizando um processo de triangulação. Finalmente a estimação do movimento bidimensional do robô é calculada aproveitando a relação epipolar entre dois ou mais pontos em pares de imagens. Experimentos realizados em ambientes virtuais e testes práticos verificam a viabilidade e robustez dos métodos em aplicações de navegação robótica.

Palavras chave: *Navegação de robôs móveis - Localização, visão de robô, processamento de imagens, mapeamento (Matemático), simulação por computador.*

Abstract

DIAZ, Carlos Andrés Espinosa, *Autonomous Application of Robotic Navigation Using Computer Stereo Vision*, Campinas: Faculty of Mechanical Engineering, Universidade Estadual de Campinas, 2010. 106 p. Dissertation Master.

The present work describes a technique for autonomous navigation using stereoscopic camera images to estimate the movement of a robot in an unknown environment. A one-dimensional image point correlation method is developed for the identification of similar image points of a scene. Boundary or contour segments are used to extract the principal characteristics of the images. A depth map is built for the points with greater similarity, among the scene objects depicted, using a triangulation process. Finally, the bi-dimensional movement of a robot is estimated through epipolar relations between two or more correlated points in pairs of images. Virtual ambient and practical robot tests are performed to evaluate the viability of employment and robustness of the proposed techniques.

Key Words: Mobile robot navigation – Location, robot vision, image processing, mapping (Mathematics), computer simulation.

Lista de Ilustrações

Figura 1.1 Metodologia de desenvolvimento do Projeto.....	6
Figura 1.2 Robô Pioneer 3, utilizado para validar os procedimentos de navegação	7
Figura 2.1 Exemplo de mapa métrico.....	11
Figura 2.2 Exemplo da construção de um mapa topológico, (a) Mapa do ambiente onde o robô foi inserido, (b) Criação do mapa topológico do ambiente.....	12
Figura 2.3 Problema de correspondência de dados	14
Figura 2.4 O modelo do FastSLAM, onde cada partícula define uma hipótese sobre a trajetória do robô.....	20
Figura 3.1 Modelo pinhole	22
Figura 3.2 Modelo em perspectiva de dois pontos no espaço	22
Figura 3.3 Semelhança de triângulos.....	23
Figura 3.4 Geometria de projeção de um ponto	24
Figura 3.5 Visão estéreo com câmeras paralelas.....	26
Figura 3.6 Correspondência baseada em feições.....	27
Figura 3.7 Correspondência baseada em áreas.....	28
Figura 3.8 Pontos de oclusão entre duas imagens paralelas com translação de 280 mm no eixo x	30
Figura 3.9 Metodologia de posicionamento do robô.....	30
Figura 3.10 Fluxograma do processamento de imagens.....	32
Figura 3.11 Simplificação da imagem. (a) Ambiente simulado em Blender, (b) mapa de intensidade unidimensional (Intensidade x Pixel) dos objetos.....	33
Figura 3.12 Isolamento dos pontos de interesse.....	34
Figura 3.13 Pontos candidatos com nível 0,9.....	36
Figura 3.14 Filtragem de intensidade com valor mínimo de 0,9.....	37
Figura 3.15 Comparação entre as etapas 4, 5 e a validação dos pontos	38

Figura 4.1 Imagens obtidas a partir de duas câmeras virtuais	42
Figura 4.2 Vista superior de um ambiente simulado	44
Figura 4.3 Simulação do ambiente I: (a) Primeira posição do robô, (b) Segunda posição do robô	44
Figure 4.4 Simulação do ambiente II: (a) Posição do robô longe dos objetos, (b) Posição do robô perto dos objetos.....	46
Figura 4.5 Simulação do ambiente IV	47
Figura 4.6 Descrição da correlação: Correlação I, correlação entre as imagens “CE_PAN” e “CE_PAT”. Correlação II, correlação entre os pontos obtidos em Correlação I e a imagem “CD_PAN”. Correlação III, correlação entre os pontos obtidos em Correlação II e a imagem “CD_PAT”	49
Figura 4.7 Segmentação da imagem CE_PAN.....	50
Figura 4.8 Correlação de pontos da simulação I: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT	51
Figura 4.9 Mapa de profundidade de cada posição, em vista superior. A câmera esquerda é o referencial relativo (0,0) do sistema de coordenadas	52
Figura 4.10 Correlação de pontos da simulação II, entre L1 e L2: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	54
Figura 4.11 Correlação de pontos da simulação II, entre P1 e P2: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	55
Figura 4.12 Mapas de profundidade das posições L1 e L2	56
Figura 4.13 Mapas de profundidade das posições P1 e P2.....	56
Figura 4.14 Correlação de pontos da simulação II, entre L2 e L3: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	58
Figura 4.15 Correlação de pontos da simulação II, entre P2 e P3: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	59
Figura 4.16 Mapas de profundidade das posições L2 e L3	60
Figura 4.17 Mapas de profundidade das posições P2 e P3.....	60
Figura 4.18 Comparação entre o movimento real e o estimado pelo método: (a) Posicionamento do robô longe dos objetos (L2’ e L3’), (b) Posicionamento do robô perto dos objetos (P2’ e P3’)	62
Figura 4.19 Correlação de pontos da simulação III, entre L1 e L3: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	64

Figura 4.20 Correlação de pontos da simulação III, entre P1 e P3: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	64
Figura 4.21 Mapas de profundidade das posições L1 e L3	65
Figura 4.22 Mapas de profundidade das posições P1 e P3.....	65
Figura 4.23 Comparação de movimento real e o estimado: (a) Posicionamento do robô longe dos objetos, (b) Posicionamento do robô perto dos objetos.....	67
Figura 4.24 Correlação de pontos da simulação IV, entre a posição 1 e 2: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	69
Figura 4.25 Mapas de profundidade das posições 1 e 2	69
Figura 4.26 Correlação de pontos da simulação IV, entre a posição 3 e 4: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	70
Figura 4.27 Mapas de profundidade das posições 3 e 4.....	70
Figura 4.38 Correlação de pontos da simulação IV, entre a posição 5 e 6: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	71
Figura 4.29 Mapas de profundidade das posições 5 e 6	71
Figura 4.30 Correlação de pontos da simulação IV, entre a posição 7 e 8: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	72
Figura 4.31 Mapas de profundidade das posições 7 e 8	72
Figura 4.32 Correlação de pontos da simulação IV, entre a posição 9 e 10: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	73
Figura 4.33 Mapas de profundidade das posições 9 e 10.....	73
Figura 4.34 Correlação de pontos da simulação IV, entre a posição 11 e 12: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	74
Figura 4.35 Mapas de profundidade das posições 11 e 12	74
Figura 4.36 Correlação de pontos da simulação IV, entre a posição 12 e 13: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT.....	75
Figura 4.37 Mapas de profundidade das posições 12 e 13	75
Figura 4.38 Simulação do percurso do robô na simulação de ambiente IV	77
Figura 5.1 Dimensões do robô Pioneer 3	80
Figura 5.2 Partes do robô Pioneer 3. (a) Visão frontal do robô, (b) Painel de controle do usuário, (c) Visão traseira do robô	80

Figura 5.3 Conexão do cliente. (a) Modem de rádio, (b) Ethernet sem fio, (c) Laptop a bordo, (d) Autônomo	81
Figure 5.4 Câmera IP AXIS 213 PTZ	82
Figura 5.5 Fixação das câmeras. (a) Base da câmera, (b) Câmera sobre a base	83
Figura 5.6 Vista do primeiro ambiente.....	85
Figura 5.7 Vista do segundo ambiente	86
Figura 5.8 Método de correlação passo a passo: Correlação I, correlação entre as imagens “CE_PAN” e “CE_PAT”. Correlação II, correlação entre os pontos obtidos em Correlação I e a imagem “CD_PAN”. Correlação III, correlação entre os pontos obtidos em Correlação II e a imagem “CD_PAT”.....	87
Figura 5.9 Correlação de pontos entre a posição 1 e 2: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT	88
Figura 5.10 Mapa de profundidade de cada posição em vista superior.....	89
Figura 5.11 Comparação de movimento real e o movimento estimado	91
Figura 5.12 Correlação de pontos entre a posição 1 e 2: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT	92
Figura 5.13 Comparação de movimento real e o movimento estimado	92
Figura 5.14 Correlação de pontos entre a posição 2 e 3: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT	93
Figura 5.15 Comparação de movimento real e o movimento estimado	93
Figura 5.16 Correlação de pontos entre a posição 3 e 4: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT	94
Figura 5.17 Comparação de movimento real e o movimento estimado	94
Figura 5.18 Correlação de pontos entre a posição 4 e 5: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT	95
Figura 5.19 Comparação de movimento real e o movimento estimado	95
Figura 5.20 Correlação de pontos entre a posição 5 e 6: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT	96
Figura 5.21 Comparação de movimento real e o movimento estimado	96
Figura 5.22 Correlação de pontos entre a posição 6 e 7: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT	97
Figura 5.23 Comparação de movimento real e o movimento estimado	97

Figura 5.24 Correlação de pontos entre a posição 7 e 8: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT	98
Figura 5.25 Comparação de movimento real e o movimento estimado	98
Figura 5.26 Correlação de pontos entre a posição 8 e 9: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT	99
Figura 5.27 Comparação de movimento real e o movimento estimado	99
Figura 5.28 Percurso do robô no ambiente.....	100

Lista de Tabelas

Tabela 4.1 Coordenada dos objetos do ambiente I.....	45
Tabela 4.2 Coordenada da câmera esquerda até o referencial absoluto do ambiente I.....	45
Tabela 4.3 Coordenada dos objetos do ambiente II.....	46
Tabela 4.4 Coordenada da câmera esquerda	46
Tabela 4.5 Coordenada dos objetos do ambiente IV	47
Tabela 4.6 Trajetória do robô em direção aos objetos.....	48
Tabela 4.7 Comparação de profundidade do primeiro ambiente.....	53
Tabela 4.8 Estimativa de movimento no ambiente I	53
Tabela 4.9 Comparação de profundidade na posição L1.....	57
Tabela 4.10 Comparação de profundidade na posição L2.....	57
Tabela 4.11 Comparação de profundidade na posição P1	57
Tabela 4.12 Comparação de profundidade na posição P2.....	57
Tabela 4.13 Comparação de profundidade na posição L3.....	61
Tabela 4.14 Comparação de profundidade na posição P3.....	61
Tabela 4.15 Comparação do movimento gerado no aplicativo Blender com o calculado	62
Tabela 4.16 Comparação do movimento gerado no aplicativo Blender com o calculado	63
Tabela 4.17 Comparação de profundidade na posição L1.....	66
Tabela 4.18 Comparação de profundidade na posição L3.....	66
Tabela 4.19 Comparação do movimento gerada no aplicativo Blender com o calculado.....	67
Tabela 4.20 Comparação do movimento gerado no aplicativo Blender com o calculado	67
Tabela 4.21 Número de pontos correlacionados e erro apresentado na estimativa de movimento durante as diferentes posições em que foram obtidas as imagens.....	76
Tabela 5.1 Coordenada de movimento do robô no ambiente I.....	84
Tabela 5.2 Coordenada dos objetos no ambiente I.....	84
Tabela 5.3 Coordenada de movimento do robô no ambiente II	85

Tabela 5.4 Coordenada dos objetos no ambiente II.....	86
Tabela 5.5 Comparação de profundidade	90
Tabela 5.6 Comparação do movimento real e estimado.....	90
Tabela 5.7 Coordenada da estimativa de movimento.....	100

Lista de Abreviaturas e Siglas

Letras Latinas

f	- foco	[mm]
k	- coordenada do pixel no eixo X em relação ao centro da janela	[pixel]
l	- coordenada do pixel no eixo Z em relação ao centro da janela	[pixel]
n	- estrutura da correlação	
p	- projeção no plano de imagem da câmera esquerda	[pixel]
q	- projeção no plano de imagem da câmera direita	[pixel]
r	- distância do pixel da imagem esquerda	[mm]
s	- distância do pixel da imagem direita	[mm]
t	- translação entre as câmeras	[mm]
t_x	- translação do robô na coordenada X	[mm]
t_z	- translação do robô na coordenada Z	[mm]
v	- ruído branco de observação	
w	- ruído branco do processo	
x	- estado verdadeiro	
y	- observação do estado verdadeiro	
A	- matriz de transição de estado	
B	- modelo de controle de entrada	
C	- matriz que transforma o espaço de estados reais no espaço de observações	
H	- parâmetro de intensidade	
I	- intensidade	
\bar{I}	- matriz identidade	
K	- ganho de Kalman	

P	- ponto no espaço
P^-	- erro a priori da matriz de covariância
\hat{P}	- matriz de covariância do erro atual
R	- matriz de rotação
\hat{S}	- solução ao sistema
U	- vetor de controle
V	- matriz de covariância do ruído de medida
W	- matriz de covariância do ruído do processo
X^-	- estado a priori
\hat{X}	- estado a posteriori
X, Y, Z	- sistema de coordenadas

.....

Letras Gregas

Θ	- ângulo do robô	[rad]
λ	- mapa de profundidade	
σ	- desvio padrão da imagem	

.....

Subscritos

ca	- câmera a
cb	- câmera b
k	- k-ésima
i	- i-ésima

.....

Abreviações

API	- application programming interface
CCD	- charge coupled device
CE_PAN	- câmera da esquerda posição anterior
CE_PAT	- câmera da esquerda posição atual
CD_PAN	- câmera da direita posição anterior
CD_PAT	- câmera da direita posição atual
EKF	- extended Kalman filter
GPS	- global positioning system
HMM	- hidden markov models
IEEE	- institute of electrical and electronics engineers
LMedS	- the least median of squares
R-B	- rao-blackwellization
SLAM	- simultaneous localization and mapping

SUMÁRIO

1. Introdução.....	1
1.1. Motivação e revisão da literatura.....	2
1.2. Objetivos do trabalho.....	5
1.3. Metodologia.....	6
1.4. Estrutura da tese.....	7
2. NAVEGAÇÃO AUTÔNOMA	9
2.1. Tipos de Mapas.....	10
2.1.1. Mapas métricos (Grid-based Maps)	10
2.1.2. Mapas Topológicos	11
2.1.3. Mapas de Características	13
2.2. O problema SLAM	13
2.3. Solução do problema de SLAM.....	15
2.3.1. Filtro de Kalman para o problema de SLAM.....	16
2.3.2. FastSLAM	19
3. CONSTRUÇÃO DE MAPAS BIDIMENSIONAIS A PARTIR DE DUAS CÂMERAS	21
3.1. Câmeras pinhole	21
3.2. Geometria projetiva	23
3.3. O problema da correlação estéreo.....	27
3.4. Metodologia para o posicionamento do robô.....	30
3.4.1. O método de correlação de imagens unidimensional	31
3.4.2. Determinação do mapa de profundidade	38
3.4.3. Estimativa de movimento bidimensional a partir de imagens estéreo	39
4. Proposta de uso de visão para navegação robótica em ambientes virtuais	41
4.1. Estrutura do ambiente virtual.....	42

4.1.1. Metodologia da simulação em ambientes virtuais.....	43
4.2. Simulação de navegação robótica em ambientes virtuais.....	43
4.3. Análise e resultados	48
5. Aplicação dos métodos em ambientes reais.....	79
5.1. Especificações técnicas do robô Pioneer 3	79
5.2. Especificações técnicas da câmera.....	81
5.3. Posicionamento real do robô.....	83
5.4. Análise e resultados	86
6. Conclusões e sugestões para trabalhos futuros	103
6.1. Conclusões	103
6.2. Sugestões para trabalhos futuros.....	104

1. Introdução

A importância da robótica estabeleceu-se a partir da segunda guerra mundial, com a necessidade de automatizar o processo de produção nas indústrias, diminuindo o risco de acidentes e o tempo na realização das atividades. Surge daí também a necessidade que o robô realize deslocamentos durante o desenvolvimento das atividades de produção, dando início à robótica móvel.

Desde então, surgiram dois grandes desafios que possibilitam a inserção dos robôs móveis na indústria, o que é alvo de um grande número de pesquisas atuais. O primeiro problema surge em conhecer em todo momento a localização do robô em um determinado ambiente, para executar estratégias de tomada de decisão para ir desde sua posição até seu objetivo de forma autônoma, cumprindo as tarefas estabelecidas. O segundo problema reside em sua interação com um ambiente arbitrário. Nesse contexto, este projeto pretende discutir um método de posicionamento e geração de mapas do entorno de navegação de um robô móvel autônomo em ambientes de interiores através de visão computacional estéreo.

No presente trabalho estudam-se métodos de correlação de imagens para o desenvolvimento de uma aplicação de robótica móvel autônoma, apresentando a revisão da literatura com respeito ao tratamento de imagens. Para isto, é definida a correlação de pontos usada na análise das imagens. Neste estudo a análise é feita para duas imagens em cada posição do robô. Em seguida é aplicado o método de correlação de pontos, modificado a imagens unidimensionais para o processamento dos dados e obtenção de mapas de profundidade. O objetivo principal do presente trabalho é simular em ambientes virtuais a estimativa de movimento bidimensional do robô para a obtenção de mapas de posicionamento, avaliando também o tempo de processamento da informação. As aplicações desenvolvidas são validadas ainda através de testes experimentais com o robô Pioneer 3.

1.1. Motivação e revisão da literatura

O uso da visão computacional como ferramenta de navegação na robótica representa um grande desafio para o controle de movimento de um robô móvel, particularmente em relação à possibilidade de estimar a sua posição e direção de movimento. Tais desafios originam-se no fato de inserir o robô em ambientes desconhecidos. Para estimar as variáveis de posição e direção de movimento do robô torna-se necessário realizar o cálculo de coordenadas absolutas ou relativas a uma referência de navegação no ambiente [36], e a execução de estratégias de navegação para deslocar-se de um ponto a outro evitando colisões. Outras estratégias de navegação baseiam-se no posicionamento absoluto ou relativo com respeito a um mapa, ou aos objetos do ambiente.

O uso da visão computacional em técnicas de navegação tem permitido melhorias na robótica móvel, no sentido de proporcionar maior autonomia e capacidade na tomada de decisões com respeito às trajetórias de navegação. Cita-se como exemplos, a exploração espacial, exploração submarina, vigilância, navegação de veículos não tripulados, manufatura, entre outros.

Um dos processos básicos da visão computacional é o reconhecimento de objetos no espaço para obter informações úteis sobre uma cena e fornecer informação ao robô para gerar mapas de profundidade destes objetos e finalmente obter sua localização. O reconhecimento de objetos está ligado com o sensoriamento do ambiente e a interpretação desses dados. Geralmente os robôs móveis estão equipados por uma variedade de sensores para o reconhecimento do entorno (como sonares, *encoders*, laser e visão estéreo, entre outros) que lhe permitem obter informações de um mesmo objeto, a fim de executar algoritmos de fusão sensorial e estimar dados mais confiáveis para reconstruir um ambiente, evitando as limitações que geram o uso de um sensor só. A literatura apresenta diversos trabalhos da fusão sensorial que facilitam a construção de mapas do ambiente para a navegação de robôs móveis [17, 22, 39, 42, 44].

Alguns exemplos do reconhecimento de objetos utilizando uma única câmera por robôs móveis foram desenvolvidos por:

Bianchi [3], descreve um sistema de controle para um robô móvel autônomo. O mini-robô Khepera possui oito sensores de luz infravermelha que permite medir a proximidade dos objetos através da reflexão da luz para assim desviar os obstáculos. Além disso, o robô está equipado com uma câmera para controlar sua trajetória em um caminho desenhado. O objetivo da câmera é extrair parâmetros que permitam associar regiões da imagem com objetos da cena. Sobre o robô é inserido um rótulo contendo dois círculos de diferentes cores: um no centro geométrico do robô e o outro na frente deste, indicando o sentido da movimentação. Deste modo, o vetor de direção do robô é um vetor que passa pelos dois centros circulares, sendo estes determinados pela identificação das cores (técnica de imposição de limiares).

Para a determinação da direção do caminho a seguir, observa-se a intersecção de um quadro imaginário ao redor do centro geométrico do robô e a cor destinada a demarcação do caminho. O deslocamento do robô é extraído das informações entre ambos os vetores.

Costa et al. [8], descrevem um robô que utiliza imagens para seguir a trajetória desenhada em uma superfície. O centro da pista é calculado a partir da imagem coletada e o sistema de controle decide a direção de movimento do robô. Nesse caso ainda, somente uma câmera é utilizada para a obtenção das imagens.

Gasper [16], utiliza um conjunto de espelhos planos e uma única câmera para adquirir imagens estéreo da área na frente do robô. O algoritmo de detecção de obstáculos sobre pavimento plano é baseado na hipótese que um objeto é qualquer ponto não pertencente ao plano do pavimento. Durante a fase de inicialização do robô, um conjunto de pontos correspondentes, selecionados manualmente, é usado para estimar o mapa de disparidade associado ao plano do pavimento. Este mapa é a referência para estimar os pontos não pertencentes ao plano, detectando assim os obstáculos ou pontos que violem este mapa.

No âmbito de visão estéreo, diversos métodos de correlação de pontos são encontrados na literatura para o reconhecimento de objetos. Entre eles destacam-se *Heipke* [18] apresentando um

resumo sobre orientação automática em imagens. Nogueira [29] por sua parte realiza correspondência bidirecional utilizando janelas de busca organizadas em ordem crescente de tamanho. Pajares [30] utiliza mínimos quadrados da distância de Mahalanobis entre a diferença das características principais e uma nuvem de pontos em torno ao centro. Want et al [43], utilizam um fator de ponderação em relação de cada ponto da imagem para encontrar as correlações. Zhou e Shi [46] utilizam o método de atribuição padrão para solucionar o problema de correlação. Por outro lado, Zhang, et al., [45] propõem um método robusto para a correlação de pontos entre duas imagens de uma mesma cena, aproveitando a restrição epipolar. A idéia subjacente é utilizar técnicas clássicas para encontrar pontos correlatos e depois utilizar a técnica LMedS (the Least Median of Squares) para excluir correlações falsas.

No âmbito do Laboratório de Processamento de Sinais e Análise de Sistemas Dinâmicos da Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas - UNICAMP, diferentes trabalhos foram realizados utilizando câmeras como instrumentos de medida geométrica e navegação robótica. Menezes [28], desenvolveu uma técnica para o registro e processamento dos sinais de movimento de um modelo reduzido de *riser* de produção de petróleo, com a utilização de câmeras. Os *risers* são dutos suspensos utilizados para movimentar os fluidos petrolíferos entre o fundo do mar e a plataforma de extração de petróleo. O modelo de *riser* possui alvos fixados com esferas de isopor de cor laranja no duto para obter informação de sua localização e o padrão de movimento, a partir de uma seqüência de imagens. Devido ao comprimento extenso do duto, foram utilizadas seis câmeras para a captura do movimento ao longo do duto, dividindo-o em três partes. Após obter a seqüência de imagens são identificados os alvos através de técnicas de caixas de contorno 2-D [23] para calcular o ângulo existente entre os alvos e encontrar a freqüência das oscilações e amplitudes do movimento do sistema. Diogenes [10], por sua parte, estabelece um conjunto de procedimentos básicos de navegação e controle de robôs autônomos, baseados na fusão sensorial de imagens e odometria, para estimar sua posição ótima. Neste modelo a obtenção da posição por odometria é calculada a partir da informação medida pelos *encoders* de rotação das rodas, considerando o modelo matemático do robô. De outro lado, a estimativa de posição pelas câmeras é baseada na geometria espacial da reconstrução dos pontos projetados nas imagens. Porém, os objetos em comum localizados por

um processamento de mapas de intensidade, são projetados no espaço, dando origem a um mapa de profundidade que permite obter sua posição, comparado este mapa com as coordenadas reais dos objetos. Obtidas as estimativas de posicionamento por odometria e pelas câmeras, o método de controle utiliza o filtro de Kalman para estimar sua posição ótima e executar comando de controle nas rodas do robô, realizando trajetórias retilíneas ideais para chegar ao objetivo final ou até um portal de passagem escolhido dentre os obstáculos presentes em sua trajetória. O trabalho descrito por Diogenes utiliza idéias fundamentais sobre o uso de câmeras e controle de trajetórias de um robô móvel. O processo de navegação utilizado nesse trabalho ocorre em um ambiente com obstáculos fixos, conhecendo suas coordenadas *a priori*. Esses obstáculos servem como pontos de referencia para o robô sair de sua posição inicial e chegar até um objetivo final. Em decorrência do anterior, Diogenes proporciona fundamentos teóricos importantes sobre a geometria projetiva de pontos e uso de visão computacional na navegação robótica autônoma, sendo ponto de partida para realizar estudos mais aprofundados no presente trabalho, sobre o processamento de imagens reais aplicadas à estimativa de posicionamento do robô em um ambiente de interior, sem o prévio conhecimento das coordenadas dos objetos. Por outro lado, complementa-se com técnicas de localização e mapeamento simultâneos (SLAM) [5, 9, 19, 41] para navegar de uma forma mais confiável.

1.2. Objetivos do trabalho

Este trabalho tem como objetivo geral desenvolver uma técnica de navegação autônoma em ambientes de interiores utilizando o processamento de imagens provenientes de duas câmeras para guiar um robô móvel sem prévio conhecimento do ambiente.

Para esse propósito apresentam-se os seguintes objetivos específicos:

- Desenvolvimento de um algoritmo de correlação de pontos entre duas imagens.

- Eleição e uso do método de estimação da profundidade dos objetos inseridos no ambiente.
- Desenvolvimento de um algoritmo para determinar o movimento bidimensional a partir de duas posições do robô.
- Teste experimental do robô Pioneer através de simulações em ambientes virtuais.

1.3. Metodologia

Foi traçada uma metodologia para o cumprimento dos objetivos. Esta metodologia é sintetizada na figura 1.1, em que se apresenta a distribuição do trabalho para chegar aos resultados finais, destacando que a revisão da literatura foi feita durante tudo o projeto e as atividades foram desenvolvidas de maneira escalonada para obter os resultados e conclusões sobre o posicionamento do robô em ambientes virtuais e reais.

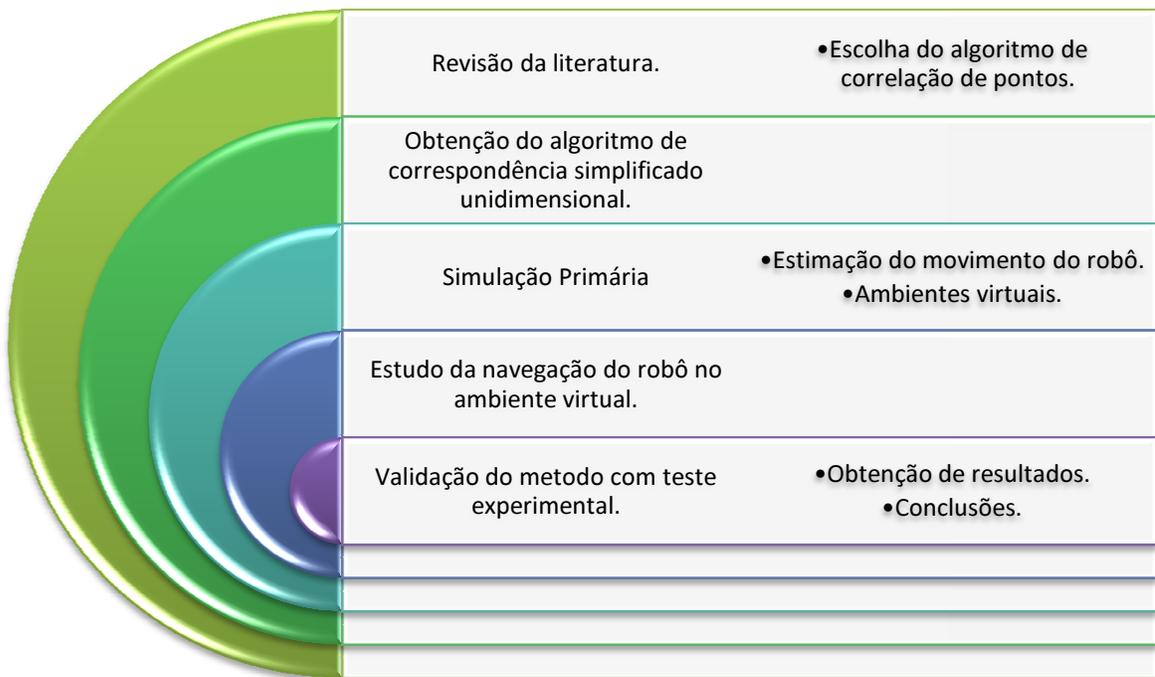


Figura 1.1 Metodologia de desenvolvimento do Projeto

A metodologia descrita na figura anterior possui três etapas. Começa-se com a escolha do método de correlação de pontos entre duas imagens, sendo este método simplificado para imagens unidimensionais e testado com imagens capturadas com câmeras em ambientes virtuais. A segunda etapa da simulação ocorre com o uso do algoritmo de correlação de pontos robusto onde se resolve o problema de mapeamento de pontos correlatos em um mapa métrico, sendo a primeira posição do robô a coordenada absoluta do sistema de mapeamento. Finalmente, depois de se realizar um deslocamento, capturando imagens numa posição subsequente, estima-se o movimento que houve entre duas posições, o que é validado com a análise dos resultados obtidos por simulação e testes experimentais com o robô Pioneer 3 (Figura 1.2). Estes métodos são implementados usando a ferramenta matemática *MatlLab*, para realizar o processamento numérico e o software livre *Blender* para a simulação do ambiente virtual.



Figura 1.2 Robô Pioneer 3, utilizado para validar os procedimentos de navegação

1.4. Estrutura da tese

No capítulo 2 é mostrada a teoria dos problemas de localização e mapeamento simultâneos (SLAM) na robótica móvel. Na primeira seção desse capítulo são apresentados os alguns tipos de

mapas utilizados na robótica. Na segunda seção do capítulo 2 aborda-se o problema de navegação através de localização simultânea e mapeamento, SLAM.

O capítulo 3 é dedicado aos métodos de visão computacional, relevantes neste trabalho. Na primeira seção apresenta-se o modelo de câmera *pinhole* e o modelo em perspectiva de dois pontos no espaço. A segunda seção é apresentada a geometria projetiva dos pontos no espaço baseado na semelhança de triângulos para obter mapas de profundidade do ambiente. A terceira e quarta seções desse capítulo abordam os problemas da correlação e o método proposto para a correlação de imagens unidimensionais. Última seção é dedicada à determinação do movimento do robô a partir das imagens correlatas em duas posições.

A primeira análise dos diferentes métodos propostos para a navegação de robôs autônomos é apresentado no capítulo 4. Utilizando Blender como software de simulação gráfica, são propostos quatro ambientes controlados para comparar os dados fornecidos neste software com os dados estimados pelos métodos, realizando a análise do comportamento dos métodos nas diferentes condições. Finalmente é feito uma análise geral, comparando os resultados de cada ambiente.

A segunda análise é feita a partir dos experimentos realizados para estimar o movimento do robô Pioneer 3 utilizando duas câmeras. Estes experimentos são apresentados no capítulo 5. Sendo mostradas nas primeiras seções as especificações técnicas do robô e das câmeras. Seguidamente são descritos os ambientes nos quais o robô vai interagir para apresentar os análises e resultados.

No capítulo 6 expõem-se as conclusões obtidas no decorrer do estudo realizado. Também são apresentadas as possíveis melhoras ou linhas a seguir neste trabalho.

2. NAVEGAÇÃO AUTÔNOMA

No século passado, durante o curso da segunda guerra mundial se deu início a inclusão de robôs na indústria como resposta à necessidade de automatizar o processo de produção, visto que eles conseguiam fazer as tarefas repetitivas com maior velocidade e maior precisão do que os homens. No entanto, a necessidade de fazer tarefas que requeriam o deslocamento do robô, deu origem aos primeiros veículos guiados, levando assim ao início da robótica móvel.

Com a robótica móvel surgiram vários problemas que impediram a inclusão dos robôs móveis na indústria e geraram campos de pesquisa para a solução desses problemas. As principais dificuldades originam-se primeiro, na necessidade de se conhecer a posição do robô em qualquer momento para poder planejar sua trajetória até o objetivo, e em segundo lugar, com a necessidade de estimar o mapa do ambiente no qual o robô vai se movimentar. Visando a superação desses problemas foi desenvolvida a técnica do SLAM (*Simultaneous Localization and Mapping*), ou localização e mapeamento simultâneos. O SLAM procura determinar a localização do robô e um mapa do seu ambiente através dos dados adquiridos na sua exploração.

O SLAM foi apresentado pela primeira vez em uma conferência de robótica e automatização da IEEE em 1986 em San Francisco, California. Neste evento, os pesquisadores Meter Cheeseman, Jim Crowley e Hugh Durrant-Whyte falaram da aplicação de métodos de estimação para os problemas de mapeamento e localização. A conferência deu como resultado o reconhecimento da análise probabilística na solução destes problemas, porém requeria uma carga computacional e conceitual complexa para ser tratado.

Ao mesmo tempo Ayache e Faugeras desenvolviam um sistema de navegação robótica com visão computacional. Crowley, Chatila e Laumond desenvolviam um sistema de navegação robótica baseados em sonares utilizando o filtro de Kalman. A ligação dos desenvolvimentos de navegação e a teoria probabilística deram início à solução do SLAM, na qual propuseram que a

solução completa e consistente aos problemas de mapeamento e localização, requeria o conjunto de estados da posição do robô e as posições dos objetos, para assim atualizar a posição do robô mediante a observação contínua dos objetos. A estrutura do problema do SLAM e o acrônimo SLAM foram apresentados pela primeira vez no Simpósio Internacional de Investigação de robótica de 1995 [13].

2.1. Tipos de Mapas

Muito dos problemas envolvidos na robótica móvel podem ser resolvidos com a construção de um mapa. Um mapa é a representação gráfica e métrica utilizada para localizar elementos em um espaço determinado. A construção de mapas depende de vários elementos, entre eles o ambiente no qual o robô está inserido, os tipos de sensores que ele utiliza e a tarefa que vai desenvolver. A finalidade dos mapas na robótica é planejar uma trajetória entre uma posição inicial e um destino; e/ou localizar o robô no ambiente. Em qualquer caso, é possível destacar três tipos de mapas.

2.1.1. Mapas métricos (Grid-based Maps)

Os mapas métricos utilizam um sistema de coordenadas similar ao sistema de coordenadas polares para representar o ambiente, gerando um modelo do ambiente desde o ponto de vista do robô [14].

O objetivo dos mapas métricos é representar o ambiente com o maior número de detalhes possíveis sobre os objetos, como tamanho e forma, ou sobre o ambiente, como os limites das áreas livres para a navegação, tais como quartos e corredores. Esta técnica divide o espaço como um denso conjunto de células indicando alguma característica do ambiente, já seja ocupado (1) ou livre (0).

Neste tipo de mapa é importante considerar três problemas básicos quais sejam manter a localização do robô, explorar o ambiente de forma completa e eficiente, e atualizar o valor de probabilidade de ocupação de acordo com os dados dos sensores.

Um exemplo de mapa métrico e seus problemas, foi colocado por Begum Momotaz (2006) [2], Este integra o método da lógica nebulosa e um algoritmo genético para solucionar o problema da acumulação de erros na odometria em grandes distâncias e estimar o melhor mapa probabilístico associado à posição espacial. O mapa métrico final é apresentado na Figura 2.1, indicando com preto às células das áreas ocupadas, com branco às células das áreas livres e com cinza as células das áreas desconhecidas.

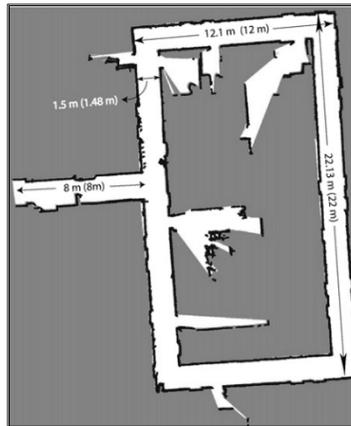


Figura 2.1 Exemplo de mapa métrico

Fonte [2] Begum, M., Mann, G., and Gosine, R. Integrated Fuzzy Logic and Genetic Algorithmic Approach for Simultaneous Localization and Mapping of Mobile Robots. ScienceDirect 2006 December.

Os mapas métricos, pelo fato de processar muitas células ao mesmo tempo, requerem para sua construção um alto custo computacional, o que é considerado uma desvantagem. Não obstante, tem a vantagem de executar a localização de forma mais simples e com maior precisão.

2.1.2. Mapas Topológicos

O objetivo dos mapas topológicos é representar por meio de uma estrutura relacional os lugares ligados que não possui objetos em um ambiente, com o propósito de representar por meio

de grafos alguns pontos de interesse no mapa, com um menor detalhamento o ambiente onde se está navegando.

A construção dos mapas topológicos é realizada em dois tipos de lugares, os “lugares previstos” e os “lugares confirmados”. Esta construção é feita de modo que o robô navega pelo ambiente verificando se os “lugares previstos” mais próximos têm obstáculos ou não. Se o caminho está livre o “lugar previsto” é modificado para “lugar confirmado”, em caso contrário o lugar é excluído do mapa [12].

Os mapas topológicos exigem menor esforço computacional em comparação com os mapas métricos, devido ao processamento de alguns nós do grafo por iteração, por isso são os mais apropriados quando se deseja mapear grandes áreas, como mostrado na Figura 2.2.

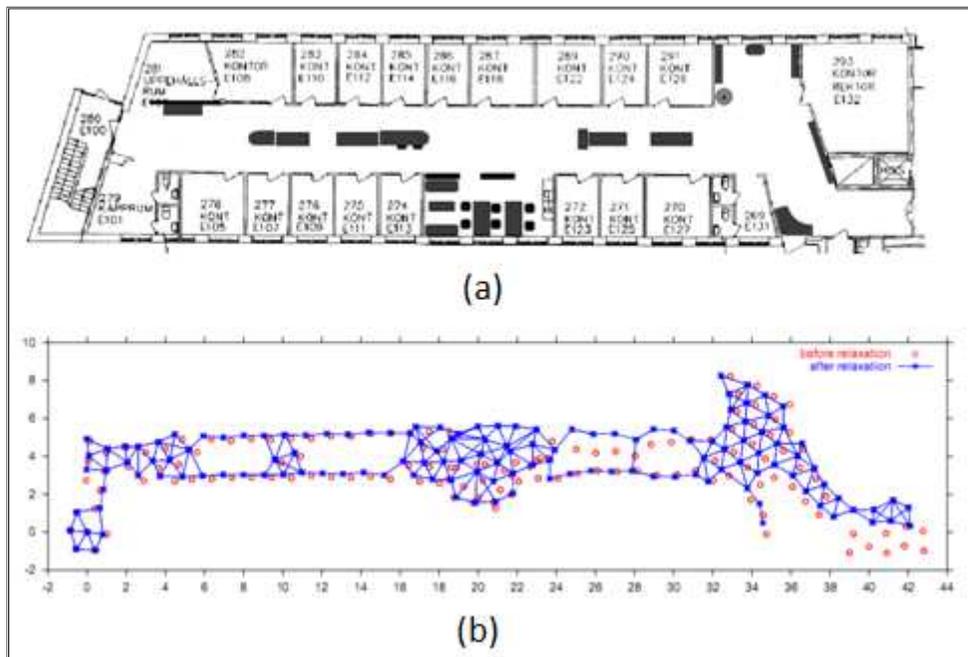


Figura 2.2 Exemplo da construção de um mapa topológico, (a) Mapa do ambiente onde o robô foi inserido, (b) Criação do mapa topológico do ambiente

Fonte [12]. Duckett, T., Saffiotti, A. Building Globally Consistent Gridmaps from Topologies, in: Proceedings of the Sixth International IFAC Symposium on Robot Control (SYROCO). 2000.

2.1.3. Mapas de Características

Os mapas de características são uma solução intermediária ao problema dos mapas métricos e topológicos, visto que este tipo de mapa representa algumas características geométricas do ambiente com linhas retas e polígonos.

A construção dos mapas de características é muito semelhante a dos mapas topológicos. A diferença destes é que não conectam os lugares livres formando um grafo, e sim geram um grafo com as características detectadas pelo robô. Deste modo, um lugar é descrito para efetuar certas tarefas na navegação robótica.

Descrever o ambiente por meio de linhas ou polígonos requer estruturas mais complexas para correlacionar as diferentes imagens.

Em qualquer dos tipos de mapas, os erros são proporcionados diretamente pela acumulação de erros sensoriais e de posicionamento do robô. Mas os erros sensoriais geralmente são limitados devido à captura de dados dos sensores em momentos diferentes. De outro lado, a existência de erros na localização do robô significa também a presença de erros no mapeamento. Por isso, se faz necessário realizar o mapeamento e a localização conjuntamente, conduzindo ao problema SLAM.

2.2. O problema SLAM

O problema SLAM estabelece a possibilidade de que um robô móvel consiga situar-se em uma posição desconhecida em um ambiente também desconhecido, construindo de maneira incremental um mapa do ambiente e simultaneamente ir determinando sua posição baseado neste mapa.

Segundo Thrum [40] o problema SLAM pode ser abordado de acordo com uma série de subproblemas. O primeiro subproblema corresponde na aquisição sensorial, visto que o robô

geralmente possui sensores como câmeras, sonares, laser, sensores de contato e GPS, que percebem os objetos no ambiente ou a posição do robô. De qualquer forma, os sensores estão sujeitos a erros que se somam ao erro do mapeamento e localização.

O segundo subproblema surge com a dimensionalidade dos elementos do mapa. Um exemplo do problema da dimensionalidade é quando se tem muitos pontos descrevendo o ambiente, isso pode ser representado por milhões de células no mapa, demandando um grande esforço computacional, sendo importante manter a complexidade computacional tratável.

O terceiro subproblema é a correspondência ou associação de dados. É importante ter certeza que os dados dos sensores tomados para diferentes pontos em diferentes tempos no ambiente correspondam ao mesmo objeto. Na Figura 2.3 se apresenta este problema no momento em que é fechado o ciclo, já que o robô não correlaciona os objetos detectados nesse instante com os objetos construídos no mapa anteriormente.

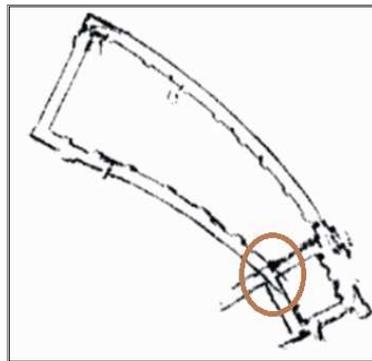


Figura 2.3 Problema de correspondência de dados

O último subproblema está relacionado com o fato de que os robôs devem escolher seu caminho durante o mapeamento. Este problema de gerar movimento no robô enquanto é construído o mapa, está geralmente designado à exploração robótica. Assim, qualquer estratégia de exploração deve ser capaz de adequar imprevistos que possam surgir durante a aquisição do mapa, como ocorre em ambientes dinâmicos. Este trabalho não considera a exploração com ambientes dinâmicos. De fato, na última década a maioria das pesquisas foram desenvolvidas em ambientes estáticos ou com pequenas variações. Uma abordagem com ambientes com variações

pequenas é apresentada por Drews e Botelho [11]. Estes autores usaram uma câmera de inspeção em um veículo subaquático para propor uma solução ao problema do SLAM.

Quando o robô escolhe sua trajetória, deve-se considerar a possível perda de informação da localização ou pose do robô ao longo do caminho e a informação no mapa. Por esta razão a literatura sugeriu resolver o problema do mapeamento e localização em conjunto, visando à solução do problema do SLAM.

2.3. Solução do problema de SLAM

Os algoritmos para a solução de localização e mapeamento simultâneos têm em comum a característica de utilizar modelos probabilísticos. Estes modelos podem ser derivados do modelo de Markov não observáveis (*Hidden Markov Models* - HMM).

O objetivo deste modelo é determinar parâmetros desconhecidos a partir de parâmetros observados. A teoria de HMM é tradicionalmente aplicada no reconhecimento de formas temporais, tais como reconhecimento de fala, de escritura manual, de gestos, entre outros [35]. Para esses problemas o desenvolvimento de técnicas na literatura tem sido amplamente estudado. No caso específico de aplicação do SLAM é adotada uma modelagem para o sistema, na qual o mapa está relacionado com as probabilidades de observação do modelo. Segundo isso, o algoritmo do SLAM deve identificar os parâmetros desconhecidos de HMM utilizando técnicas com alta complexidade computacional [33].

As soluções do problema SLAM implicam em encontrar uma representação aproximada para o modelo de observação e o modelo de movimentação que permita um cálculo eficiente e consistente com os passos iterativos de predição e correção. A representação mais comum é mediante o modelo estado-espço com ruído gaussiano adicionado, com a utilização do Filtro de Kalman Estendido (*Extended Kalman Filter* – EKF) [31]. O EKF é aplicado para manter uma estimativa da localização atual do robô e as localizações de algumas características geométricas, quase sempre pontos. Uma representação alternativa é descrever o movimento do robô como um

conjunto de distribuições de probabilidade não gaussianas, visando a implementação do filtro de partículas Rao-Blackwellized ou o algoritmo FastSLAM para convergir à solução do problema do SLAM [13].

2.3.1. Filtro de Kalman para o problema de SLAM.

Um das primeiras técnicas desenvolvidas na história do SLAM foi o filtro de Kalman. Este algoritmo proposto por Rudolf Emil Kalman é utilizado para prever e corrigir um sistema, ou seja, este algoritmo prognostica o novo estado a partir de uma estimativa prévia agregando um termo de correção ao erro de predição, sendo que este último é minimizado estatisticamente.

O filtro de Kalman (KF) é aplicado a partir de um sistema que está descrito por um modelo linear submetido a um ruído branco [32], onde o erro associado ao sistema e a informação adicional incorporada no mesmo sistema, tem distribuição normal com média zero e variância determinada. O objetivo do KF é calcular um estimador linear e ótimo do estado do sistema em um tempo t de acordo com a informação disponível em $t-1$ e atualizar estas estimativas com a informação adicional [7]. Dito em outras palavras, o KF para estimar um sistema dinâmico estável, gera uma cópia do sistema original para realizar medições do estado interno diretamente. Se o sistema original e sua cópia são submetidos aos mesmos estímulos, pode se esperar um comportamento similar no passar do tempo, já que seus estados internos tendem a ser similares. Para garantir uma convergência mais rápida entre ambos os sistemas, é realimentada a cópia do sistema com uma entrada do sistema original mais a diferença das saídas dos sistemas multiplicada por um fator calculado a partir das variâncias dos ruídos. Na equação 2.1 é apresentado o KF em tempo discreto e na equação 2.2 a medida de observação no tempo k do estado verdadeiro X_k .

$$x(k) = Ax(k-1) + BU(k) + w(k) \quad (2.1)$$

$$y(k) = Cx(k) + v(k) \quad (2.2)$$

O KF é dividido em dois grupos, o primeiro grupo referente às equações de predição e o segundo grupo referente à equação de correção. Às do primeiro grupo estão encarregadas de projetar o estado anterior (no instante $k-1$) no estado atual (k) e de atualizar a matriz de covariância do estado. O segundo grupo é encarregado da retroalimentação, ou seja, agrega informação nova à estimação anterior para melhorar o estado.

A primeira etapa consiste em prognosticar o estado do sistema e a matriz de covariância atual de acordo à informação disponível do estado anterior. Na equação 2.3 é estimado o estado a priori do sistema e o erro a priori da matriz de covariância.

$$\begin{aligned} X_k^- &= A\hat{X}_{k-1} + Bu_k \\ P_k^- &= \hat{P}_{k-1} + W_k \end{aligned} \quad (2.3)$$

A segunda etapa consiste em corrigir o estado do sistema. Para isto, primeiro é calculado o ganho de Kalman por médio da equação 2.4, onde este fator minimiza o erro de covariância do novo estado (a posteriori). O valor entre os parênteses é chamado de inovação.

$$K_k = P_k^- (P_k^- + V_k)^{-1} \quad (2.4)$$

Após é gerada uma nova estimação (a posteriori) do estado do sistema otimizado, dado pela equação a seguir.

$$\hat{X}_k = X_k^- + K_k(y_k - X_k^-) \quad (2.5)$$

Finalmente a equação 2.6 obtém a estimação da matriz de covariância do erro atual.

$$\hat{P}_k = (\bar{I} - K_k)P_k^- \quad (2.6)$$

Na equação 2.7 é apresentado o KF em tempo contínuo.

$$\frac{d}{dt}x(t) = A(t)x(t) + B(t)U(t) + w(t) \quad (2.7)$$

$$y(t) = C(t)x(t) + v(t)$$

Onde,

$w(t)$ e $v(t)$ é o ruído branco de valor médio zero e com covariância $Q(t)$ e $R(t)$ no instante t .

$A(t)$ é o modelo de transição de estado.

$B(t)$ é o modelo de controle de entrada.

$C(t)$ é o modelo de observação que transforma o espaço de estados reais no espaço de observações.

O KF trata do problema da estimação do vetor de estados em tempo discreto $x(k+1)$ de um sistema dinâmico estocástico representado no espaço a partir das medições anteriores de $u(k)$, $y(k)$, $Q(k)$, $R(k)$ e $x(k+1)$. Da mesma maneira para estimar o vetor de estado em tempo contínuo $x(t+dt)$ a partir das medições anteriores de $U(t)$, $y(t)$, $Q(t)$, $R(t)$ e $x(t)$.

No caso em que o sistema dinâmico não seja linear é utilizada uma modificação do algoritmo do filtro de Kalman denominado EKF, que lineariza o sistema em torno à pose estimada atualmente. O EKF aplicado ao problema do SLAM é feito para descrever o movimento do robô e o modelo de observação, por meio das equações 2.8 e 2.9 respectivamente, onde as matrizes $A(t)$, $B(t)$ e $C(t)$ das equações 2.1 e 2.2 são funções $f(x, U, w)$ e $h(x, v)$.

$$P(x_k|x_{k-1}, u_k) \Leftrightarrow x_k = f(x_{k-1}, U_k) + w_k \quad (2.8)$$

$$P(y_k|x_k, m) \Leftrightarrow y_k = h(x_{k-1}, U_k) + v_k \quad (2.9)$$

Onde,

$f(\cdot)$ é o modelo dinâmico do robô e w_k é ruído aditivo, se x_k é zero, significa que não há correlação entre as alterações do movimento gaussiano com a covariância Q_k .

$h(\cdot)$ descreve a geometria da observação e v_k é o ruído aditivo, se y_k é zero significa que não há correlação entre os erros de observação gaussiano com a covariância R_k .

2.3.2. FastSLAM

O FastSLAM é um algoritmo que calcula recursivamente a distribuição posterior sobre a pose do robô e a localização dos objetos. Este algoritmo foi introduzido por Koller et al. [20] [21] utilizando um filtro de partículas¹ para manter as amostras durante a trajetória do robô, sendo o primeiro em representar o modelo do processo não linear e distribuição não gaussiana da posição.

Grandes dimensões do estado-espço do problema do SLAM tornam inviáveis para as aplicações diretas do filtro de partículas. Não obstante, aplicando o filtro de Rao-Blackwellization² (R-B) é possível reduzir o número de amostras do espaço dividindo o conjunto de estados de acordo com a regra do produto $P(x_1, x_2) = P(x_2 | x_1)P(x_1)$ e se $P(x_2 | x_1)$ puder ser representada analiticamente, então somente $P(x_1)$ precisaria ser amostrado. Sendo assim, o conjunto da distribuição é representado pela equação 2.10 e as estatísticas como a marginal (equação 2.11) podem ser obtidas com maior precisão em comparação com a amostragem de todo o conjunto de estados do espaço [13].

$$\{x_1^{(i)}, P(x_2|x_1^{(i)})\}_i^N \quad (2.10)$$

$$P(x_2) \approx \frac{1}{N} \sum_i^N P(x_2|x_1^{(i)}) \quad (2.11)$$

O conjunto de estados do SLAM pode ser reduzido para uma componente de estado do robô e uma componente do mapa, para que assim o maior peso da distribuição probabilística seja sobre a trajetória e não sobre uma única posição. Deste modo, o mapa dos objetos começa a ser independente, como apresenta a Figura 2.4, onde, para cada posição do robô (círculos azuis) existem observações independentes (círculos vermelhos) gerando independência nos estados do mapa. Esta característica do FastSLAM faz com que o algoritmo seja de rápida execução.

¹ O filtro de partículas é um método que permite estimar o estado de um sistema que varia ao longo do tempo.

² O filtro R-B permite estimar o sistema principal de estados utilizando um tamanho reduzido de partículas. [37] Rapoport, I., and Oshman, Y. *Fault-Tolerant Particle Filtering by Using Interacting Multiple Model-Based Rao-Blackwellization*. Journal of Guidance, Control, and Dynamics, v. 28, n. 6, pp. 1171-1177, 2005.

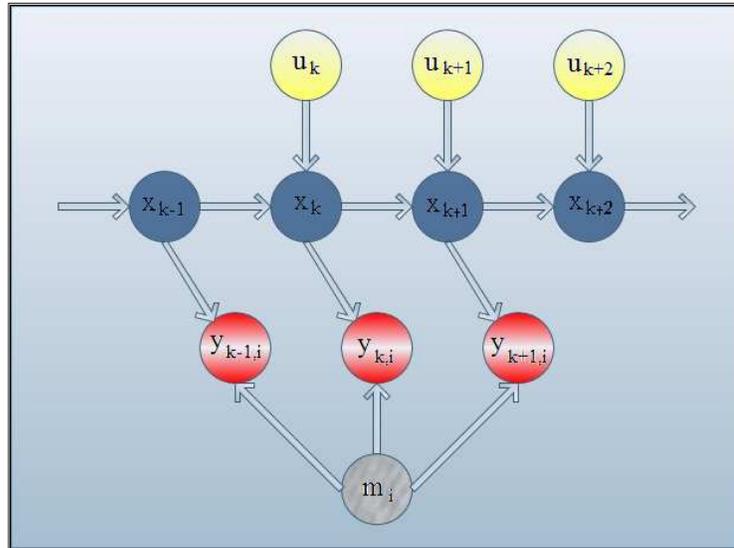


Figura 2.4 O modelo do FastSLAM, onde cada partícula define uma hipótese sobre a trajetória do robô

O mapa é representado como um conjunto de gaussianas independentes, com complexidade linear e não como um mapa de covariâncias conjuntas com complexidade quadrática. A estrutura essencial do FastSLAM utiliza o filtro de partículas para a estimação recursiva dos estados da posição e os estados do mapa são calculados analiticamente pelo EKF [13].

Desta maneira, desde o ponto de vista teórico e numérico o problema do SLAM pode ser considerado resolvido.

3. CONSTRUÇÃO DE MAPAS BIDIMENSIONAIS A PARTIR DE DUAS CÂMERAS

Neste capítulo é apresentado o modelo da câmera *pinhole* ideal utilizada para mapear os objetos de um espaço tridimensional (3-D). Utilizando duas câmeras com uma translação entre elas, pode-se obter projeções bidimensionais (2-D) a partir de uma cena 3-D. Estas projeções são refletidas em duas imagens com informação do espaço tridimensional, a fim de reconstruir os objetos pertencentes à cena e assim obter medidas de profundidade. A reconstrução de objetos é possível aplicando métodos de correspondência de pontos entre os dois planos de imagens para localizar os pontos homólogos representativos dos objetos da cena e posteriormente achar a distância dos objetos com respeito às câmeras.

3.1. Câmeras pinhole

A câmera pinhole é um modelo muito simples de uma câmera CCD (Charge Coupled Device). Este modelo é formado por um orifício e um plano de imagem, onde o orifício da câmera é a origem do sistema de coordenadas e todos os raios de projeção dos objetos no espaço convergem ao centro óptico gerando uma imagem invertida em um plano chamado de “plano de imagem”, como ilustra a Figura 3.1. A geometria da formação da imagem pode ser descrita como a projeção de cada ponto do cone no espaço através do centro óptico (orifício) para divergir a um plano [4] com distância f desde a origem de coordenadas e formar projeções em perspectivas de cada ponto dos objetos [38].

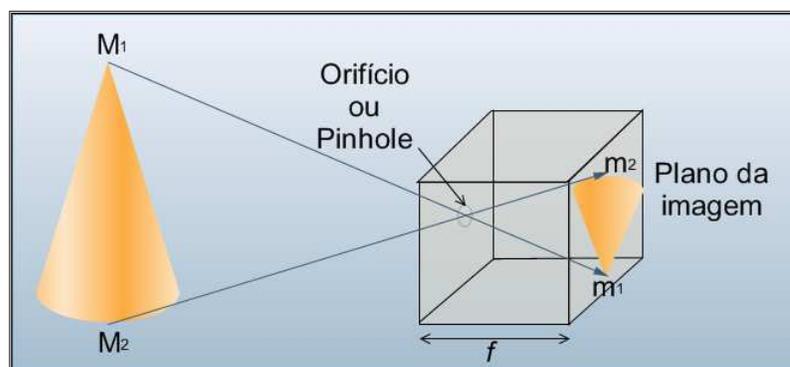


Figura 3.1 Modelo pinhole

Segundo D. Forsyth e J. Ponce [15], o plano da imagem pode ter muitos tamanhos dependendo da distância focal f , formando assim um cone de raios de luz. Os tamanhos das projeções dependem diretamente da distância focal f , o qual gera facilidade no momento de interpretar a cena, se o plano da imagem é representado à frente do centro óptico da câmera com o objetivo de não interagir com a imagem invertida e assim facilitar a interpretação da cena como é apresentada na Figura 3.2, na qual o plano está a uma distância f a frente do centro óptico.

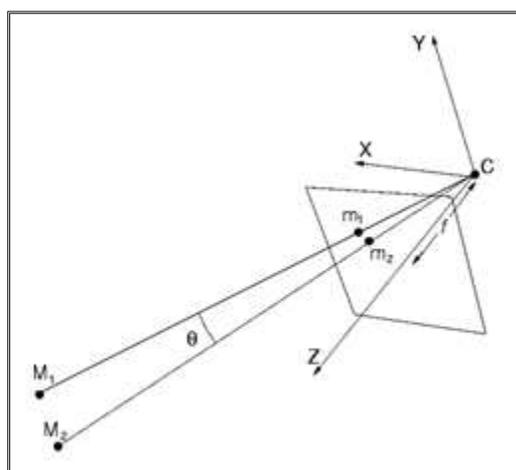


Figura 3.2 Modelo em perspectiva de dois pontos no espaço

As câmeras são representadas por modelos matemáticos que descrevem a visualização de uma cena para definir transformações que permitam interagir com os objetos no espaço através

de representações bidimensionais. Este trabalho é baseado no modelo em perspectiva definida através do modelo *pinhole*, que é descrito pela geometria projetiva.

3.2. Geometria projetiva

Sendo M um ponto no mundo representado por coordenadas $[X \ Y \ Z]^T$, a sua projeção no plano da imagem é representada por m com coordenadas $[x \ y]^T$. A posição do ponto no plano da imagem pode ser obtida pela semelhança de triângulos, como é apresentado a seguir:

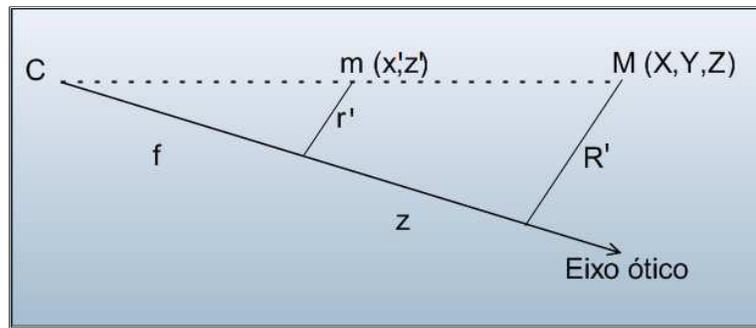


Figura 3.3 Semelhança de triângulos

Conforme a figura anterior, a semelhança de triângulos esta dada por:

$$\frac{f}{z} = \frac{r'}{R'} \quad (3.1)$$

Para estimar a profundidade dos objetos no mundo, são utilizadas duas câmeras, que permitem aproveitar o princípio da geometria epipolar³. Nesse caso, um ponto P_i posicionado no espaço com coordenadas $[X \ Y \ Z]^T$ é projetado nas duas câmeras. Na primeira câmera, o centro óptico coincide com o referencial primário XYZ e com o plano da imagem paralelo ao eixo

³ A teoria sobre a geometria epipolar está descrita por [38] Rudek, M. **Método de Posicionamento e Dimensionamento 3D Baseado em Imagens Digitais**. 2006.[Doutorado]. Universidade Estadual de Campinas, SP - BRASIL.

X e Y a uma distância focal f . A projeção desse ponto no plano da imagem é representado por p_i com coordenadas $[x_i \ y_i \ f]^T$. Através da semelhança de triângulos neste caso, se define a relação entre o ponto no espaço P_i e o ponto projetado p_i por meio da equação 3.2. O mesmo ponto P_i é projetado na segunda câmera, onde o centro óptico coincide com o referencial $X'Y'Z'$ e com o plano da imagem paralelo ao eixo X' e Y' , com a mesma distância focal f . A projeção desse ponto no plano da imagem está representado pelo ponto q_i com coordenadas $[x'_i \ y'_i \ f]^T$. A relação entre estes dois pontos é dada pela equação 3.3. A geometria de projeção de um ponto P_i nas duas câmeras é ilustrada na Figura 3.4, onde o segundo sistema de coordenadas está definido pela matriz de rotação R , e um vetor de translação t , de coordenadas $[t_x \ t_y \ t_z]^T$.

$$p_i = \frac{f}{z_i} P_i \quad (3.2)$$

$$q_i = \frac{f}{z'_i} P_i \quad (3.3)$$

As equações anteriores representam a transformação de um ponto no espaço para um ponto no plano da imagem, sendo relativas ao referencial da câmera, em coordenadas métricas.

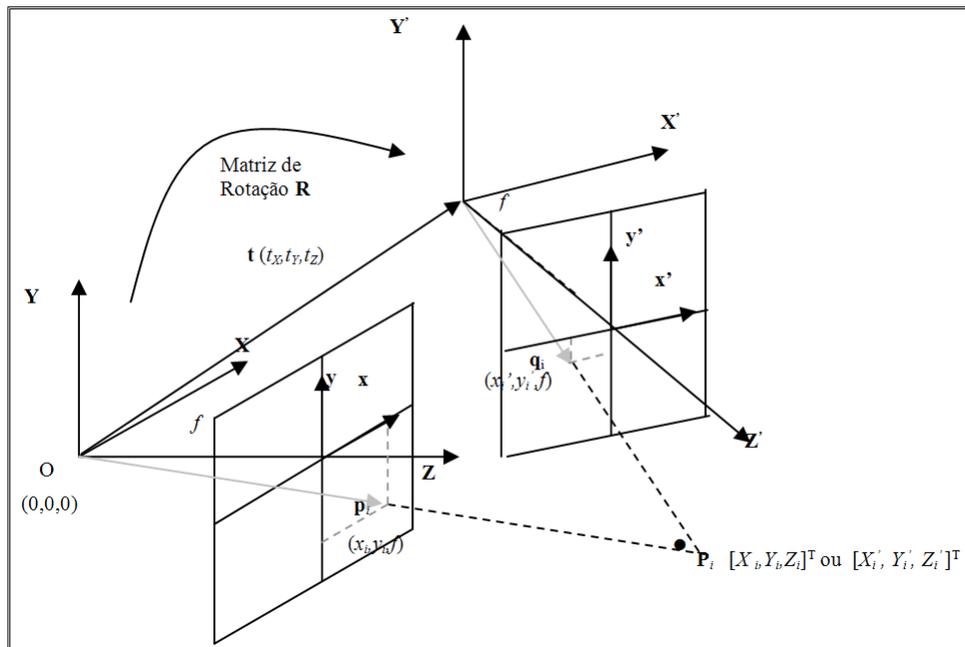


Figura 3.4 Geometria de projeção de um ponto

Conforme a figura anterior, um ponto P_i no espaço possui uma projeção p_i em uma câmera e uma projeção q_i na outra. Dessa forma um ponto da imagem de uma câmera contém somente um ponto correspondente na outra imagem. Essa correlação entre os pontos nas duas imagens dá origem ao problema de correlação de pontos.

Aproveitando o princípio da geometria epipolar, pode se escrever o referencial secundário em função do referencial primário, como ilustrado na Figura 3.4, onde se pode observar que o sistema de referência secundário fica a uma distância t do sistema de referencial primário e com uma rotação R relativa entre os dois sistemas de eixos. Portanto, a relação entre dois pontos pertencentes aos dois sistemas de referencias é dada pela equação 3.4. A partir disso, podem ser obtidos os mapas de profundidade dos pontos refletidos nas imagens, mediante uso da equação 3.5:

$$\frac{f}{z_i} p_i = \frac{f}{z'_i} R^T q_i + t \quad (3.4)$$

$$\lambda_{pi} = \frac{f}{z_i} \quad e \quad \lambda_{qi} = \frac{f}{z'_i} \quad (3.5)$$

Se as duas câmeras estão perfeitamente alinhadas como mostrado na Figura 3.5, a profundidade do ponto P para cada câmera é igual, ou seja, $Z_i = Z'_i = Z$. Para procurar esta profundidade considera-se o sistema como bidimensional, ou seja, exclui-se o eixo y do sistema para encontrar os pontos correspondentes no referencial da câmera a .

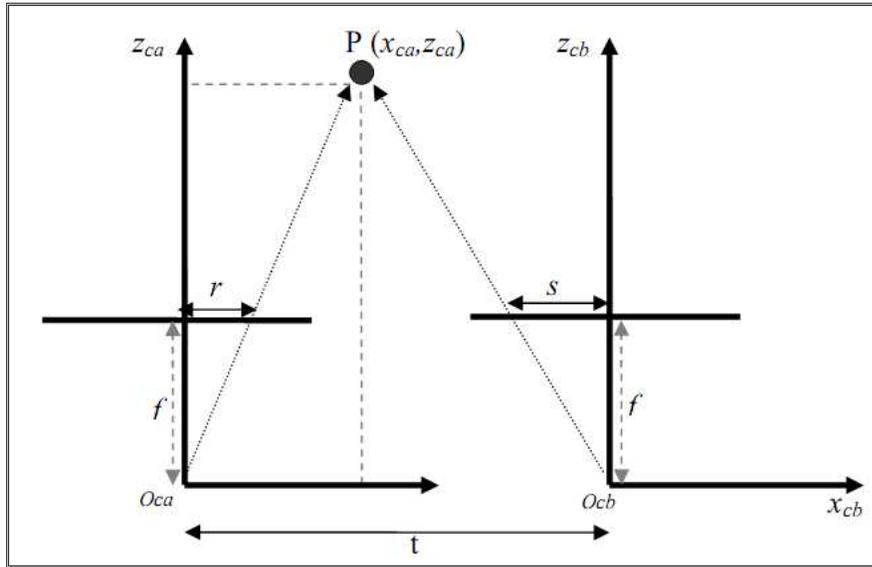


Figura 3.5 Visão estéreo com câmeras paralelas.

Neste contexto e se o ponto no espaço é refletido nas duas imagens, pode-se obter a posição do ponto com respeito ao referencial da câmera *a*. Por semelhança de triângulos tem-se:

$$\frac{x_{ca}}{z_{ca}} = \frac{r}{f} \quad (3.6)$$

$$\frac{x_{cb}}{z_{ca}} = \frac{s}{f} \quad (3.7)$$

Sabendo que *t* é a distância entre as câmeras, a posição do ponto *P* referente à câmera *a* é descrita pela equação 3.8 [10]:

$$\begin{bmatrix} x_{ca} \\ z_{ca} \end{bmatrix} = \frac{t}{(r-s)} \begin{bmatrix} r \\ f \end{bmatrix} \quad (3.8)$$

3.3. O problema da correlação estéreo

Um dos processos fundamentais na Visão Estéreo é a medição e a identificação de pontos de uma cena, a partir de duas ou mais imagens. Este procedimento é conhecido como correspondência de imagens. Para procurar pontos homólogos nas imagens é necessário ter como consideração fundamental que cada ponto de uma cena tem uma pequena alteração entre as suas posições relativas nas imagens captadas por câmeras em posições distintas [26]. Partindo desta consideração, a solução do problema de correspondência é dada pela determinação da correspondência por feições ou por áreas.

A correspondência baseada em feições procura extrair primitivas das imagens segmentadas, como por exemplo, área e raio para objetos geométricos, comprimento e orientação para objetos lineares, entre outras. Deste modo é descrita uma imagem por um gráfico com primitivas, definindo nós e relações geométricas que definem feições homologas entre as duas imagens, [45].

O exemplo deste método é dado pela Figura 3.6. Nela se apresentam duas imagens e os objetos segmentados, para se comparar cada primitiva com a feição correspondente.

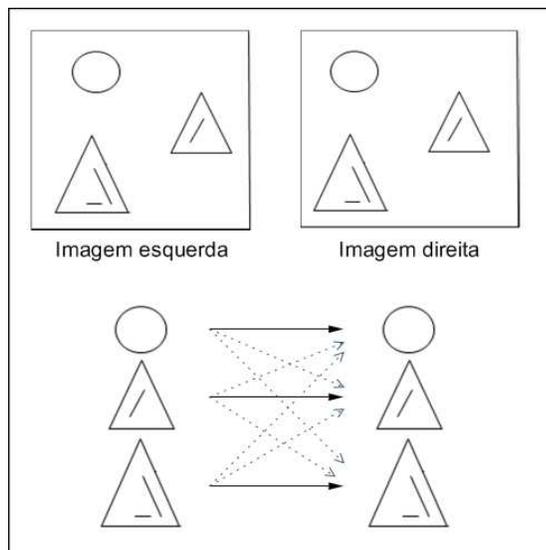


Figura 3.6 Correspondência baseada em feições

As principais vantagens desta técnica são a precisão de correspondência e o baixo tempo de processamento das imagens para se encontrar os pontos homólogos. Porém, encontrar a medida justa para todos os pontos dos objetos segmentados nas imagens representa uma desvantagem.

Por outro lado, a correspondência baseada em áreas consiste em delimitar uma região ou “janela de referência” na imagem esquerda para comparar com várias “janelas de busca” posicionadas em diferentes coordenadas da imagem da direita, como é apresentado na Figura 3.7. Esta correspondência é feita comparando os tons de cinza dos *pixels* da “janela de referência” com os tons de cinza dos *pixels* das “janelas de busca”. Desta maneira, determina-se a área onde as imagens possuem comportamentos similares. Esta técnica é realizada para todos os elementos das imagens.

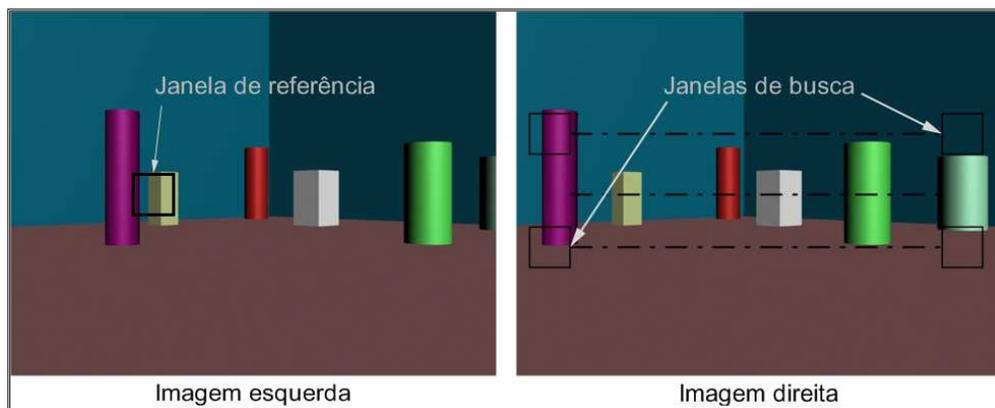


Figura 3.7 Correspondência baseada em áreas

Faz-se necessário destacar nesta técnica a possibilidade de determinar a correspondência de todos os *pixels* das imagens evitando-se o pré-processamento das imagens na extração de feições. Por outro lado, este método é vulnerável aos problemas de distorções geométricas.

Segundo Marquez F. [26], basicamente o problema de correlação estéreo possui vários fatores que prejudicam a correspondência de pontos, pode se destacar:

As distorções geométricas são causadas no momento da formação da imagem, ocorrendo de maneira relativa entre as imagens, e gerando valores de intensidades diferentes nas regiões, o que dificulta a correspondência entre essas regiões.

O ruído nas imagens está relacionado com as variações de iluminação e por características da câmera, as quais são independentes para cada imagem de câmeras independentes. O ruído é por manchas aleatórias na imagem, as quais mudam os valores de intensidade dos pixels da região a ser correspondida, o qual dificulta a correspondência de pontos entre as imagens.

Quando as texturas da superfície são transparentes ou homogêneas, ou seja, quando se tem padrões periódicos (variância nula), torna-se difícil encontrar uma correlação correta entre as imagens. Neste problema é importante haver uma boa iluminação, já que uma imagem com excesso ou falta de luz, possui baixo contraste, gerando com isso regiões com texturas homogêneas. Quando se trabalha com texturas transparentes é aconselhável a utilização de sensores ativos como o sonar ou o laser para se reconstruir estes tipos de superfícies.

Os pontos de oclusão das imagens não são considerados como problema na correlação, mas é importante considerá-los no momento de buscar pontos correspondentes. Estes pontos são aqueles pontos da cena que só são visíveis por uma das imagens, como é apresentado na Figura 3.8. Para estes pontos é impossível determinar as suas coordenadas tridimensionais, porque não podem ser reconstruídos por triangulação, portanto, é importante que sejam detectados nas imagens para assim desprezar as regiões de oclusão, evitando falsas correspondências entre as imagens.

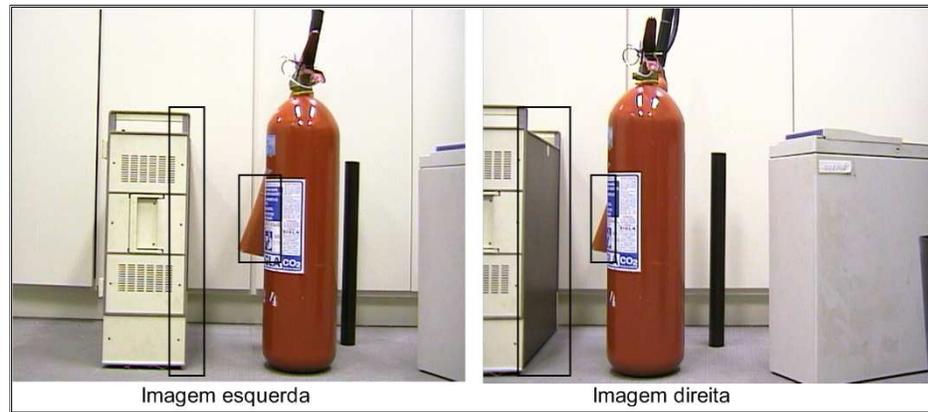


Figura 3.8 Pontos de oclusão entre duas imagens paralelas com translação de 280 mm no eixo x

3.4. Metodologia para o posicionamento do robô

Para estimar o posicionamento do robô é desenvolvida uma metodologia cujos passos são apresentados a seguir:



Figura 3.9 Metodologia de posicionamento do robô

Este trabalho está baseado nos resultados de Zhang e Longuet-Higgins [24, 45] para solucionar o problema da correlação entre duas imagens e estimar os mapas de profundidade dos

objetos inseridos no ambiente. Zhang et al., utilizam o método de correspondência por áreas empregando uma janela de referência de tamanho $(2n + 1) \times (2m + 1)$ na primeira imagem e selecionando janelas de busca na outra imagem com tamanhos de $(2d_u + 1) \times (2d_w + 1)$. A correlação dos pontos meios entre as janelas estão definidas pela seguinte equação:

$$Score(p_1, q_2) = \frac{\sum_{i=-n}^n \sum_{j=-m}^m ([I_1(k_1 + i, l_1 + j) - \overline{I_1(k_1, l_1)}] x [I_2(k_2 + i, l_2 + j) - \overline{I_2(k_2, l_2)}])}{(2n + 1)(2m + 1) \sqrt{\sigma^2(I_1) x \sigma^2(I_2)}}^{-1} \quad (3.9)$$

Onde,

k, l são as coordenadas do pixel em relação ao centro da janela

$\overline{I_1(k_1, l_1)} = \frac{\sum_{i=-n}^n \sum_{j=-m}^m I_k(k+i, l+j)}{(2n+1)(2m+1)}$ é o valor de intensidade médio do ponto (k, l) de I_k ($k=1,2$).

$$\sigma(I_k) = \sqrt{\frac{\sum_{i=-n}^n \sum_{j=-m}^m I_k^2(k, l)}{(2n+1)(2m+1)} - \overline{I_k(k, l)}}^2$$
 é o desvio padrão da imagem I_k .

3.4.1. O método de correlação de imagens unidimensional

Zhang, et al., [45] propõem um método robusto para a correlação de pontos entre duas imagens aproveitando a restrição epipolar. Com duas imagens não calibradas, ou seja, quando os parâmetros intrínsecos das câmeras e o movimento entre ambas as imagens são desconhecidos. A idéia subjacente é utilizar técnicas clássicas para encontrar pontos correlatos e depois utilizar a técnica LMedS (the Least Median of Squares) para excluir correlações falsas.

Para levar a cabo este método é necessário partir das seguintes hipóteses:

As imagens capturadas pelas câmeras devem ter exclusivamente rotações em torno do eixo perpendicular ao chão e com direção constante (ambientes planos). Além disso, todos os objetos da cena devem possuir posições fixas com relação a um referencial inercial.

Por outro lado, as câmeras são tratadas matematicamente de acordo com o modelo *pinhole* descrito anteriormente e devem ter uma distância constante entre si permanecendo paralelas.

O processo de correspondência de pontos desenvolvido consiste em seis etapas fundamentais, como é apresentado na seguinte figura:

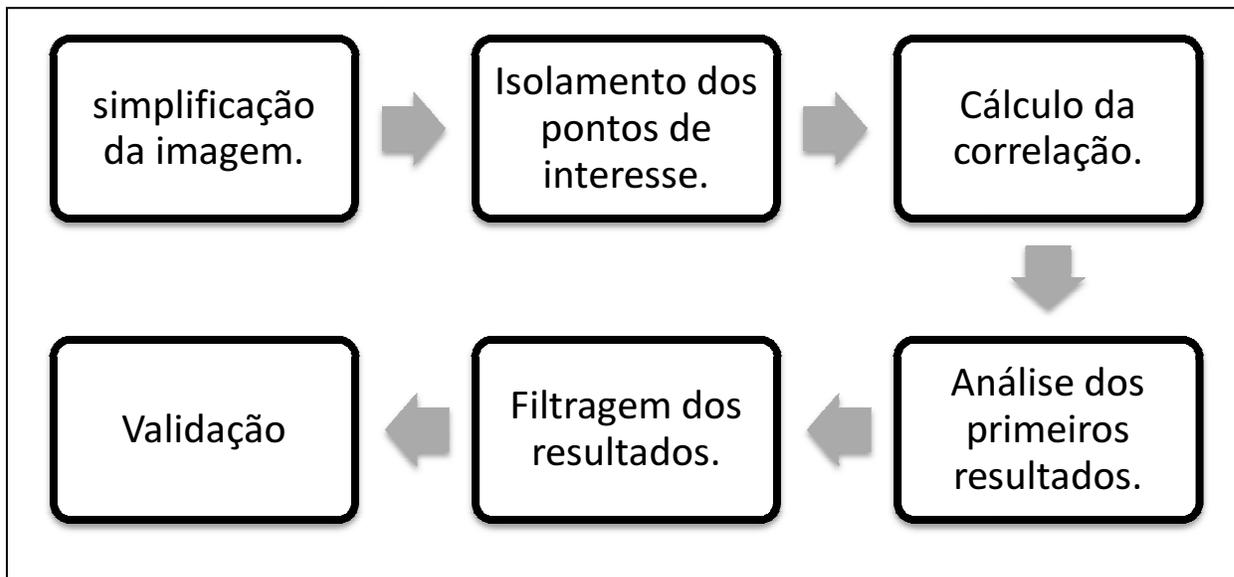


Figura 3.10 Fluxograma do processamento de imagens

A primeira etapa é a simplificação da imagem. Esta etapa consiste em extrair o vetor de *pixel* correspondente à linha horizontal média da imagem em escala de cinza para formar a imagem unidimensional e assim facilitar a análise da correspondência. Esta primeira etapa é apresentada na Figura 3.11. A Figura 3.11 (a) apresenta o estudo de caso, um ambiente simulado em Blender, com objetos de diferentes tamanhos e cores no chão e com fundo homogêneo. Na Figura 3.11 (b) é apresentado o mapa de intensidade unidimensional da linha horizontal média da imagem.

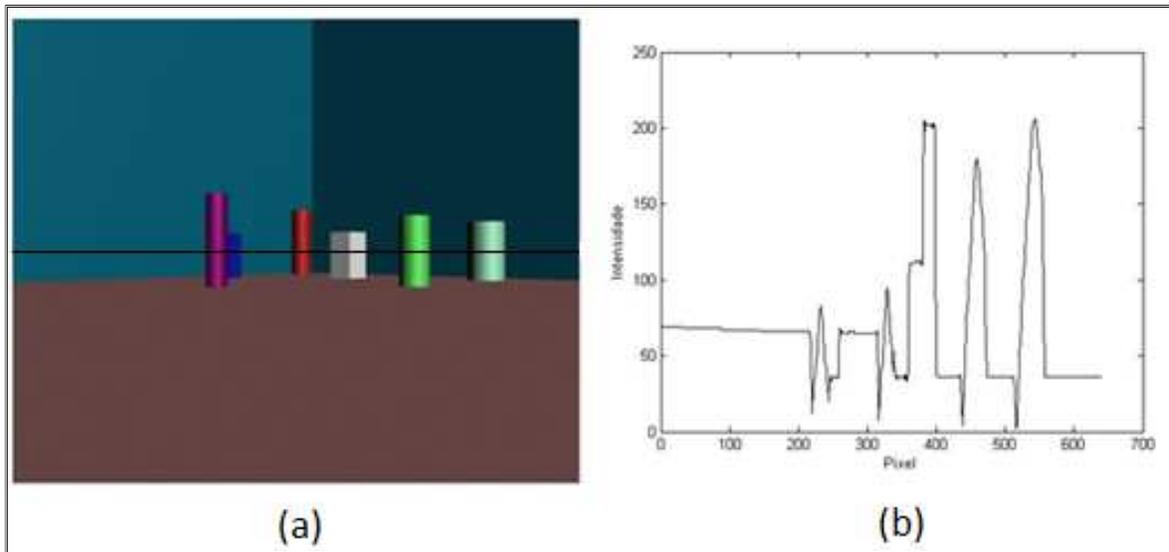


Figura 3.11 Simplificação da imagem. (a) Ambiente simulado em Blender, (b) mapa de intensidade unidimensional (Intensidade x Pixel) dos objetos

A segunda etapa consiste em isolar os pontos de interesse. Para isolar os pontos aplica-se o método *feature matching*, simplificado para o caso de uma imagem unidimensional. Este método, descrito por Z. Zhang, et al [45] consiste em extrair “características principais” inerentes nas imagens, tais como, bordas ou contornos coincidentes nas duas ou mais imagens, com intuito de direcionar pontos específicos para a correlação e assim mesmo reduzir o tempo de processamento da imagem. Aplicando esse método em imagens unidimensionais, são identificados os contornos ou as bordas dos objetos por meio de métodos de segmentação, onde são isolados os picos de derivadas apresentados na Figura 3.12 por círculos vermelhos.

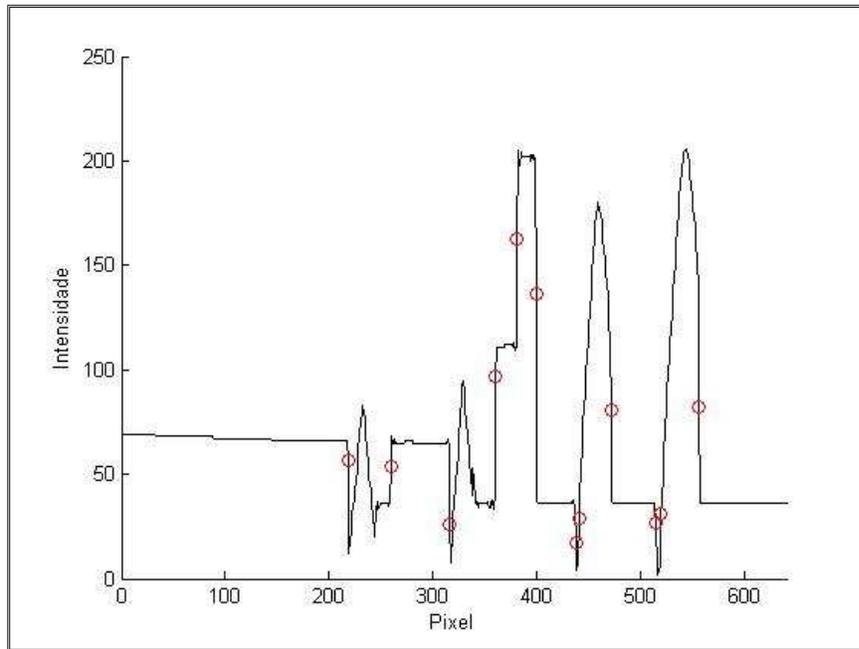


Figura 3.12 Isolamento dos pontos de interesse

A terceira etapa executa a operação de correlacionar os pontos. Sendo p_1, p_2, \dots, p_m os pontos de pico de derivada da primeira imagem e q_1, q_2, \dots, q_m os pontos de pico de derivada da segunda imagem. Isto é equivalente a reduzir os pontos de busca para os pontos correspondentes da outra imagem. O nível de correlação ou *correlation score* entre os pontos das duas imagens simplificadas está definido pela equação 3.10. Esta equação é obtida da simplificação da equação 3.9 descrita por Zhang, et al:

$$Score(p_1, q_2) = \sum_{i=-n}^n \left(\frac{[I_1(p_1+i) - \bar{I}_1(p_1)]}{\|I_1(p)\|} \times \frac{[I_2(q_1+i) - \bar{I}_2(q_1)]}{\|I_2(q_1)\|} \right) \quad (3.10)$$

Onde:

$I_k(a)$ é o valor de intensidade em escala de cinza da posição (a) da k -ésima imagem ($k=1,2$).

$\overline{I_k(a)} = \sum_{i=-n}^n \frac{I_k(a+i)}{2n+1}$ é o valor médio de intensidade em torno da posição a da k -ésima imagem.

$\|I_k(a)\| = \sqrt{\sum_{i=-n}^n I_k^2(a+i)}$ é a raiz da soma dos quadrados das intensidades dos elementos em torno da posição a da k -ésima imagem.

O valor de n é linear quando a correlação é feita em uma imagem unidimensional e este valor determina a chamada estrutura de correlação.

Na quarta etapa são analisados os primeiros resultados da correlação. Nesta etapa de análise, os níveis de correlação variam de -1 até 1, sendo -1 o valor para os pontos que não tem nenhuma similaridade e o valor de 1 para os pontos com correlações idênticas. Desta maneira são testadas todas as combinações de pontos das duas imagens. Para considerar válida a correlação de dois pontos é imposto o valor mínimo de 0,9 ao nível de correlação. Embora, para um ponto p_i da primeira imagem, geralmente exista um conjunto de pontos q_i na segunda imagem que satisfaz essa condição, denominados pontos candidatos (*candidate matches*), faz-se necessário utilizar uma filtragem nos resultados. Se representados graficamente os pontos candidatos, ou seja, o eixo y com as coordenadas dos pontos da imagem da esquerda e o eixo x com as coordenadas dos pontos da imagem da direita, como apresenta a Figura 3.13, observa-se que um nível alto de correlação não é suficiente para encontrar pontos correspondentes nas imagens. Nesta Figura pode se observar que para a borda da caixa branca da imagem da esquerda foram calculados vários pontos na outra imagem que satisfazem o umbral do cálculo de correlação. Também, para a borda do cilindro de cor magenta da imagem da esquerda é calculada um único ponto correspondente à mesma borda no cilindro da imagem da direita. Desta Figura, observa-se que os pontos correlatos têm comportamento linear, gerando uma linha diagonal de 45 graus aproximadamente.

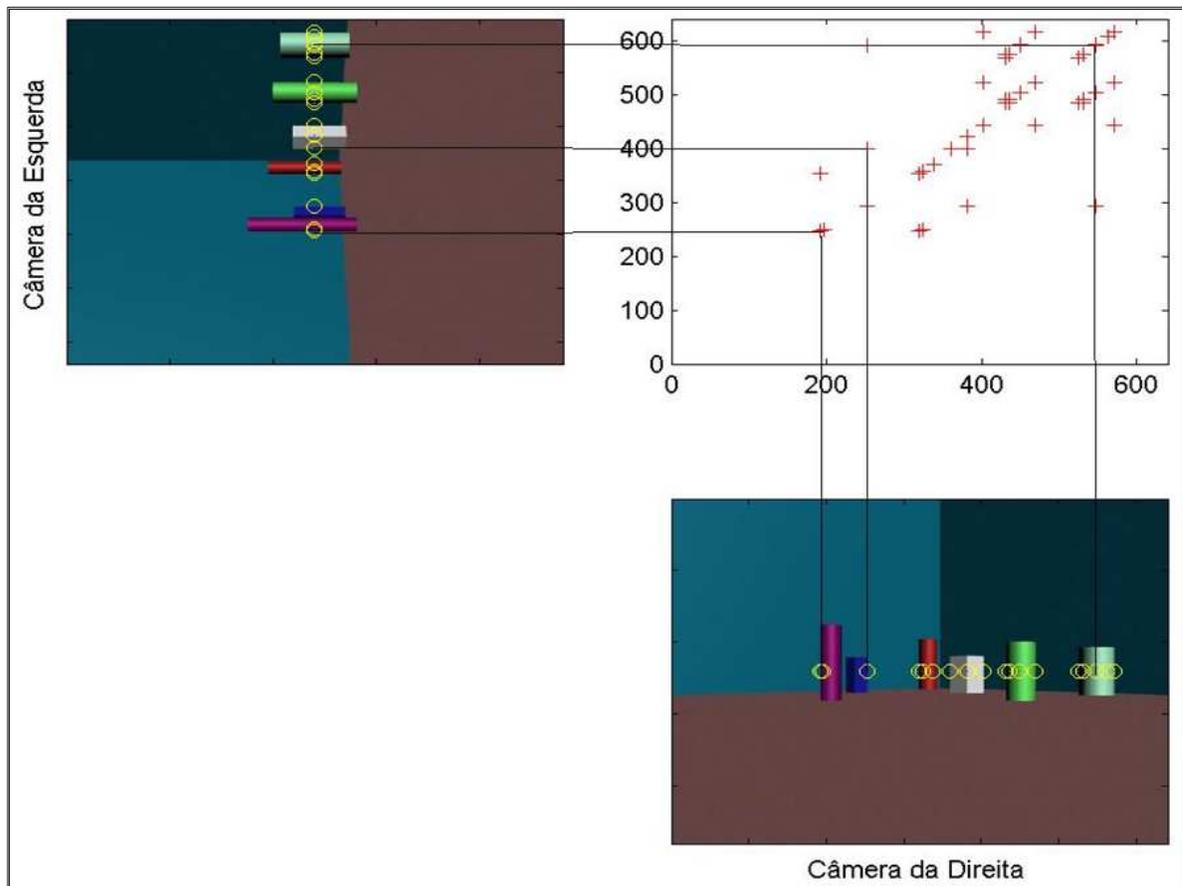


Figura 3.13 Pontos candidatos com nível 0,9

Passa-se a uma quinta etapa, encarregada de filtrar os pontos candidatos. Esta técnica busca correlacionar os pontos das imagens por meio do valor da intensidade dos *pixels*, para isso, é definido o parâmetro de intensidade H :

$$\mathbf{H} = \left| \frac{I_1(\mathbf{u}_1) - I_2(\mathbf{u}_2)}{\sqrt{I_1^2(\mathbf{u}_1) + I_2^2(\mathbf{u}_2)}} \right| \quad (3.11)$$

O parâmetro H leva em consideração o nível de intensidade dos pontos candidatos a serem correlacionados. Esse parâmetro é utilizado para avaliar todas as combinações de picos de derivada das imagens. É necessário lembrar que esse valor de intensidade foi perdido na etapa três, quando foi calculado o nível de correlação.

O valor do parâmetro H está diretamente relacionado com a correlação de intensidade dos pontos candidatos, entre mais próximo este o valor H de 1, significa que os níveis médios de intensidade serão mais semelhantes, indicando assim, que aqueles pontos são correspondidos. Esta etapa é apresentada na Figura 3.14, considerando o valor mínimo de 0,9 ao nível de correlação por intensidade.

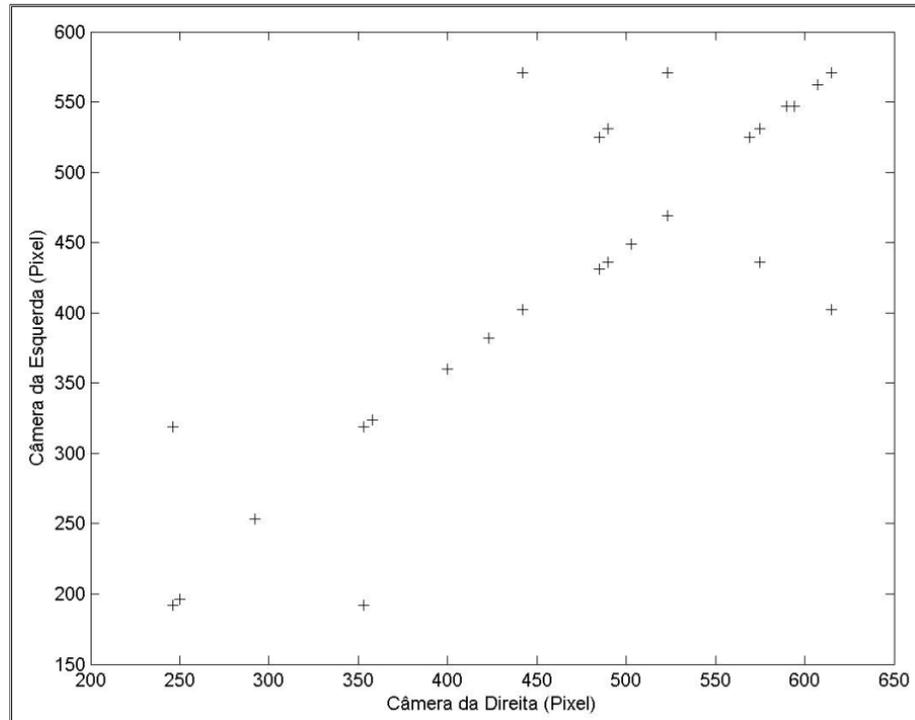


Figura 3.14 Filtragem de intensidade com valor mínimo de 0,9

Observando a figura anterior, percebe-se que apesar de haver aplicado filtragem de pontos ainda restam pontos falsos correlatos. Finalmente, para verificar a correlação dos pontos é aplicada a etapa de validação. Esta etapa faz a análise da relação epipolar, comparando o comportamento linear, como é apresentado no gráfico abaixo. Os pontos encontrados na terceira etapa são ilustrados com pontos vermelhos. O resultado da filtragem de intensidade é ilustrado com círculos azuis e finalmente os pontos validados a partir da sexta etapa são mostrados com círculos e estrelas azuis, notando-se a linearidade dos pontos.

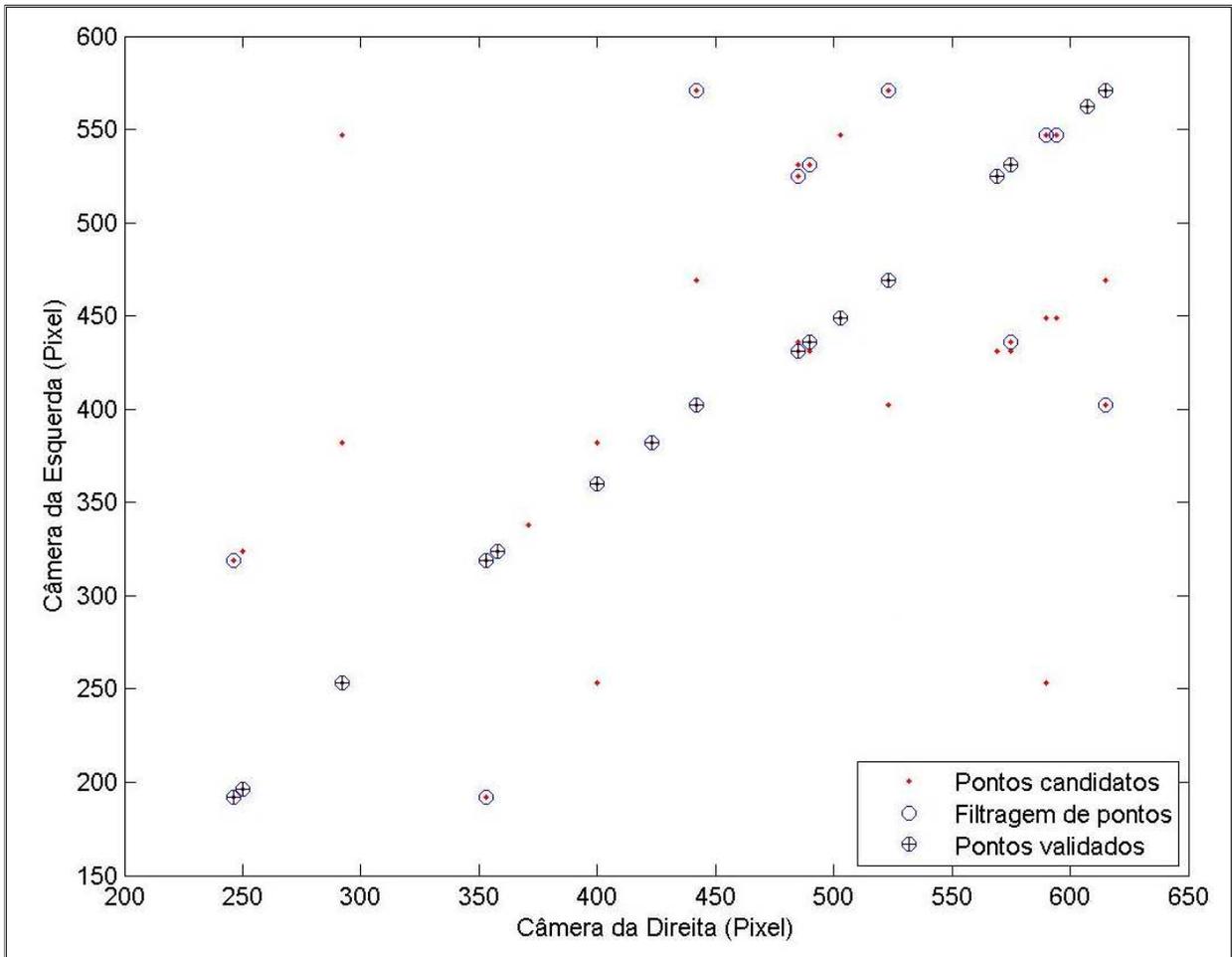


Figura 3.15 Comparação entre as etapas 4, 5 e a validação dos pontos

3.4.2. Determinação do mapa de profundidade

Encontrados os pontos correlatos entre as duas imagens utilizando o método descrito anteriormente, pode-se reconstruir o ambiente à frente do robô, por meio da relação epipolar. Isto é, dividindo a relação epipolar em suas componentes escalares X e Z , isolando os mapas de profundidade e lembrando que as câmeras são paralelas entre si. O último significa que a matriz de rotação das câmeras é a matriz identidade. Na equação 3.12 é descrita a operação de cálculo das profundidades para o i -ésimo ponto de correlação, com $p_i = x_i$ e $q_i = x'_i$:

$$\begin{bmatrix} x_i & -x'_i \\ f & -f \end{bmatrix} \begin{bmatrix} \lambda_{pi} \\ \lambda_{qi} \end{bmatrix} = \begin{bmatrix} t_x \\ t_z \end{bmatrix} \quad (3.12)$$

Dessa forma utilizando a equação 3.5 são obtidos também os valores de Z_i e Z'_i . Da mesma forma é calculada a distância lateral dos pontos de correlação em relação às câmeras (X_i e X'_i).

Sendo obtidos os valores de X_i , Z_i , X'_i e Z'_i para uma posição do robô e se essas mesmas variáveis foram encontradas depois que o robô realiza um deslocamento, é possível determinar o movimento bidimensional do robô a partir da rotação e translação comparativa nas duas cenas.

3.4.3. Estimativa de movimento bidimensional a partir de imagens estéreo

A determinação bidimensional é dada pela equação 3.13 na qual, para a posição $k=1$, são obtidas as posições $X_{k,i}$ e $Z_{k,i}$ dos pontos correlatos em relação à câmera esquerda, através dos métodos descritos anteriormente. Dessa mesma forma, depois de que as câmeras realizam um deslocamento, posição $k=2$, e se os mesmos pontos são encontrados nesta nova posição ($X_{k+1,i}$ e $Z_{k+1,i}$), pode ser utilizada a relação epipolar entre as duas posições para estimar a orientação e translação entre os mapas de profundidade:

$$\begin{bmatrix} X_{k,i} \\ Z_{k,i} \end{bmatrix} = R^T \begin{bmatrix} X_{k+1,i} \\ Z_{k+1,i} \end{bmatrix} + t \quad (3.13)$$

$$R = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

Onde k indica a posição do robô e i indica o par de pontos que foram correlacionados. Por convêniencia a rotação e translação devem ser pequenas, de maneira que os pontos correlacionados na primeira posição sejam encontrados na segunda posição.

Substituindo a matriz de rotação R em função de seno e cosseno do ângulo, já que o ângulo θ é incógnita e substituindo o vetor de translação t em termos de suas componentes escalares, obtem se:

$$\begin{aligned}
X_{k,i} &= (X_{k+1,i})(\text{Cos}(\theta)) - (Z_{k+1,i})(\text{Sin}(\theta)) - t_x \\
Z_{k,i} &= (X_{k+1,i})(\text{Sin}(\theta)) + (Z_{k+1,i})(\text{Cos}(\theta)) - t_z
\end{aligned}
\tag{3.14}$$

Das equações anteriores note que tem-se 2 equações e 3 incógnitas (θ , t_x e t_z). Sendo necessário corresponder pelo menos 2 pontos para resolver o sistema de equações da forma $Ax=b$, descrito a seguir.

$$\begin{bmatrix}
X_{k+1,i} & -Z_{k+1,i} & 1 & 0 \\
Z_{k+1,i} & X_{k+1,i} & 0 & 1 \\
\vdots & \vdots & \vdots & \vdots \\
X_{k+N,i} & -Z_{k+N,i} & 1 & 0 \\
Z_{k+N,i} & X_{k+N,i} & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\text{Cos}(\theta) \\
\text{Sen}(\theta) \\
t_x \\
t_z
\end{bmatrix}
=
\begin{bmatrix}
X_{k,i} \\
Z_{k,i} \\
\vdots \\
X_{N,i} \\
Z_{N,i}
\end{bmatrix}
\tag{3.15}$$

A equação anterior descreve o sistema linear com duas equações para cada ponto e quatro incógnitas. Para resolver este sistema linear devem ser correspondidos pelo menos 2 pontos, mas na realidade, são correlatos mais de 2 pontos nas imagens, fazendo com que o sistema possua mais equações que incógnitas. Dessa maneira, uma solução do sistema é dada pelo método dos mínimos quadrados, descrito pela equação 3.16.

$$\hat{S} = (A^T A)^{-1} A^T b
\tag{3.16}$$

O vetor \hat{S} contém a solução ao sistema, composto pelo ângulo θ (descrito pela equação 3.17) e o vetor de translação t , determinando o movimento realizado da posição k até $k+1$.

$$\theta = \text{atan} \left(\frac{-\text{Sin}(\theta)}{\text{Cos}(\theta)} \right)
\tag{3.17}$$

4. Proposta de uso de visão para navegação robótica em ambientes virtuais

Este capítulo é dedicado à apresentação dos estudos de caso em simulações de ambientes virtuais dos diferentes métodos descritos no capítulo anterior. Na atualidade não existe informação que permita validar os diversos algoritmos descritos, sendo necessário analisar o comportamento dos métodos propostos para a navegação de robôs autônomos utilizando imagens provenientes de câmeras virtuais, para se obter informação do ambiente à frente do robô e conseguir estimar mapas de profundidade dos objetos, movimento e posição do robô. As simulações dos diferentes estudos de caso são efetuadas na plataforma BLENDER [25], para a obtenção de imagem pelos sistemas de câmeras e projeções de objetos simulados em ambientes virtuais, e da plataforma MATLAB [27] como ferramenta matemática, com o fim de se obter informação que permita validar os diferentes métodos.

O Blender é um software de simulação de ambientes virtuais de código aberto, otimizado para a produção de gráficos 3D. O Blender fornece um amplo espectro de modelagem, texturização, iluminação, animação, renderização⁴ e pós-processamento de vídeo. Este software permite modelar e construir ambientes virtuais controlados, ou seja, todas as coordenadas (absoluta ou relativa), tamanhos e formas dos diferentes objetos inseridos no ambiente virtual são conhecidas. Isso, permite posicionar o robô e os objetos de forma semelhante aos ambientes reais, gerando cenários que permitam testar os métodos descritos no capítulo 3. A vantagem de utilizar ambientes controlados é a obtenção de dados reais que permitam comparar e avaliar a estimativa de profundidade dos objetos e a estimação de movimento.

⁴ Renderizar é o processo de gerar uma imagem 2D a partir de uma cena 3D, simulando condições reais.

Para a obtenção das imagens, o Blender permite simular câmeras ideais, segundo o modelo *pinhole* descrito, com a opção de ajustar o foco, orientação, número de pixel e sua posição, dentre outras.

4.1. Estrutura do ambiente virtual

O ambiente virtual criado com o auxílio da plataforma Blender teve em conta as seguintes variáveis:

- Descrição de um ambiente interno.
- Objetos fixos.
- Chão plano.
- Fundo homogêneo.
- Iluminação constante e controlada.
- Duas câmeras ideais e paralelas.

Na Figura 4.1 são apresentadas imagens captadas de duas câmeras virtuais.

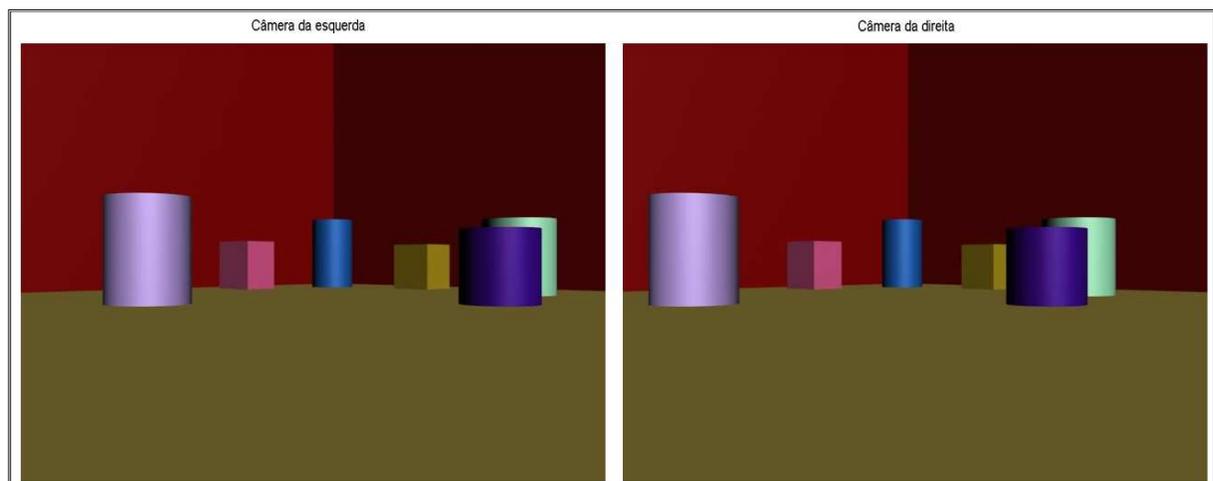


Figura 4.1 Imagens obtidas a partir de duas câmeras virtuais

4.1.1. Metodologia da simulação em ambientes virtuais

São propostos quatro ambientes virtuais para analisar o comportamento dos diferentes métodos. O objetivo é estimar o movimento do robô entre duas posições, utilizando duas câmeras como únicos sensores de movimento. Assim, pode-se obter um padrão comparativo entre os dados reais e os dados calculados. Para estimar o movimento entre as duas posições, posiciona-se inicialmente o robô em uma coordenada conhecida e se adquire e armazena duas imagens nessa posição. Em seguida o robô é posicionado em uma segunda coordenada e se obtém duas imagens nessa nova posição. Após se obter as quatro imagens, aplica-se o método de correlação de imagens unidimensionais na busca de pontos correlatos nas quatro imagens do ambiente à frente do robô, gerando-se mapas de profundidades associados a cada posição. Com estas informações calcula-se o movimento do robô entre as duas posições.

4.2. Simulação de navegação robótica em ambientes virtuais

Nos quatro ambientes simulados são utilizadas duas câmeras sobre um robô, um conjunto de objetos de diferentes cores e um sistema de referência absoluto. Os ambientes simulados variam no número dos objetos, cor do fundo das paredes e posicionamento do robô. Na figura 4.2 é mostrada uma vista superior do ambiente, onde se observam os objetos, o referencial absoluto e o robô. O robô é representado por uma caixa preta com duas câmeras. As coordenadas utilizadas nos ambientes são descritas desde a coordenada (0,0) do referencial absoluto até o centro dos objetos.

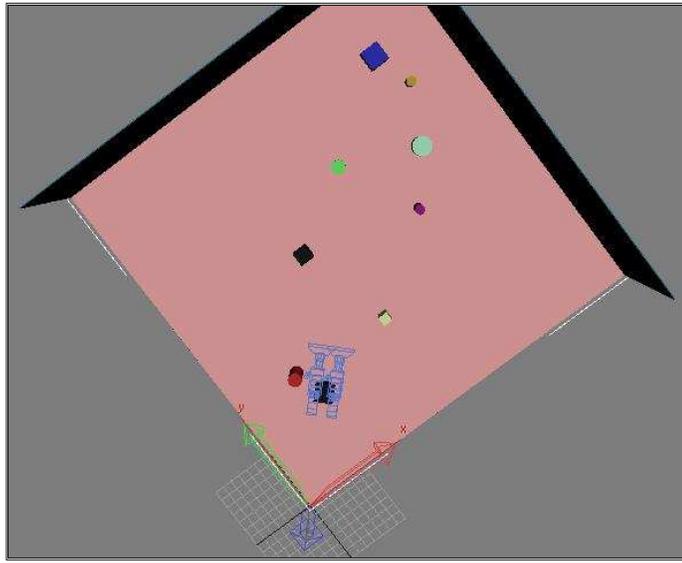


Figura 4.2 Vista superior de um ambiente simulado

Simulação ambiente I: O primeiro ambiente é criado por duas paredes de cor azul e oito objetos com coordenadas cujas posições são descritas pela Tabela 4.1. Os objetos estão distribuídos à frente do robô com diferentes profundidades. O posicionamento da câmera esquerda do robô está descrita na Tabela 4.2. No primeiro teste o robô é posicionado conforme mostrado na Figura 4.3 (a) onde se obtém o primeiro par de imagens. O robô é em seguida deslocado em linha reta para uma distância de 141.42 mm da primeira posição, obtendo-se o par de imagens da segunda posição, conforme a Figura 4.3 (b).

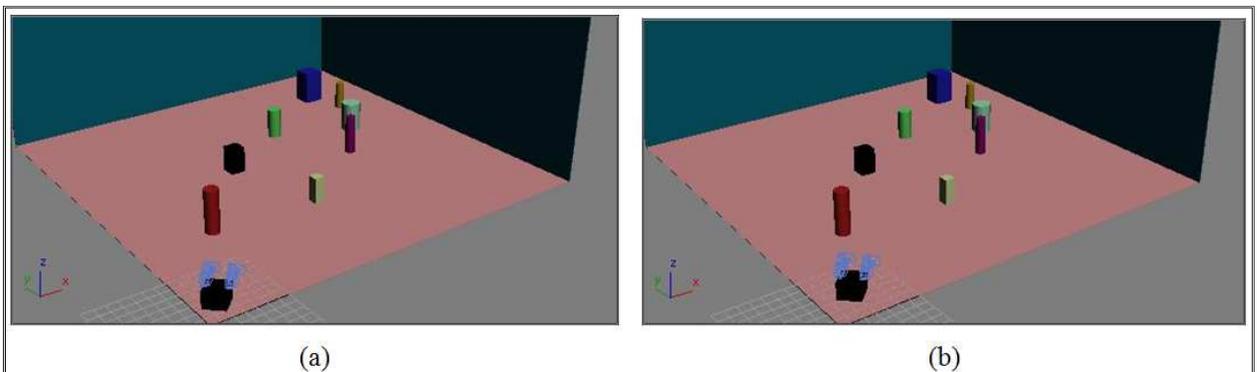


Figura 4.3 Simulação do ambiente I: (a) Primeira posição do robô, (b) Segunda posição do robô

Tabela 4.1 Coordenada dos objetos do ambiente I

Objetos	Coordenada em X (mm)	Coordenada em Y (mm)
Vermelho	730	1200
Preto	1530	2130
Verde claro	1800	1130
Verde	2340	2620
Roxo	2730	1810
Verde água	3120	2290
Azul	3270	3270
Mostarda	3410	2860

Tabela 4.2 Coordenada da câmera esquerda até o referencial absoluto do ambiente I

Posição	Coordenada em X (mm)	Coordenada em Y (mm)
1	193.72	370.71
2	293.72	470.71

Simulação ambiente II: Para o segundo caso são utilizados cinco objetos posicionados perto de uma parede branca, as profundidades entre os objetos são relativamente pequenas, como ilustra a Tabela 4.3. Este ambiente busca comparar os resultados obtidos quando o robô está longe e quando o robô está perto dos objetos, para isto, são apresentadas duas simulações em paralelo. A primeira simulação é feita posicionando o robô longe dos objetos, como ilustra a Figura 4.4 (a). Na segunda simulação é posicionado o robô perto dos objetos, conforme ilustrado na Figura 4.4 (b). As posições do robô em ambos os casos são mostrados na Tabela 4.4, onde L descreve a posição “longe” e P a posição “perto” dos objetos. A partir das posições L1 e P1 do robô, são obtidos pares de imagens conforme a tabela.

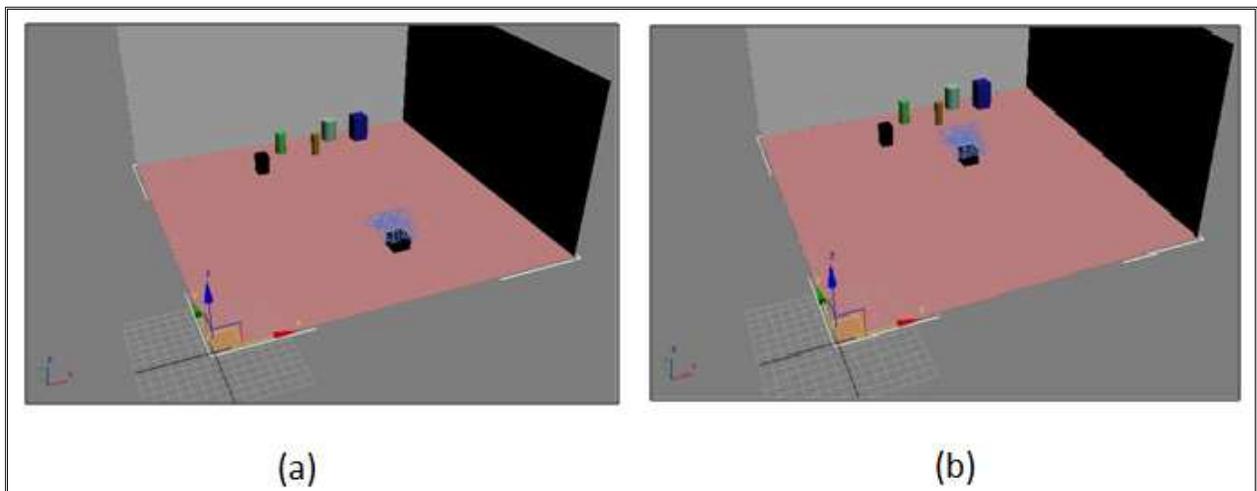


Figure 4.4 Simulação do ambiente II: (a) Posição do robô longe dos objetos, (b) Posição do robô perto dos objetos

Tabela 4.3 Coordenada dos objetos do ambiente II

Objetos	Coordenada em X (mm)	Coordenada em Y (mm)
Preto	1500	3190
Verde	1920	3660
Mostarda	2340	3430
Verde claro	2700	3810
Azul	3100	3680

Tabela 4.4 Coordenada da câmera esquerda

Posição	Coordenada em X (mm)	Coordenada em Y (mm)
L1	2224.85	954.85
L2	2224.85	1054.85
L3	2224.85	1154.85
P1	2224.85	2474.85
P2	2224.85	2574.85
P3	2224.85	2674.85

Simulação ambiente III: Na simulação anterior foram obtidos pares de imagens cada 100 mm a partir de uma posição distante e outra perto dos objetos. O terceiro ambiente está baseado no mesmo cenário. Os pares de imagens são obtidos em L1 e L3; e P1 e P3, conforme descritos na Tabela 4.4. O objetivo principal deste estudo é analisar o desempenho dos métodos de

estimação de deslocamento quando um movimento retilíneo de 200 mm é realizado na direção dos objetos.

Simulação ambiente IV: Este ambiente pretende analisar a performance dos métodos de estimação de movimento durante o percurso do robô. O ambiente simulado é composto de duas paredes azuis de fundo e seis objetos. A coordenada de posição dos objetos é descrita pela Tabela 4.5. Os objetos estão distribuídos à frente da primeira posição do robô, com diferentes profundidades conforme ilustrado na Figura 4.5. O posicionamento da câmera esquerda, que registra a trajetória do robô está descrita na Tabela 4.6.

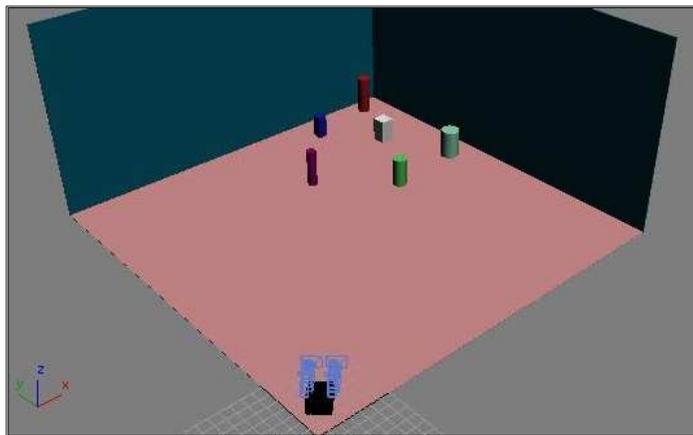


Figura 4.5 Simulação do ambiente IV

Tabela 4.5 Coordenada dos objetos do ambiente IV

Objetos	Coordenada em X (mm)	Coordenada em Y (mm)
Roxo	2170	2860
Verde	2850	2250
Azul	2890	3640
Branco	3370	3100
Vermelho	3680	3800
Verde água	3680	2350

Tabela 4.6 Trajetória do robô em direção aos objetos

Posição	Coordenada em X (mm)	Coordenada em Y (mm)
1	41.25	56.88
2	62.25	76.88
3	83.25	97.88
4	102.25	115.88
5	122.25	136.88
6	144.25	156.88
7	164.25	178.88
8	184.25	199.88
9	204.25	221.88
10	224.25	241.88
11	248.25	265.88
12	270.25	286.88
13	288.25	308.88

4.3. Análise e resultados

A seguir são apresentados os resultados de cada ambiente simulado:

Ambiente I:

Após se obter as quatro imagens, é aplicado o método de correlação de imagens para extrair informação útil do ambiente. Para se otimizar o tempo de processamento dos dados, correlacionam-se primeiro as imagens da câmera esquerda na posição anterior (CE_PAN) e da câmera esquerda na posição atual (CE_PAT), chamando-se “Correlação I” a esse processo. Os pontos correlatos entre estas duas imagens são procurados na imagem da câmera direita na posição anterior (CE_PAN), chamado a esse processo “Correlação II”. Finalmente é realizado o último cálculo de correlação entre os pontos obtidos entre as três imagens descritas anteriormente e a imagem “CD_PAT”, “Correlação III”. O método de correlação de imagens unidimensional descrito passo a passo anteriormente é apresentado na seguinte Figura 4.6 abaixo.

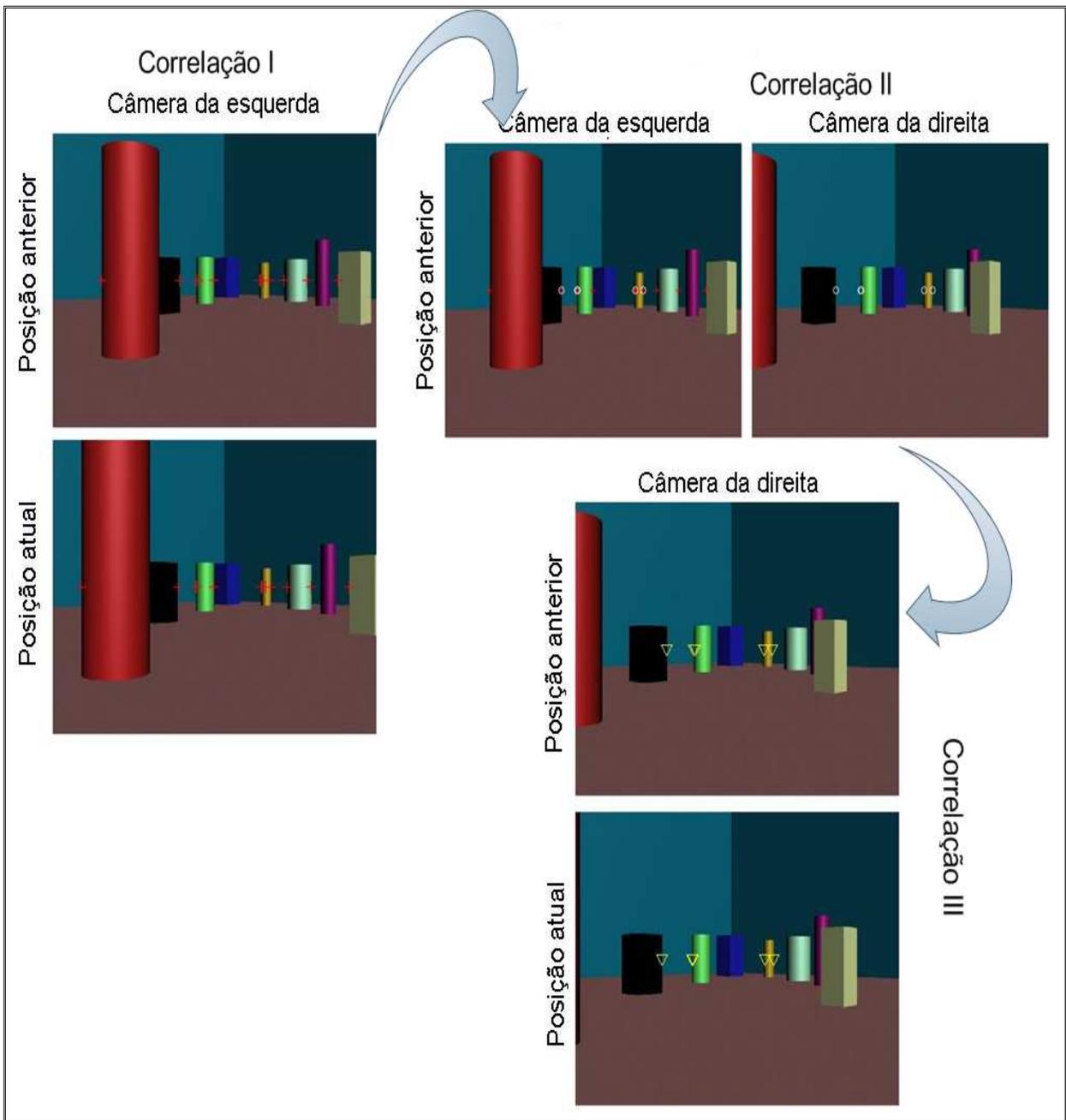


Figura 4.6 Descrição da correlação: Correlação I, correlação entre as imagens “CE_PAN” e “CE_PAT”. Correlação II, correlação entre os pontos obtidos em Correlação I e a imagem “CD_PAN”. Correlação III, correlação entre os pontos obtidos em Correlação II e a imagem “CD_PAT”

Observa-se que na “correlação I” uma quantidade razoável de pontos é inicialmente selecionada através do método de segmentação, conforme a Figura 4.7. Na “correlação II” a quantidade de pontos diminuiu, devido à existência de bordas não visíveis na imagem “CD_PAN”. Nesta imagem também pode-se apreciar que alguns objetos estão muito juntos, apresentando comportamentos diferentes nos pixel vizinhos entre algumas bordas, o que não leva a uma indicação positiva de correlação. Este comportamento é observado na imagem “CD_PAN”, entre os objetos de cor verde claro, roxo e verde água.

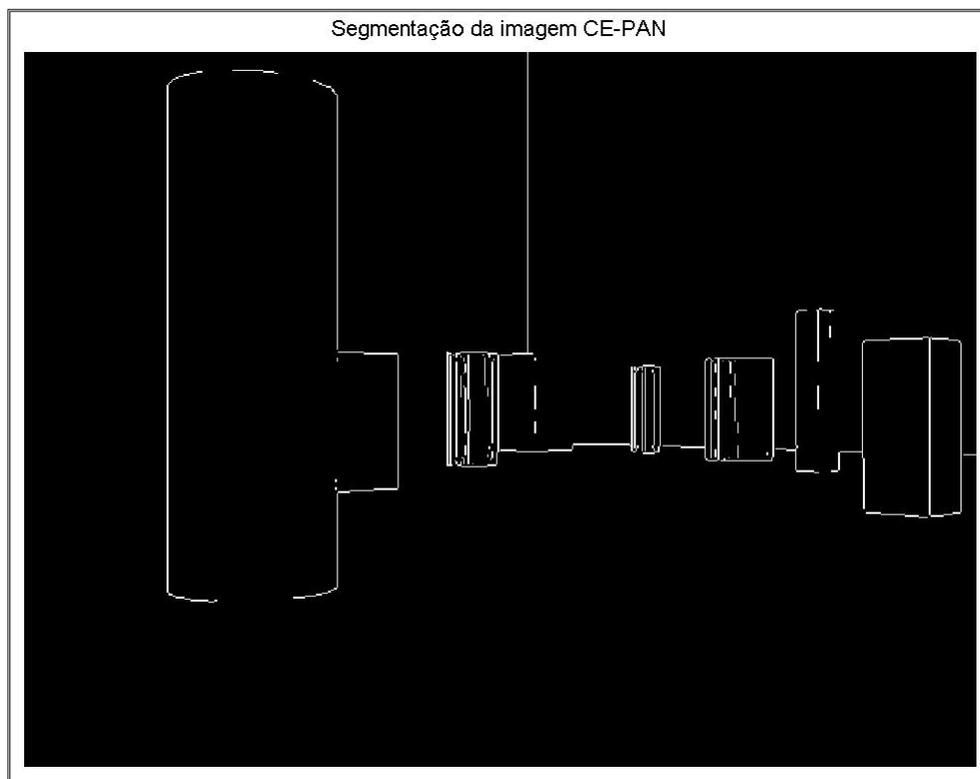


Figura 4.7 Segmentação da imagem CE_PAN

Na Figura 4.8 apresenta-se o resultado das correlações entre pontos visíveis nas quatro imagens, lembrando que o objetivo principal das correlações é determinar as suas coordenadas de pixels, para assim utilizar a relação epipolar descrita no capítulo 3.

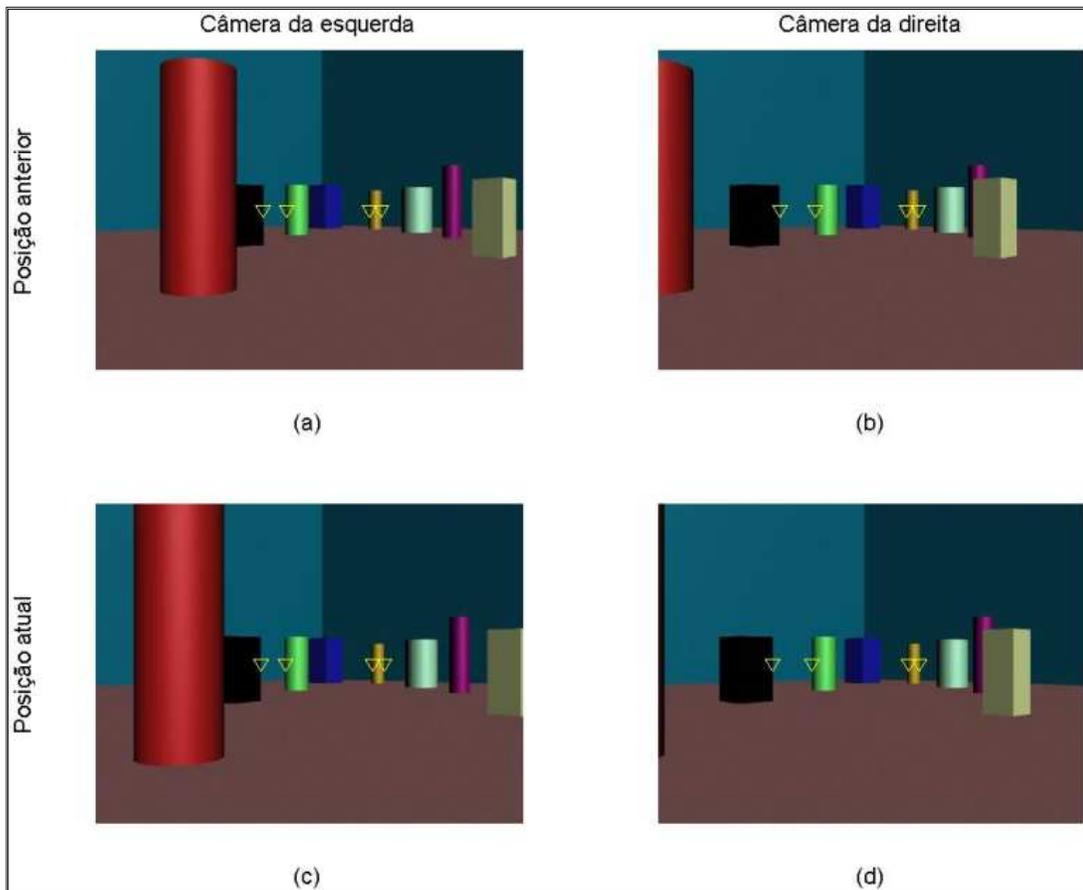


Figura 4.8 Correlação de pontos da simulação I: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

A partir das correlações de pontos são estimados os valores de distância entre a câmera esquerda e as bordas correlatas, gerando mapas de profundidade de cada posição. A figura 4.9 ilustra a reconstrução do ambiente visto pelo robô em cada posição. As coordenadas dos objetos são plotadas a partir do referencial relativo.

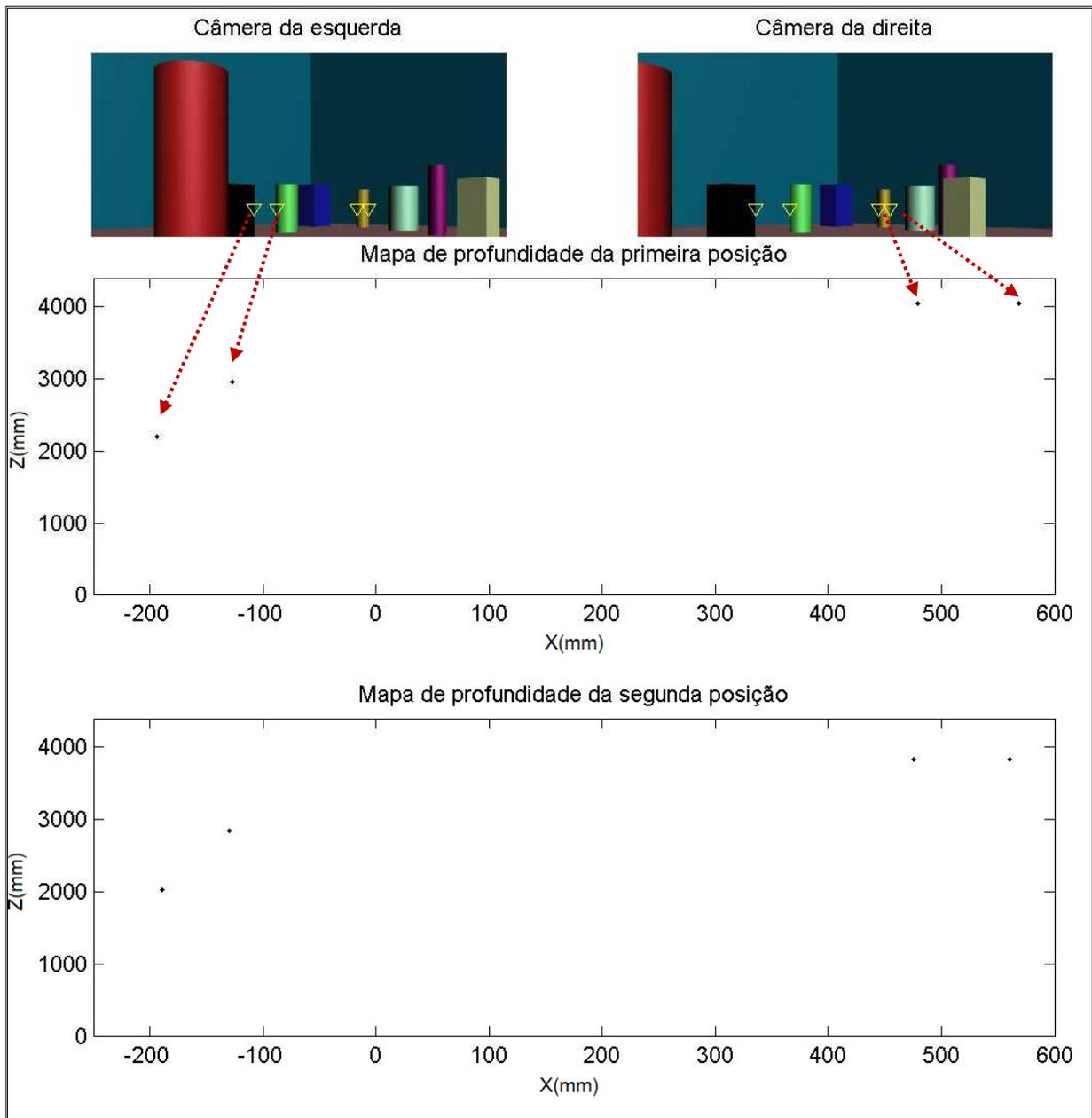


Figura 4.9 Mapa de profundidade de cada posição, em vista superior. A câmera esquerda é o referencial relativo (0,0) do sistema de coordenadas

A comparação entre os dados do ambiente controlado descrito para esta simulação e os dados obtidos pelo método de estimação de profundidade, são apresentados na Tabela 4.7. A coluna “Objeto” desta tabela representa as cores das bordas dos objetos correlacionados anteriormente, sendo apresentada da esquerda para a direita.

Tabela 4.7 Comparação de profundidade do primeiro ambiente

Posição	Objeto	Distância pelo aplicativo Blender (mm)	Distância calculada (mm)	Erro (%)
I	Preto	2195.06	2202.87	0.35
I	Verde	3092.33	2956.57	4.39
I	Mostarda	4077.47	4070.38	0.17
I	Mostarda	4077.98	4081.87	0.09
II	Preto	2062.84	2016.94	2.22
II	Verde	2963.84	2854.24	3.69
II	Mostarda	3923.37	3885.07	0.97
II	Mostarda	3939.39	3885.88	1.35

Analisando-se a média dos valores dos erros para cada posição tem-se 1.25% para o primeiro mapa e 2.05% para o segundo mapa. Considerando que a distância entre as câmeras na simulação é de 200 mm, distância relativamente pequena em comparação com o ambiente em que o robô navega, conclui-se que os mapas de profundidade gerados correspondem com as medidas do ambiente simulado.

Com os mapas de profundidade obtidos, estima-se a rotação e translação que ocorre entre as duas posições, usando a relação epipolar descrita pela equação 3.13 do capítulo anterior. Tabela 4.8 mostra os resultados obtidos do movimento realizado entre os dois pares de imagem.

Tabela 4.8 Estimativa de movimento no ambiente I

	Movimento no aplicativo Blender	Movimento calculados	Erro (%)
Translação em X (mm)	100	104.28	4.28
Translação em Z (mm)	100	108.51	8.51
Rotação em Y (Graus)	45	45.45	1

Os erros mostrados na tabela anterior ocorrem porque a relação epipolar estabelece que se um mesmo ponto é visto em duas posições, a translação e rotação entre as essas posições podem ser calculadas, mas na realidade um ponto da imagem “CE_PAN” pode ser correspondido com o pixel vizinho da imagem “CD_PAN”, gerando um pequeno erro de profundidade. Também pode se observar que os mapas de profundidade calculados anteriormente para as duas posições,

apresentam erros. Isto significa que os pontos utilizados para o cálculo da estimativa de movimento possuem incerteza devido a erros constantes no transcurso dos métodos. Por outro lado, os erros de estimativa da translação e rotação do robô, são pequenos quando comparados com as dimensões do ambiente e robô.

Ambiente II:

Os pontos correlatos entre os pares de imagens obtidos na posição L1 e L2 são apresentados na Figura 4.10. Os pontos correlatos entre os pares de imagens das posições P1 e P2 são apresentados na Figura 4.11.

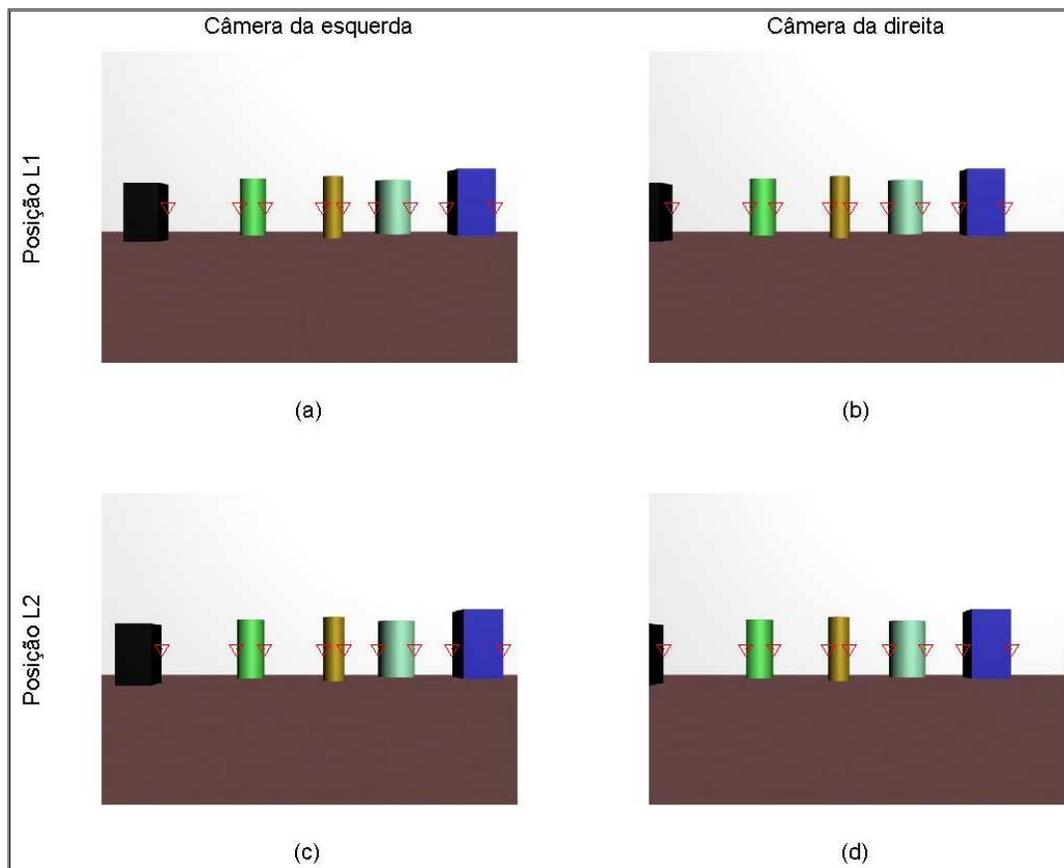


Figura 4.10 Correlação de pontos da simulação II, entre L1 e L2: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

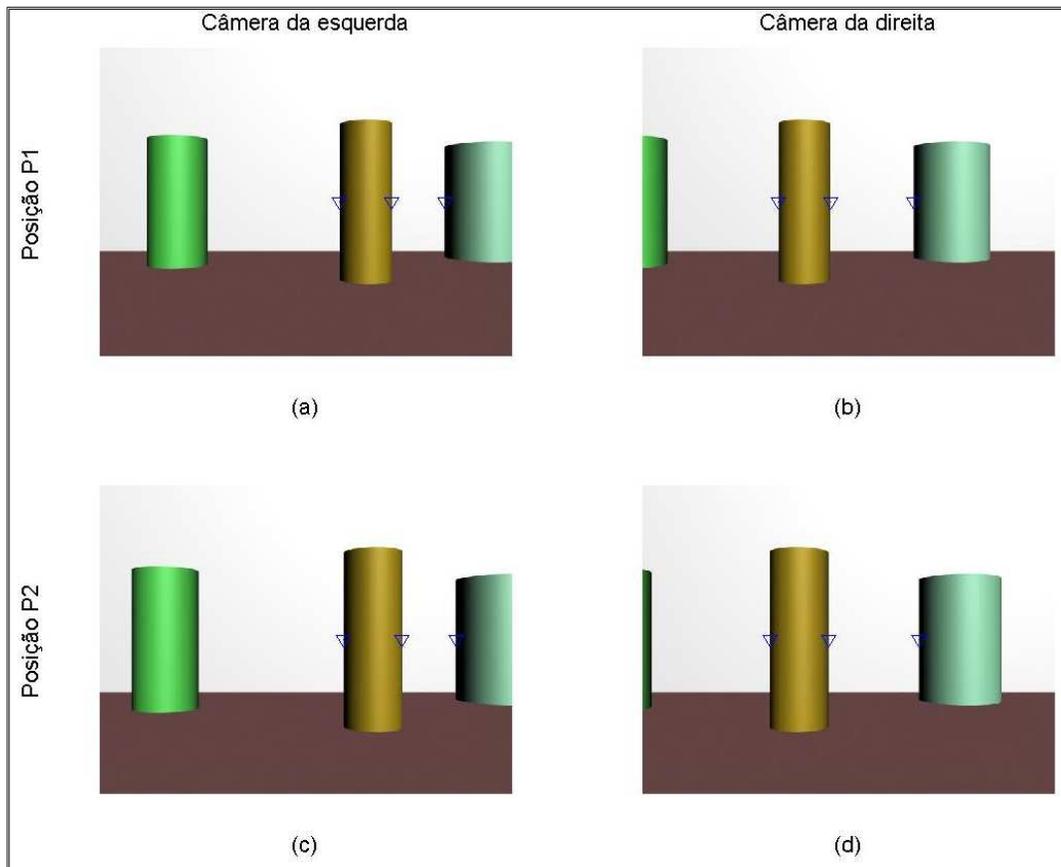


Figura 4.11 Correlação de pontos da simulação II, entre P1 e P2: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

A partir das correlações de pontos estimam-se os valores de distância entre a câmera da esquerda e as bordas correlatas, gerando mapas de profundidade de cada posição. A figura 4.12 ilustra a reconstrução do ambiente visto pelo robô na posição L1 e L2. A figura 4.13 ilustra a reconstrução do ambiente visto pelo robô na posição P1 e P2.

Os dados medidos no programa Blender são comparados na Tabela 4.9 e 4.10, com as profundidades estimadas na posição L1 e L2, e na Tabela 4.11 e 4.12, com as profundidades estimadas na posição P1 e P2.

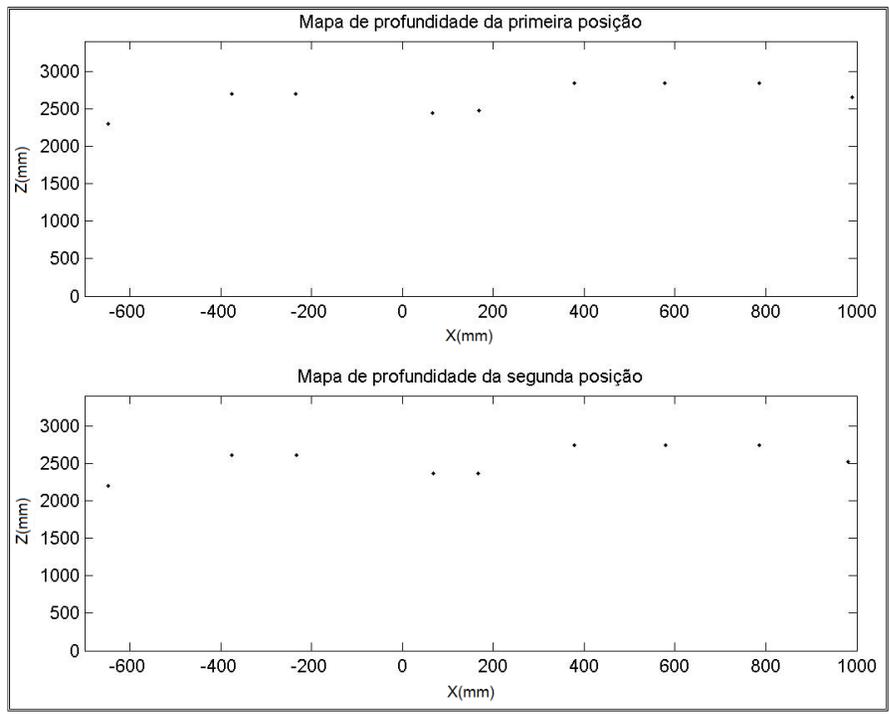


Figura 4.12 Mapas de profundidade das posições L1 e L2

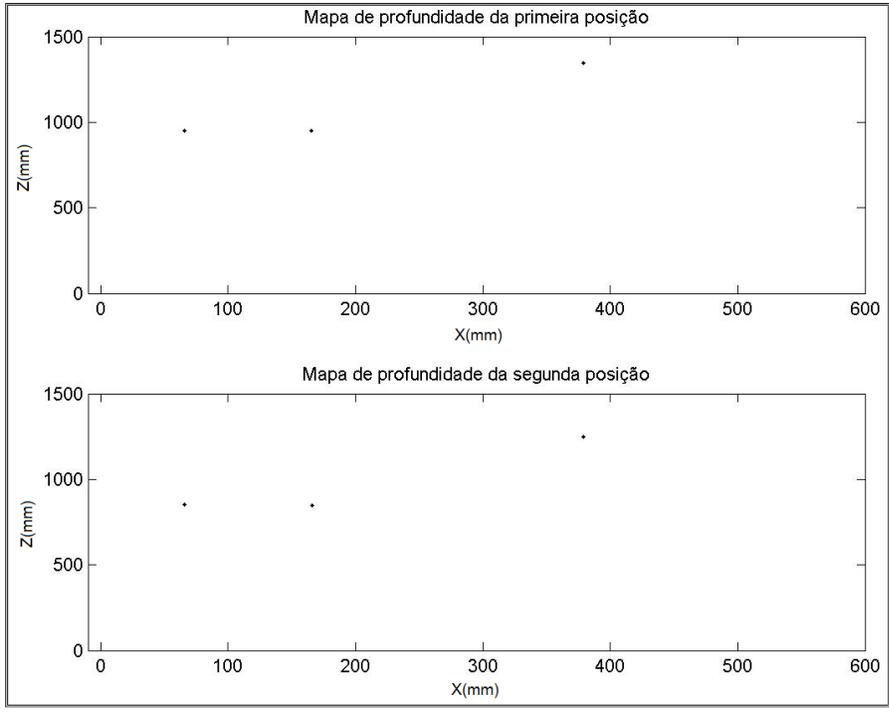


Figura 4.13 Mapas de profundidade das posições P1 e P2

Tabela 4.9 Comparação de profundidade na posição L1

Objeto	Distância pelo aplicativo Blender (mm)	Distância calculada (mm)	Erro (%)
Preto	2399.05	2382.29	0.69
Verde	2738.1	2720.76	0.63
Verde	2716.76	2704.97	0.43
Mostarda	2484.17	2439	1.81
Mostarda	2489.15	2483.09	0.24
Verde água	2872.72	2869.42	0.11
Verde água	2881.53	2902.53	0.72
Azul	2938.32	2950.82	0.42
Azul	2806.4	2827.15	0.73

Tabela 4.10 Comparação de profundidade na posição L2

Objeto	Distância pelo aplicativo Blender (mm)	Distância calculada (mm)	Erro (%)
Preto	2300.97	2288.12	0.55
Verde	2637.07	2630.44	0.25
Verde	2620.49	2613.87	0.25
Mostarda	2387.55	2364.04	0.98
Mostarda	2385.86	2368.91	0.71
Verde água	2769.57	2768.85	0.02
Verde água	2771.96	2803.21	1.12
Azul	2847.3	2853.17	0.2
Azul	2703.71	2702.13	0.05

Tabela 4.11 Comparação de profundidade na posição P1

Objeto	Distância pelo aplicativo Blender (mm)	Distância calculada (mm)	Erro (%)
Mostarda	959.02	950.4	0.89
Mostarda	967.43	962.47	0.51
Verde água	1374.63	1399.64	1.81

Tabela 4.12 Comparação de profundidade na posição P2

Objeto	Distância pelo aplicativo Blender (mm)	Distância calculada (mm)	Erro (%)
Mostarda	855.23	855.84	0.07
Mostarda	871.45	864.65	0.77
Verde água	1282.86	1304.98	1.72

Com os mapas de profundidade obtidos, estima-se a rotação e translação que ocorre entre as duas posições nos dois casos (longe e perto). Depois de estimar o movimento, são armazenados o par de imagens da posição L2 e P2, para assim, posicionar o robô na coordenada L3 e P3, para obter o par de imagens destas posições. Na Figura 4.14 são apresentados os pontos correlatos entre a posição L2 e L3; e os pontos correlatos entre a posição P2 e P3 na Figura 4.15.

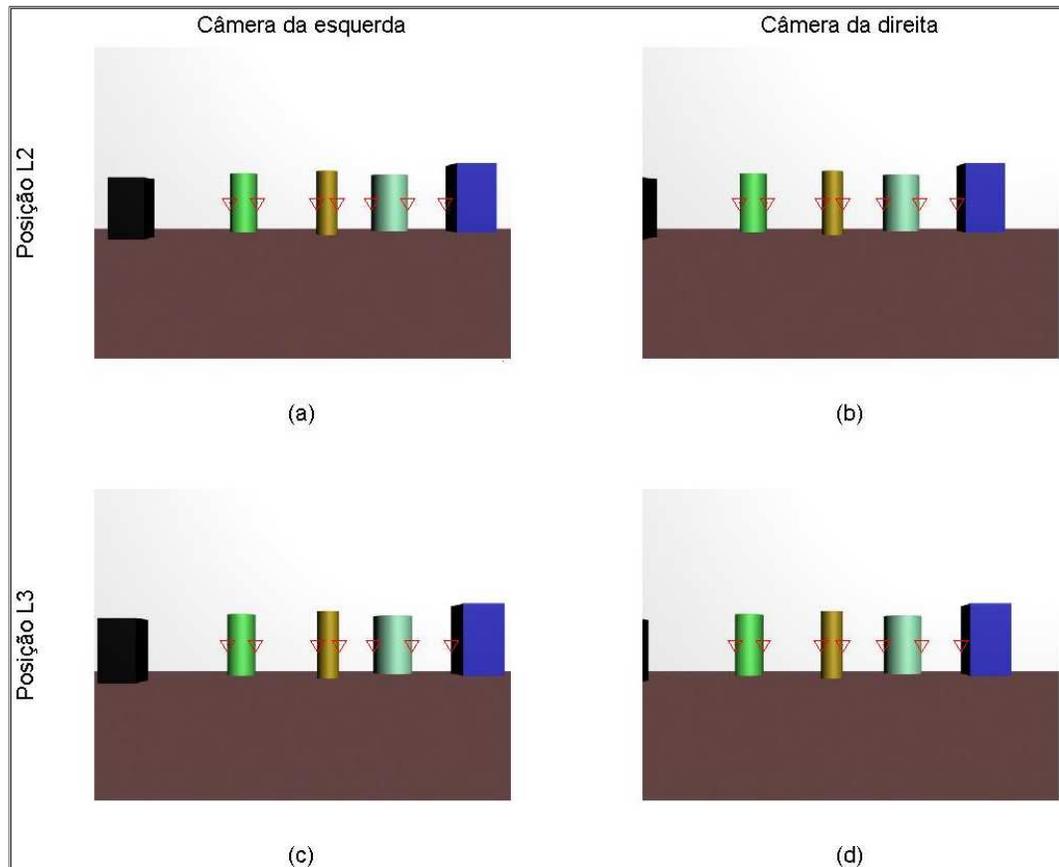


Figura 4.14 Correlação de pontos da simulação II, entre L2 e L3: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

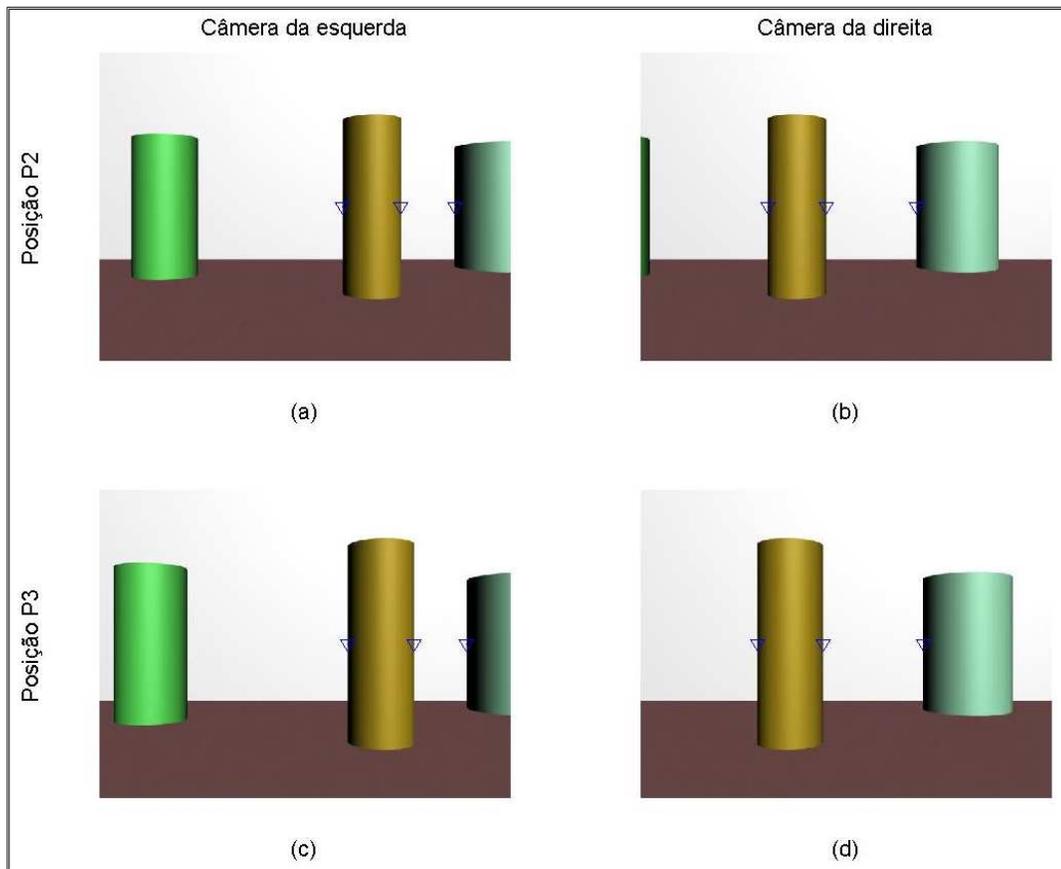


Figura 4.15 Correlação de pontos da simulação II, entre P2 e P3: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

A partir das correlações de pontos anteriores, estimam-se os mapas de profundidade de cada posição. As Figuras 4.16 e 4.17 ilustram a reconstrução do ambiente visto pelo robô nas posições L2 e L3; P2 e P3, respectivamente.

As Tabelas 4.13 e 4.14 comparam os valores das profundidades da posição L3 e P3, gerados através do aplicativo Blender e os valores obtidos através da estimativa de profundidade.

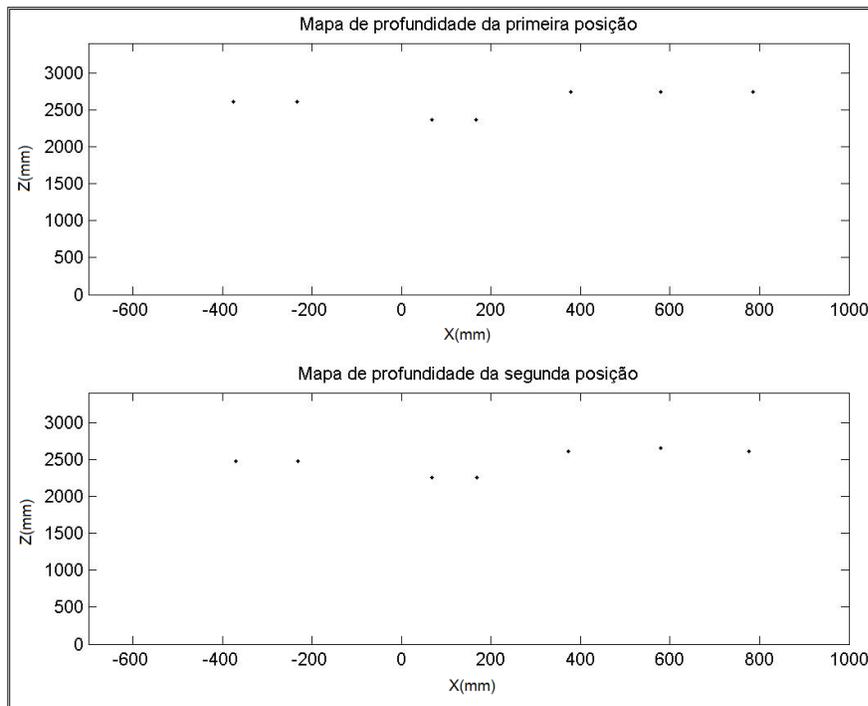


Figura 4.16 Mapas de profundidade das posições L2 e L3

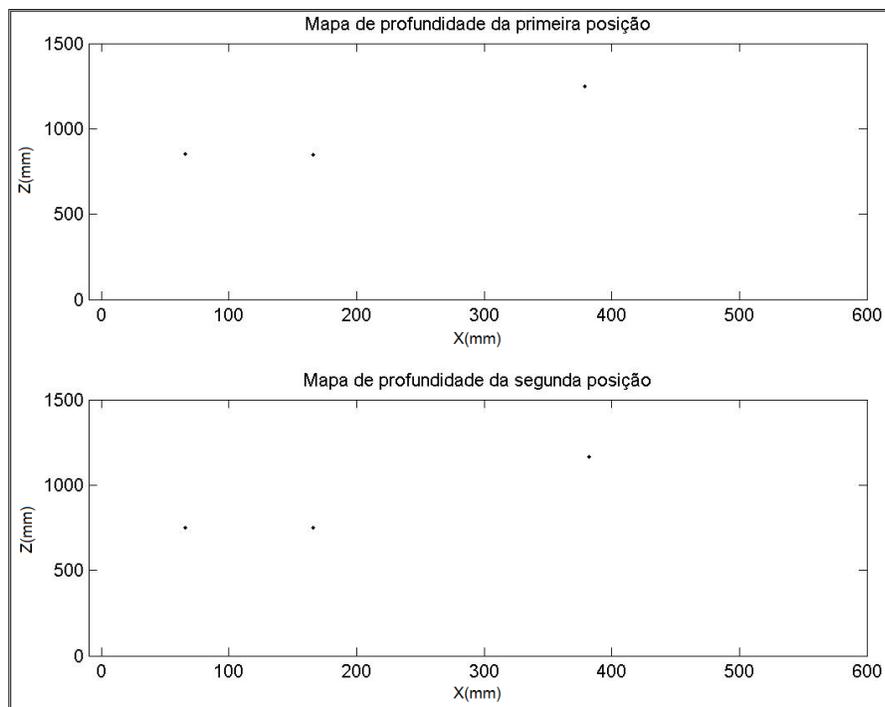


Figura 4.17 Mapas de profundidade das posições P2 e P3

Tabela 4.13 Comparação de profundidade na posição L3

Objeto	Distância pelo aplicativo Blender (mm)	Distância calculada (mm)	Erro (%)
Verde	2525.65	2505.04	0.81
Verde	2508.75	2488.28	0.81
Mostarda	2268.64	2259.83	0.38
Mostarda	2266.13	2265.03	0.04
Verde água	2680.18	2629.95	1.87
Verde água	2708.04	2710.89	0.1
Azul	2743.2	2716.65	0.96

Tabela 4.14 Comparação de profundidade na posição P3

Objeto	Distância pelo aplicativo Blender (mm)	Distância calculada (mm)	Erro (%)
Mostarda	758.7	752.11	0.86
Mostarda	770.77	767.4	0.43
Verde água	1194.25	1224.67	2.54

Analisando-se a média dos erros quando o robô está longe e perto dos objetos, observa-se que L1 apresenta 0.62%, P1 1.07%, L2 0.45%, P2 0.85%, L3 0.71% e P3 1.27% de erro. Isto significa que o ambiente reconstruído pelo método de profundidade nas diferentes posições descritas apresenta grande similitude com o ambiente proposto na simulação, com apenas poucos mm de diferença.

Com os mapas de profundidade estimados nas três posições do robô para cada simulação, estima-se o movimento que houve entre cada posição. A Figura 4.18 (a) ilustra a estimativa de movimento quando o robô navega longe dos objetos e a Figura 4.18 (b) a estimativa de movimento do robô perto dos objetos. Os quadros vermelhos desta figura representam a pose do robô onde foram obtidas as imagens, e os quadros pretos representam o movimento estimado utilizando os mapas de profundidade descritos anteriormente.

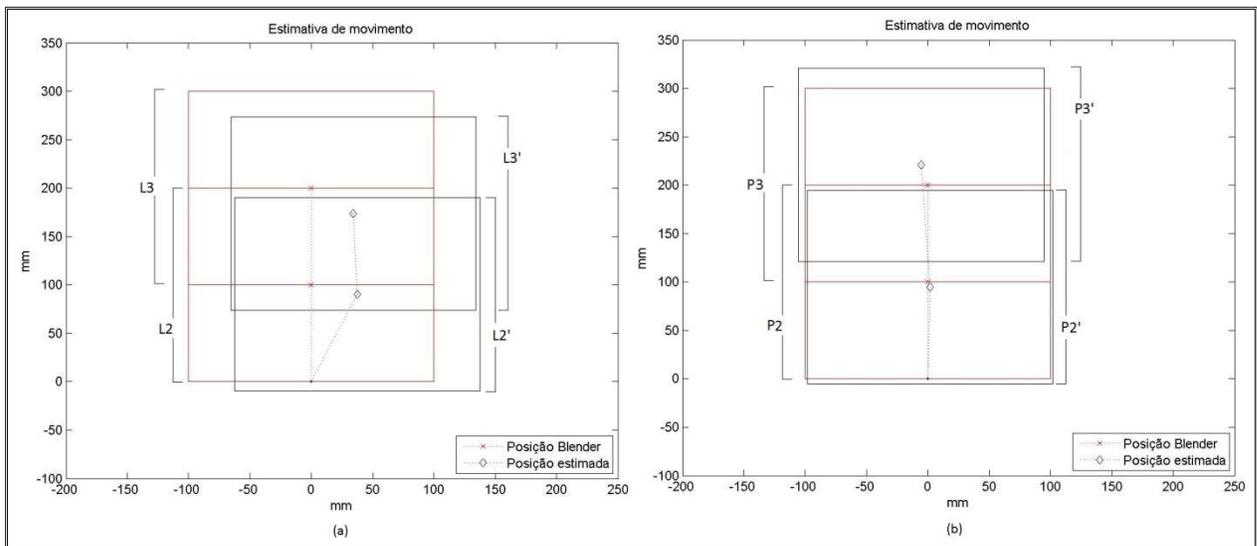


Figura 4.18 Comparação entre o movimento real e o estimado pelo método: (a) Posicionamento do robô longe dos objetos (L2' e L3'), (b) Posicionamento do robô perto dos objetos (P2' e P3')

Na figura anterior observa-se que o erro é cumulativo, ou seja, a partir da estimativa L2' é somada a estimativa de movimento L3'. Na tabela 4.15 é comparada a estimativa obtida no programa Blender e a estimativa de movimento calculado, apresentando os erros de movimento na Tabela 4.16.

Tabela 4.15 Comparação do movimento gerado no aplicativo Blender com o calculado

Posição		Blender	Calculado
L2	X (mm)	0	37.62
	Z (mm)	100	90.54
	Θ (Graus)	90	90.83
L3	X (mm)	0	-3.21
	Z (mm)	100	83.16
	Θ (Graus)	90	89.97
P2	X (mm)	0	1.88
	Z (mm)	100	94.55
	Θ (Graus)	90	90.14
P3	X (mm)	0	-7.17
	Z (mm)	100	126.32
	Θ (Graus)	90	89.16

Tabela 4.16 Comparação do movimento gerado no aplicativo Blender com o calculado

Posições (mm)	Blender (mm)	Calculado (mm)	Erro (%)
L1-L2 (mm)	100	98.05	1.95
L2-L3 (mm)	100	83.22	16
P1-P2 (mm)	100	94.57	5.43
P2-P3 (mm)	100	126.52	26

Analisando os dois casos de estudo propostos na simulação de ambiente II, nota-se o bom desempenho do algoritmo de correlação de pontos entre imagens, já que a partir destes pontos são estimados os mapas de profundidade dos objetos. Comparando a estimativa de profundidade quando o robô é posicionado longe e perto dos objetos, note-se o desempenho ótimo do método, apresentando erros nos mapas de profundidade em torno de 0.59%. No entanto, quando o robô se encontra perto dos objetos apresenta erros nos mapas de profundidade da ordem de 1.06%. Tal erro é considerado pequeno, com apenas poucos mm de diferença.

Analisando os erros calculados na estimativa de movimento do robô nos dois casos, pode se observar menor erro quando o robô está longe, devido à similitude dos mapas de profundidade utilizados. No entanto, analisando as coordenadas do movimento calculadas em X e Z em ambos os casos, observa-se um maior desvio em X quando o robô está longe e um maior desvio em Z estando perto.

Ambiente III:

Nas Figuras 4.19 e 4.20 observam-se os pontos correlatos entre o par de imagens obtido nas posições L1 e L3; P1 e P3.

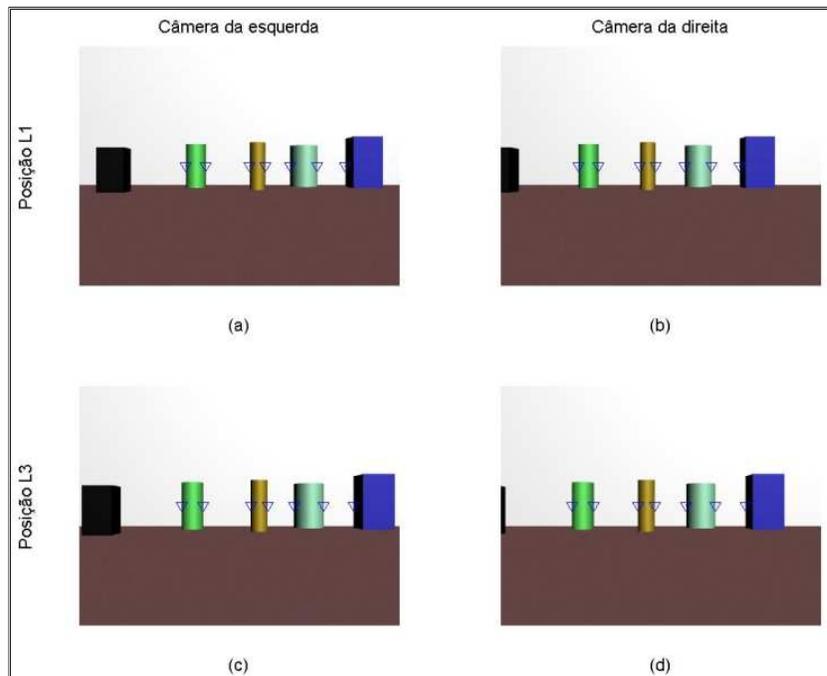


Figura 4.19 Correlação de pontos da simulação III, entre L1 e L3: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

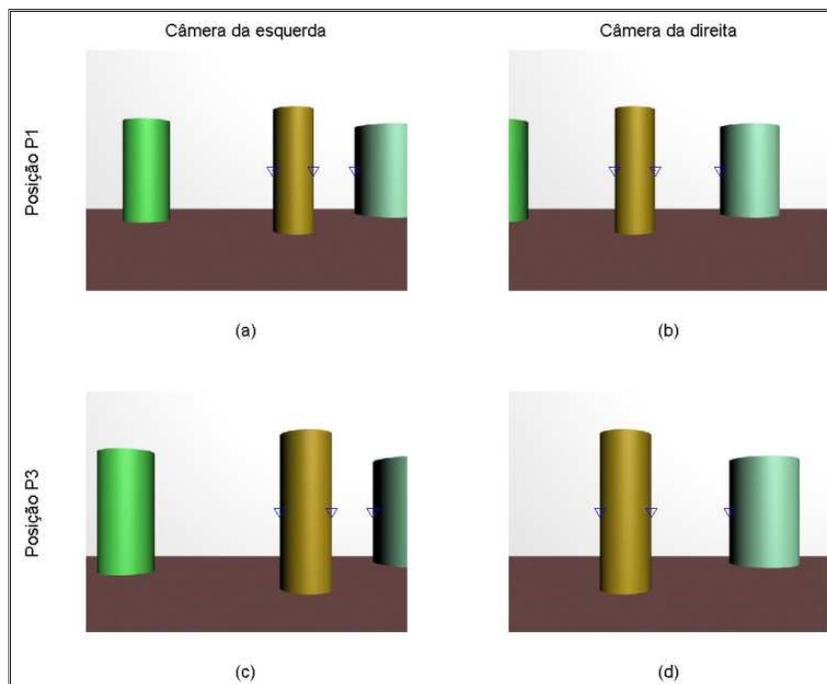


Figura 4.20 Correlação de pontos da simulação III, entre P1 e P3: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

A estimação do ambiente dos pontos correlatos nas posições L1 e L3, apresenta-se na Figura 4.21 e das posições P1 e P3 na Figura 4.22.

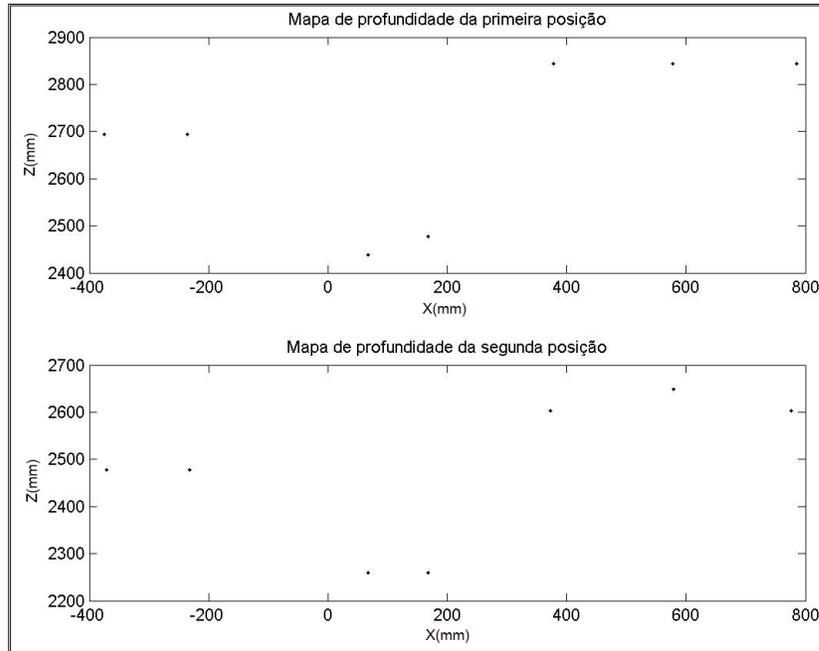


Figura 4.21 Mapas de profundidade das posições L1 e L3

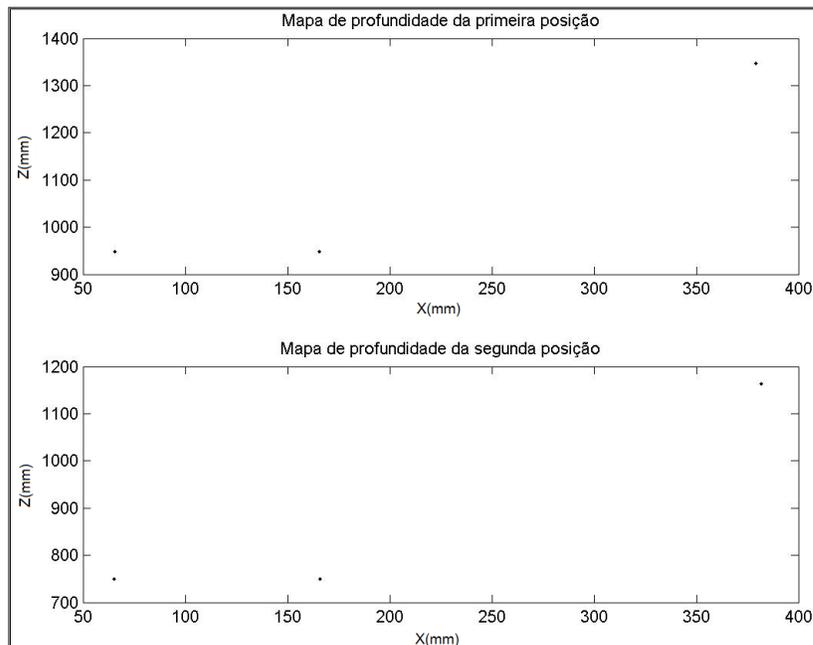


Figura 4.22 Mapas de profundidade das posições P1 e P3

Os dados medidos no software Blender são comparados na Tabela 4.17 e 4.18, com as profundidades estimadas nas posições L1 e L3.

Tabela 4.17 Comparação de profundidade na posição L1

Objeto	Distância pelo aplicativo Blender (mm)	Distância calculada (mm)	Erro (%)
Verde	2738.1	2720.76	0.63
Verde	2716.76	2704.97	0.43
Mostarda	2484.17	2439	1.81
Mostarda	2489.15	2483.09	0.24
Verde água	2872.72	2869.42	0.11
Verde água	2881.53	2902.53	0.72
Azul	2938.32	2950.82	0.42

Tabela 4.18 Comparação de profundidade na posição L3

Objeto	Distância pelo aplicativo Blender (mm)	Distância calculada (mm)	Erro (%)
Verde	2525.65	2505.04	0.81
Verde	2508.75	2488.28	0.81
Mostarda	2268.64	2259.83	0.38
Mostarda	2266.13	2265.03	0.04
Verde água	2680.18	2629.95	1.87
Verde água	2708.04	2710.89	0.1
Azul	2743.2	2716.65	0.96

Observe-se que os erros apresentados nas tabelas anteriores, correspondem às profundidades estimadas no “ambiente II”. Apesar de o movimento haver sido de 200 mm entre as imagens, a reconstrução dos mapas de profundidade não varia, significando que a precisão da reconstrução do cenário não depende do número de mostras ou pontos correlatos e sim das coordenadas de pixel de cada borda no par de imagens.

Utilizando os mapas de profundidade anteriores estima-se o movimento entre cada posição, como ilustra a Figura 4.23. Na tabela 4.19 é comparada a estimativa de movimento obtida pelo aplicativo Blender e a estimativa de movimento calculado, apresentando os erros de movimento na Tabela 4.20.

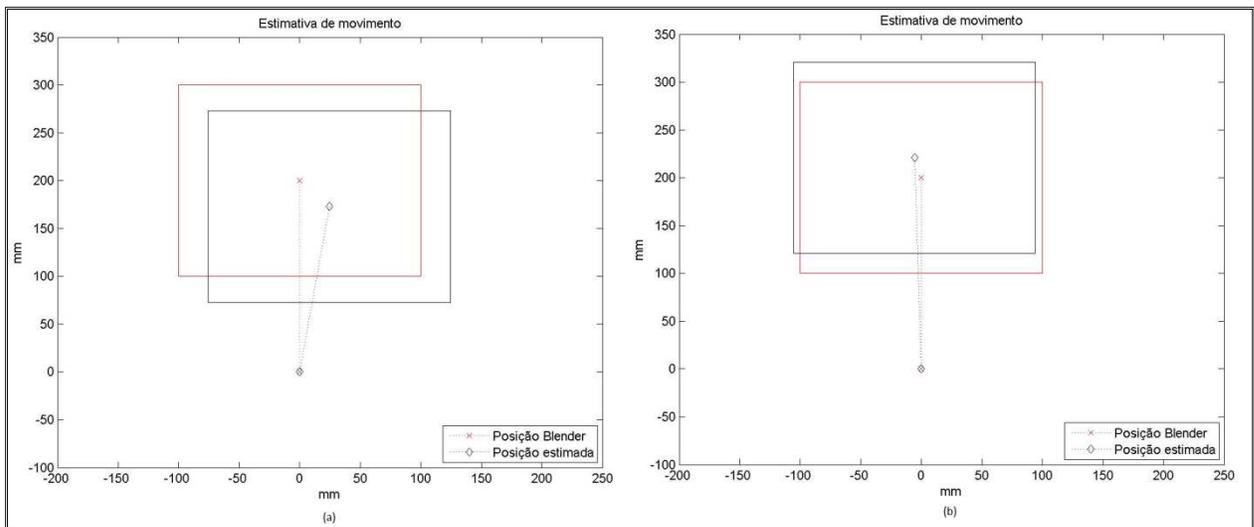


Figura 4.23 Comparação de movimento real e o estimado: (a) Posicionamento do robô longe dos objetos, (b) Posicionamento do robô perto dos objetos

Tabela 4.19 Comparação do movimento gerada no aplicativo Blender com o calculado

Translação (mm)	Valores em Blender	Valores calculados
L1-L3 em X	0	24.37
L1-L3 em Z	200	172.82
P1-P3 em X	0	-5.67
P1-P3 em Z	200	221.15

Tabela 4.20 Comparação do movimento gerado no aplicativo Blender com o calculado

Posições (mm)	Blender	Calculados	Erro (%)
L1-L3	200	174.53	12.7
P1-P3	200	221.22	10.6

Comparando os dois casos, observe-se que as imagens correlacionadas apresentam estimativas boas, com acerto em todas as possibilidades de correlação de bordas entre as imagens. O interessante neste ambiente é a observação na estimativa de movimento, já que o erro de movimento nos dois casos diminuiu comparado com o movimento a 100 mm, descrito no ambiente II. No entanto, quando o robô é posicionado perto dos objetos, esta redução de erro é mais significativa.

Ambiente IV:

Inicialmente o robô é posicionado na coordenada 1 descrito pela Tabela 4.6, onde são obtidas e armazenadas duas imagens. Em seguida, o robô é posicionado na coordenada 2 dessa tabela, e são obtidas duas novas imagens. É aplicado o método de correlação de pontos unidimensionais em busca de pontos de similares entre as quatro imagens, sendo calculado o mapa de profundidade em cada posição. Estes pontos são finalmente utilizados para estimar o movimento entre os dois pares de imagens. Depois de estimar o movimento entre a posição 1 e 2, armazena-se as imagens da posição dois e posiciona-se o robô na coordenada 3 da tabela, para que se executem os diferentes métodos de estimativa de deslocamento. Este processo se repete para todas as posições da tabela, simulando o movimento do robô em direção aos objetos.

O objetivo final deste ambiente é observar o comportamento dos algoritmos durante a trajetória do robô, sendo ilustradas as imagens correlatas e os mapas de profundidade entre as posições 1-2, 3-4, 5-6, 7-8, 9-10, 11-12 e finalmente as posições 12-13. O método de correlação de pontos entre a posição 1-2 é apresentado na Figura 4.24 e os mapas de profundidade destes pontos são ilustrados na Figura 4.25.

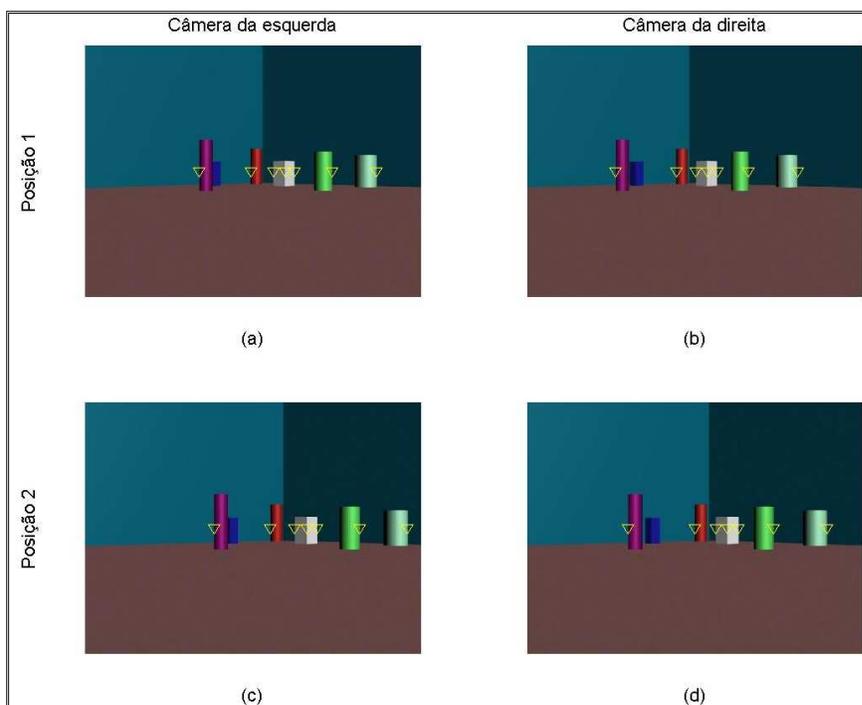


Figura 4.24 Correlação de pontos da simulação IV, entre a posição 1 e 2: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

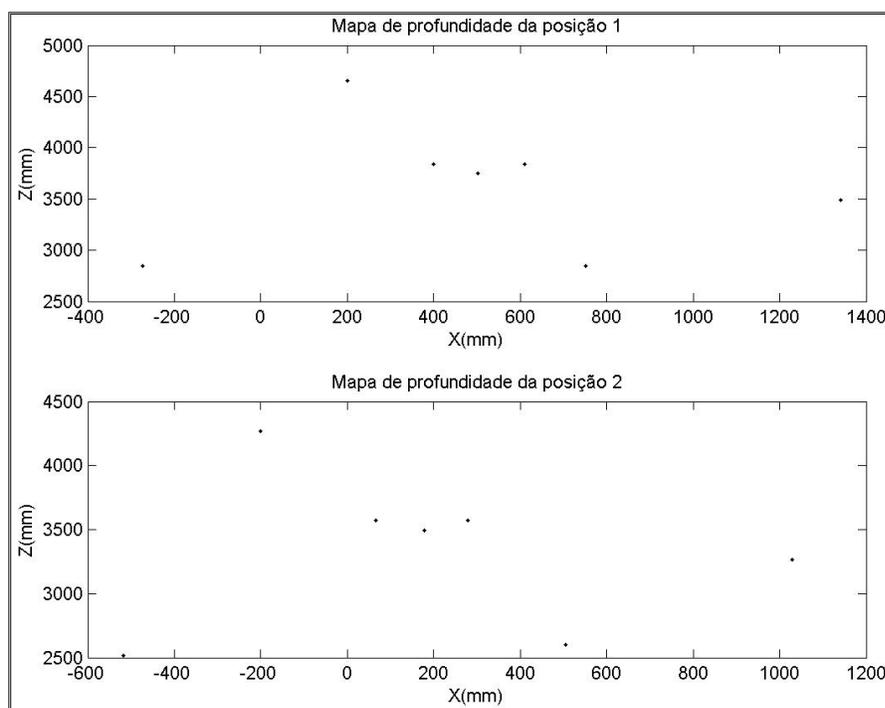


Figura 4.25 Mapas de profundidade das posições 1 e 2

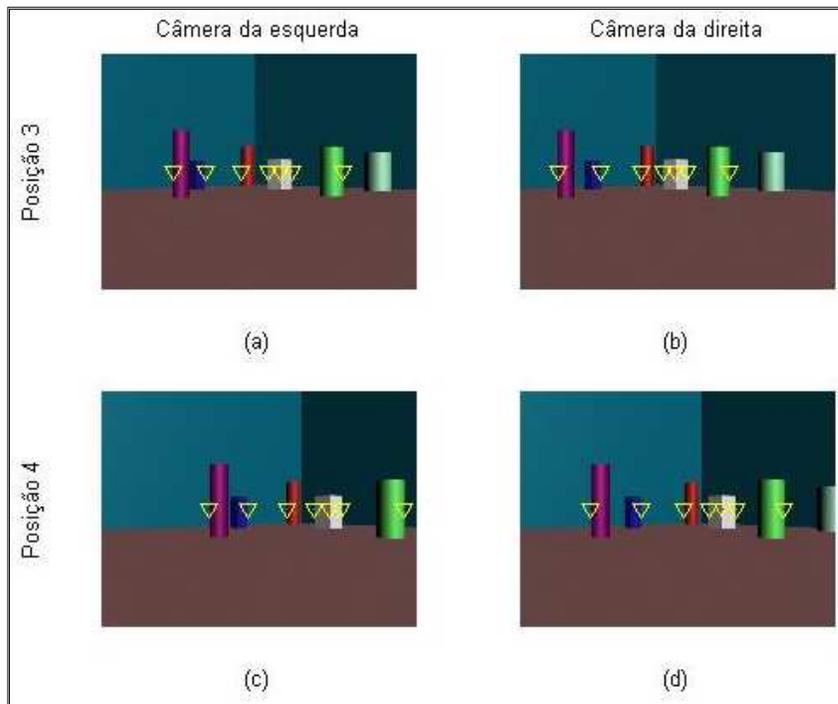


Figura 4.26 Correlação de pontos da simulação IV, entre a posição 3 e 4: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

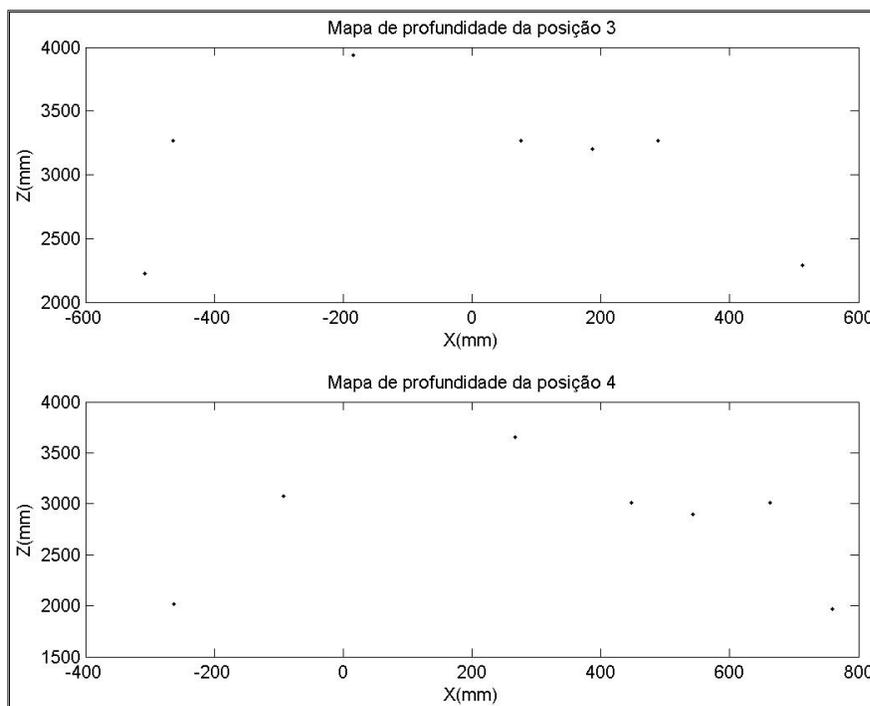


Figura 4.27 Mapas de profundidade das posições 3 e 4

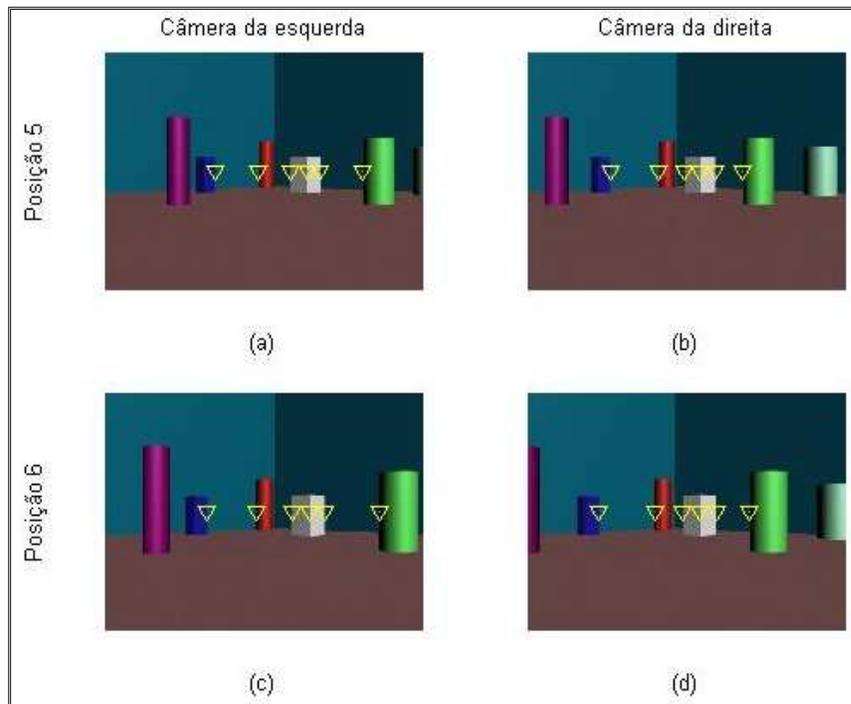


Figura 4.38 Correlação de pontos da simulação IV, entre a posição 5 e 6: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

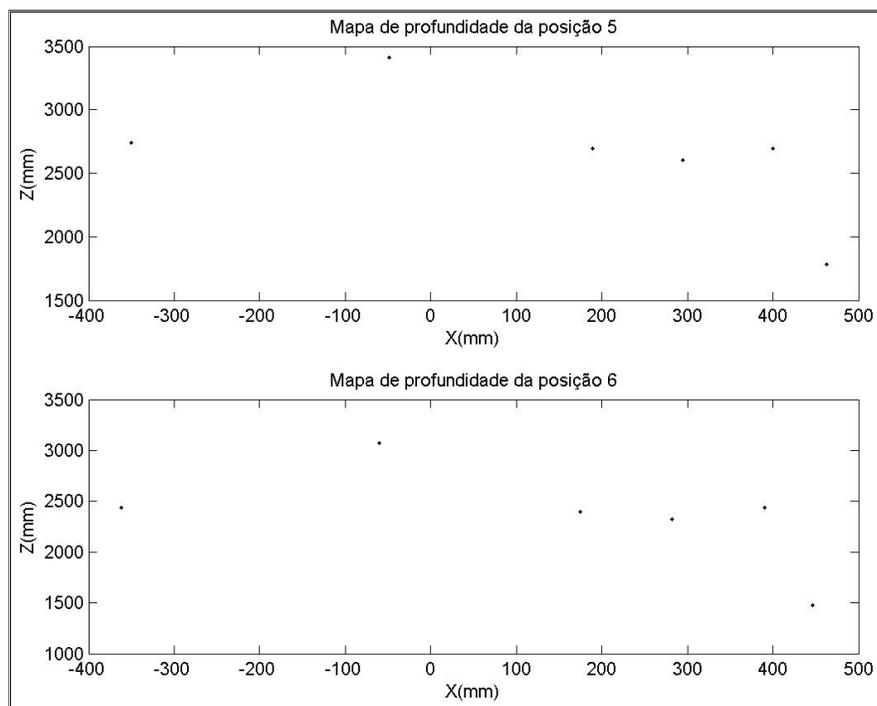


Figura 4.29 Mapas de profundidade das posições 5 e 6

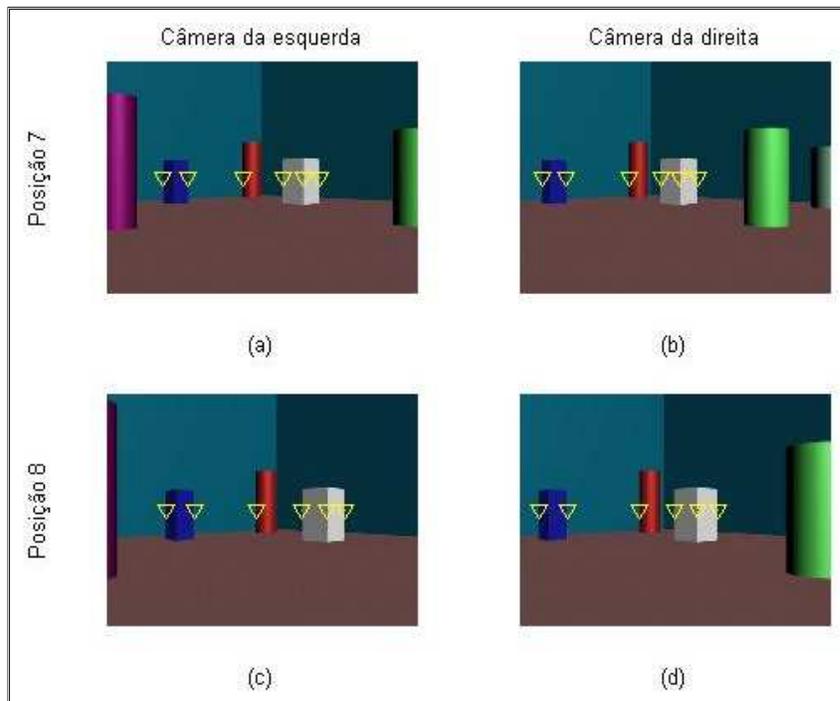


Figura 4.30 Correlação de pontos da simulação IV, entre a posição 7 e 8: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

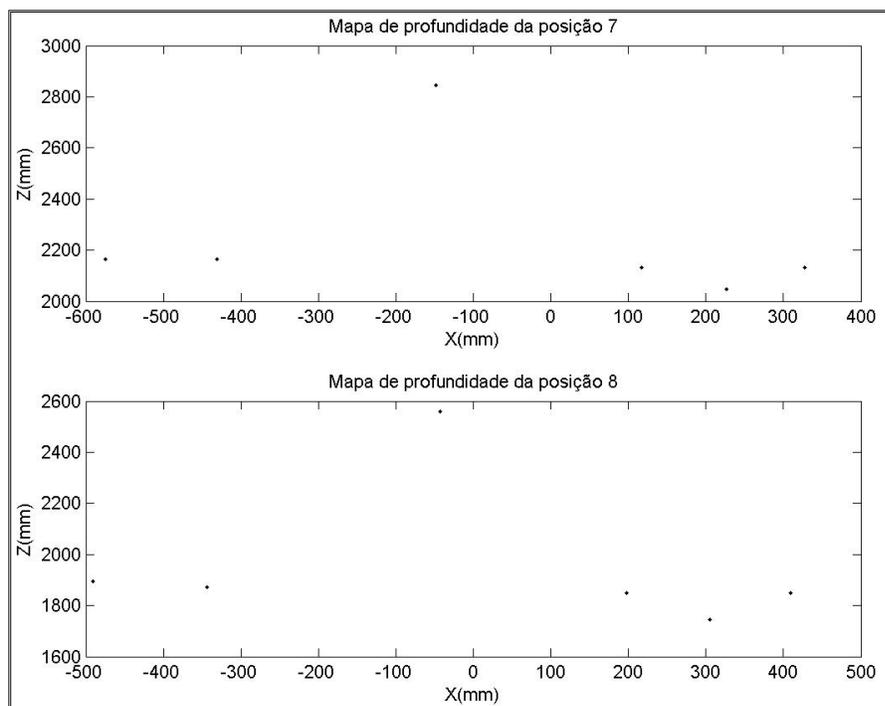


Figura 4.31 Mapas de profundidade das posições 7 e 8

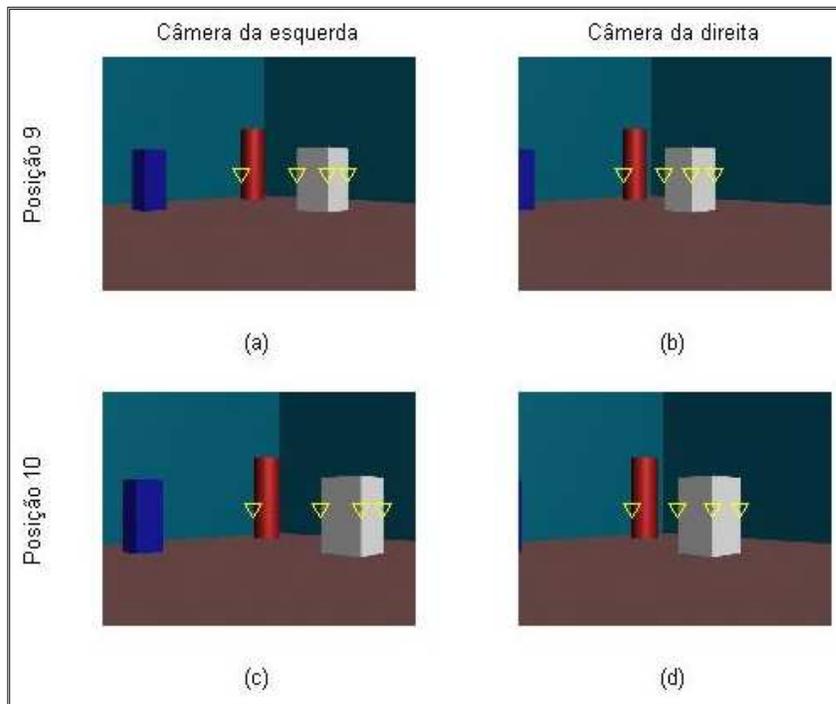


Figura 4.32 Correlação de pontos da simulação IV, entre a posição 9 e 10: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

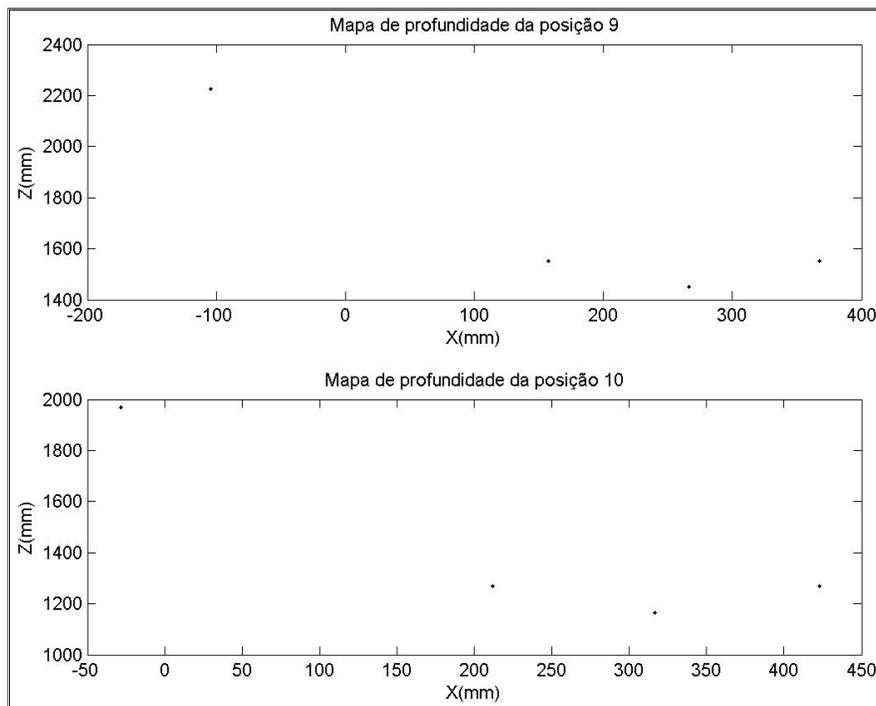


Figura 4.33 Mapas de profundidade das posições 9 e 10

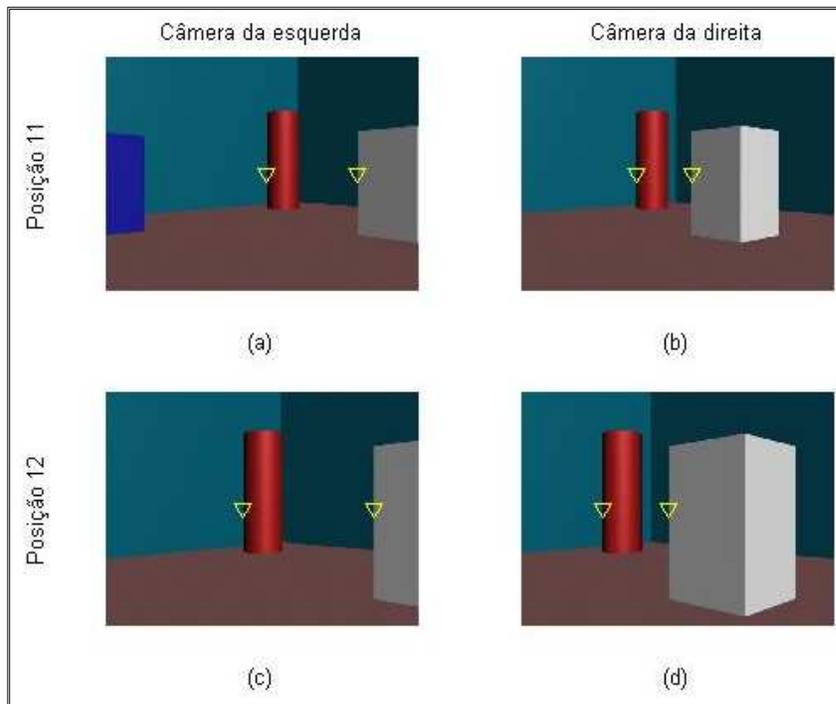


Figura 4.34 Correlação de pontos da simulação IV, entre a posição 11 e 12: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

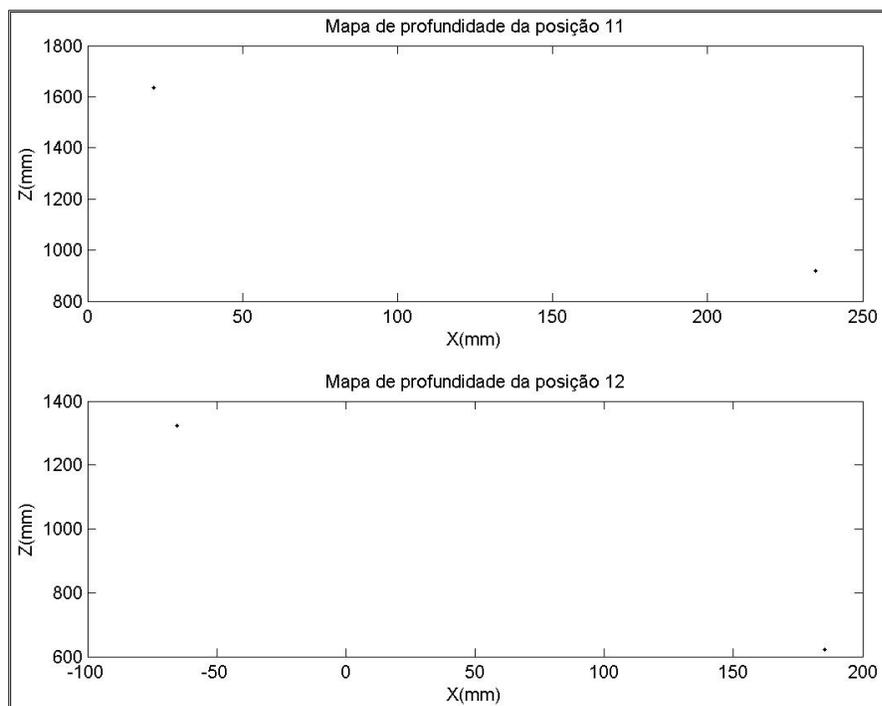


Figura 4.35 Mapas de profundidade das posições 11 e 12

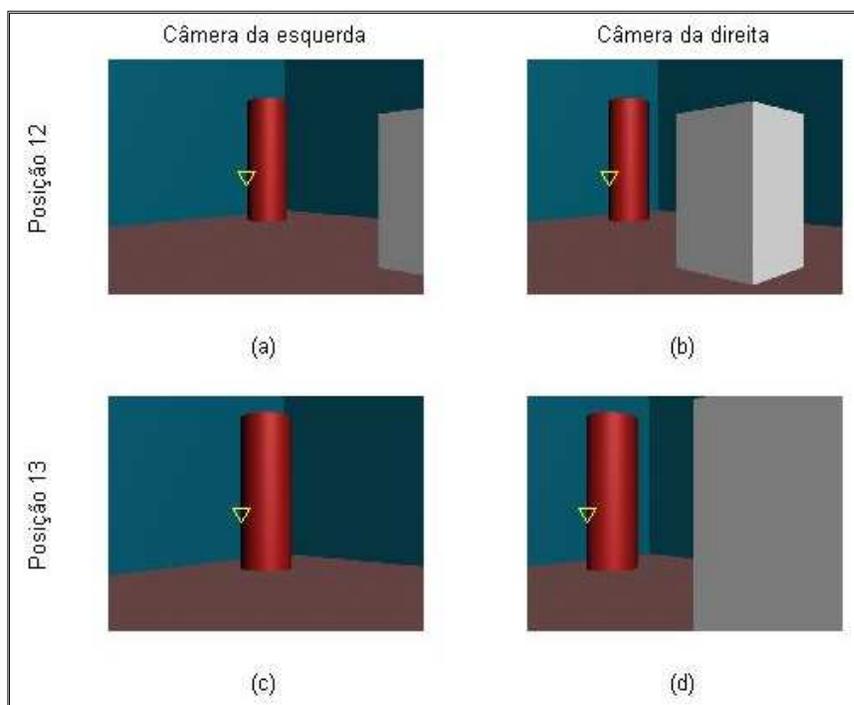


Figura 4.36 Correlação de pontos da simulação IV, entre a posição 12 e 13: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

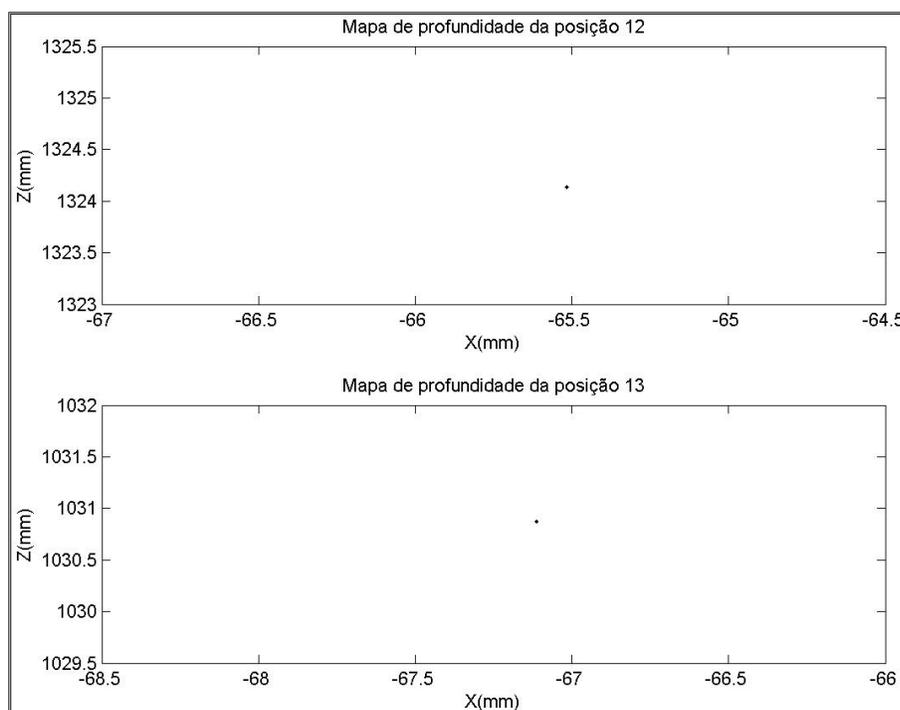


Figura 4.37 Mapas de profundidade das posições 12 e 13

Efetua-se em seguida a comparação do deslocamento calculado durante a trajetória e o medido no software Blender. Na tabela 4.21 apresenta os pontos correlatos entre as quatro imagens obtidas a cada duas posições e as comparações entre as medidas de movimento.

Tabela 4.21 Número de pontos correlacionados e erro apresentado na estimativa de movimento durante as diferentes posições em que foram obtidas as imagens

Posições	No. De pontos correlatos	Movimento no aplicativo Blender (mm)	Movimento calculado (mm)	Erro %
1-2	7	290	229.7	20
2-3	10	296.9	341.3	14.9
3-4	7	261.7	239.5	8.4
4-5	7	289.8	303.5	4.7
5-6	6	292.2	271.9	6.94
6-7	5	296.8	365.7	23.2
7-8	6	289.9	284	2
8-9	5	296.9	275.2	7.3
9-10	4	282.8	292.3	3.3
10-11	2	346.5	348.5	0.5
11-12	2	304.1	309.2	1.67
12-13	1	282.8	NaN	

Pode-se ver da tabela anterior, na estimativa de movimento, que o método contém mais erro quando o robô se encontra mais distante dos objetos. Na medida em que o robô se aproxima dos objetos, o erro diminui, já que as coordenadas do conjunto de pixels correlacionados são definidas com maior precisão. Observa-se na posição 12-13 o método identificou apenas um ponto em cada uma das quatro imagens, fazendo com que a solução do sistema linear descrito no capítulo anterior seja inexistente.

A trajetória do robô durante todo o percurso é apresentada na Figura 4.38, que representa a vista superior do ambiente; Círculos vermelhos representam cilindros e estrelas azuis representam as caixas do ambiente. O deslocamento realizado da primeira posição até a última foi de 3530.4 mm, distância considerável onde se observa também erros constantes durante a trajetória. A trajetória estimada é representada pela linha azul.

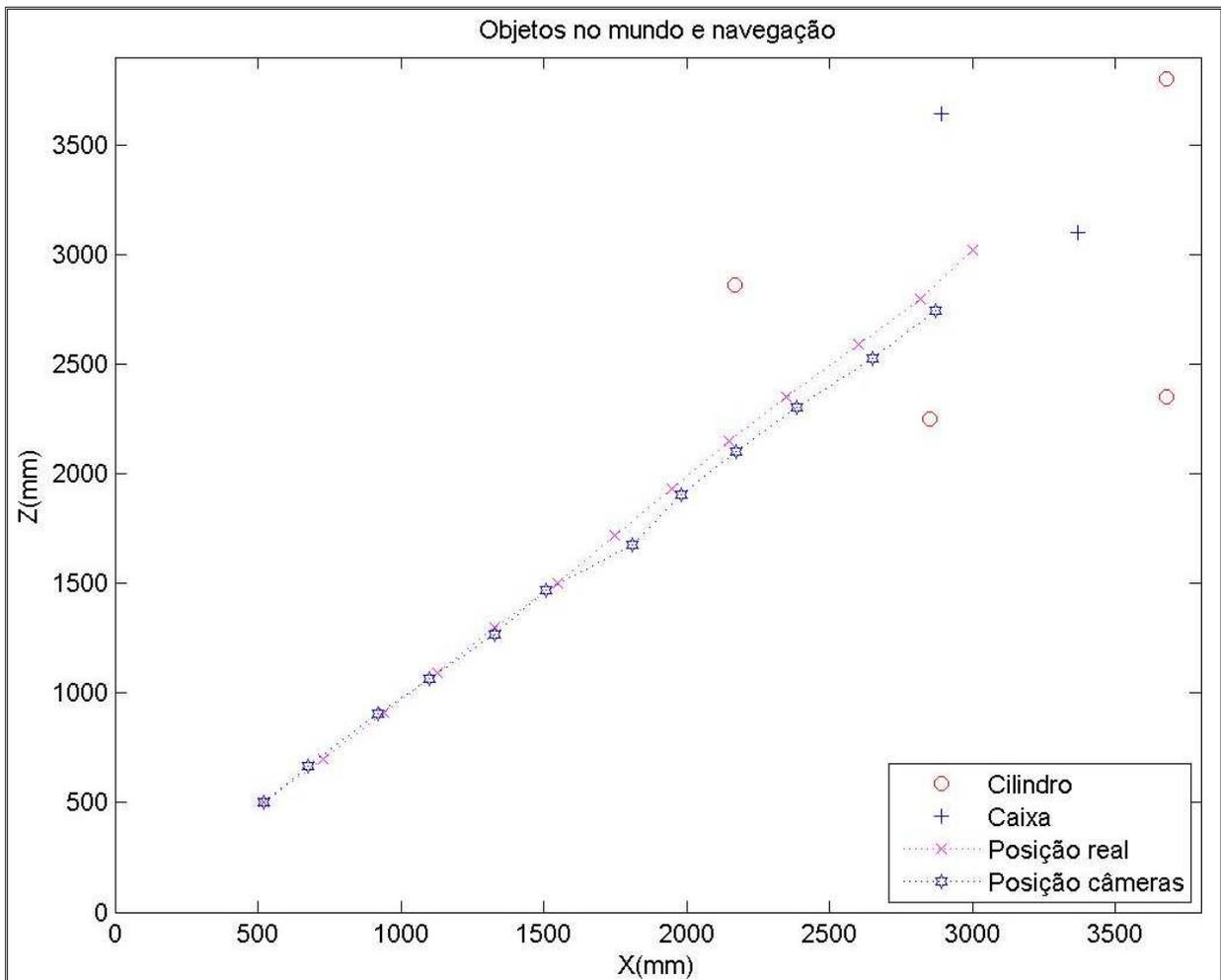


Figura 4.38 Simulação do percurso do robô na simulação de ambiente IV

Apesar do erro constante no transcurso da trajetória, a estimativa de movimento do robô é próxima à navegação real.

- Análise geral das simulações

O método de correlação de imagens unidimensionais desenvolvido apresenta ótimo desempenho nos diferentes ambientes simulados. Observou-se que independentemente da cor de fundo, as bordas são detectadas eficientemente, identificando os objetos visíveis nas imagens. O número de pontos correlatos identificados pelo método proposto está relacionado com o número de bordas extraídas pela segmentação, como verificado na análise do ambiente I. Em relação aos pontos de oclusão presentes no ambiente I, observou-se uma boa detecção, o que é importante para se evitar uma correlação errada.

Analisando os mapas de profundidade obtidos a partir das correlações, observa-se uma boa correspondência com as dimensões conhecidas nos diversos ambientes simulados. O erro máximo de estimativa de movimento aconteceu quando as câmeras se encontram distantes dos objetos, apresentando uma média de erros de 0.95%; Os maiores erros são obtidos quando os objetos possuem profundidades mais pronunciadas. Por outro lado, quando o robô é posicionado perto dos objetos, a média dos erros obtidos é de 1.06%. A eficiência dos mapas de profundidade não está ligada ao número de pontos correlacionados, mas a precisão das coordenadas dos pixels correspondentes às bordas dos objetos. Apesar de se obter resultados visualmente consistentes, é importante ressaltar que a eficiência do método depende do grau de precisão que se deseja para a aplicação prática.

A relação epipolar estabelece que se 1 mesmo ponto é visto em duas posições, pode-se então calcular a translação e rotação entre essas duas posições. Se porém a estimativa da profundidade do ponto é obtida com erro nessas duas posições, observa-se embora que o movimento estimado é próximo ao real, com apenas um pequeno desvio. Na estimativa de movimento de uma trajetória longa, como a de 3530.4 mm do exemplo mostrado, observa-se que os erros são constantes. O erro entretanto diminui conforme se vai aproximando dos objetos presentes nas imagens, o que é equivalente ao movimento real do robô.

5. Aplicação dos métodos em ambientes reais

Neste capítulo são mostrados os resultados obtidos nos experimentos realizados de navegação em ambientes de interiores, tendo como sistema móvel o robô Pioneer 3 com duas câmeras AXIS 213. Para validar os métodos propostos neste trabalho é necessário avaliar o comportamento dos algoritmos de identificação de movimento quando submetidos a imagens e ambientes reais, como ocorre na navegação de robôs autônomos. Inicialmente são apresentadas as especificações do robô e da câmera. Em seguida são descritos os ambientes nos quais se testam os algoritmos e finalmente são apresentados as análises e resultados.

5.1. Especificações técnicas do robô Pioneer 3

O robô Pioneer 3 é usado para pesquisas acadêmicas e possui tamanho mediano como apresenta a Figura 5.1. É completamente autônomo e útil para navegar em lugares estreitos como escritórios e laboratórios. Consiste de uma base móvel com um par de motores, dois arranjos de oito sonares, um *encoder* em cada motor e dois arranjos de cinco *bumpers*. Utiliza um microcontrolador H8S para processar as medidas sensoriais e enviar sinais de movimento aos motores. Além disso, o robô Pioneer 3 tem um servidor Arcos, software MobileRobots e um PC a bordo com sistema operativo AROS de baixo nível.

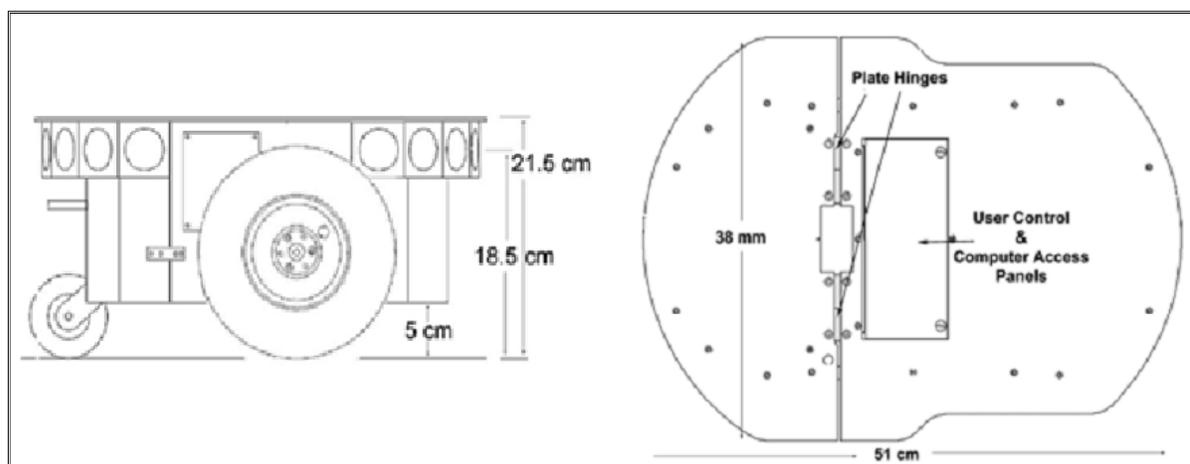


Figura 5.1 Dimensões do robô Pioneer 3

Fonte [34]. Pioneer 3 Operations Manual. [cited 2009 Novembro 9]; Available from: <http://143.106.50.145:8080/Cursos/IA368N/02-09/P3OpMan5.pdf>

Nas Figuras 5.2 (a) e (c) ilustram-se os arranjos de sonares e *bumpers* da parte frontal e parte traseira do robô, e na Figura 5.2 (b) observa-se o painel de usuário.

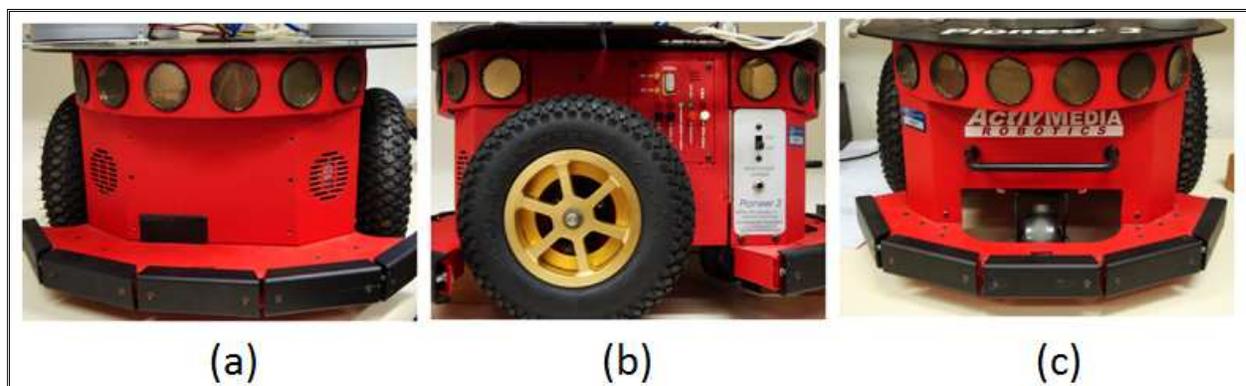


Figura 5.2 Partes do robô Pioneer 3. (a) Visão frontal do robô, (b) Painel de controle do usuário, (c) Visão traseira do robô

O robô utilizado nesse projeto foi cedido por empréstimo de cooperação com a Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas. O grupo de pesquisa daquele instituto, sob coordenação do Professor Eleri Cardozo, desenvolve aplicações de simulação, comunicação e controle remoto de sistemas robóticos [6]. O servidor da plataforma

MobileRobots tem arquitetura cliente-servidor, ou seja, é uma arquitetura que liga clientes e servidores, através de uma rede de computadores. O cliente pode ser um PC conectado através de modems de rádio, Ethernet serial sem fio, laptop a bordo ou incorporado, como apresenta a Figura 5.3. Neste trabalho foi utilizada uma rede Ethernet sem fio para conectar-se ao robô.

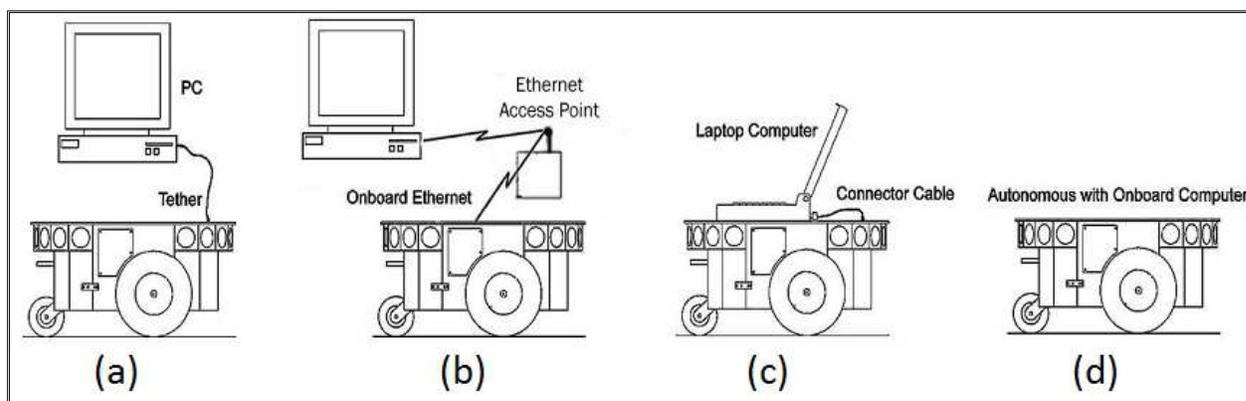


Figura 5.3 Conexão do cliente. (a) Modem de rádio, (b) Ethernet sem fio, (c) Laptop a bordo, (d)

Autônomo

Fonte [34]. Pioneer 3 Operations Manual. [cited 2009 Novembro 9]; Available from: <http://143.106.50.145:8080/Cursos/IA368N/02-09/P3OpMan5.pdf>

Para realizar o teste experimental no robô Pioneer foram desativados os arranjos de sonares e *bumpers*, e acrescentadas duas câmeras IP. Os comandos de movimento do robô, foram feitos utilizando API's (Application Programming Interface) criadas em MatLab.

5.2. Especificações técnicas da câmera

Para o desenvolvimento deste trabalho foram utilizadas duas câmeras do tipo AXIS 213 PTZ Network Camera apresentada na Figura 5.4. Esta câmera é desenvolvida pela empresa AXIS Communication e apresenta as seguintes especificações técnicas [1]:

- Oferece controle horizontal e vertical em tempo real através de uma conexão de rede IP.
- 26x de zoom óptico e 12x de zoom digital, enfoque automático e auto Iris.
- Tem projetor de luz infravermelho integrado.
- 10 resoluções até 768x576 pixel.
- Fácil instalação utilizando AXIS Internet Dynamic DNS Service ou a utilidade AXIS IP.



Figure 5.4 Câmera IP AXIS 213 PTZ

Fonte [34]. AXIS 213 PTZ Network Camera User's Manual. [cited 2010 Junho 21]; Available from: http://www.axis.com/es/products/cam_213/index.htm

Para fixar as câmeras na cobertura do robô, foram construídas duas bases com o propósito de aproveitar os furos próprios da cobertura do robô. Na Figura 5.5 (a) é ilustrada a base utilizada, onde os cilindros vermelhos representam os parafusos que fixam a base da câmera com a cobertura do robô e os cilindros brancos representam os parafusos que fixam a base com a câmera. A distância entre as câmeras é de 280 mm.

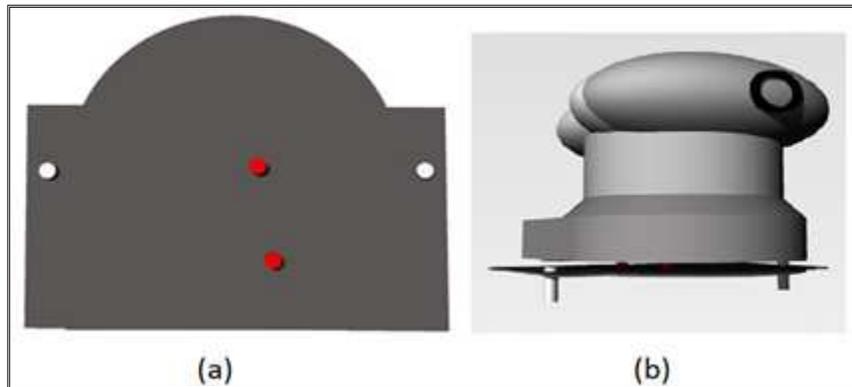


Figura 5.5 Fixação das câmeras. (a) Base da câmera, (b) Câmera sobre a base

Após conectar as câmeras ao robô são testadas as API's utilizadas em Matlab para a captura das imagens e controle de movimento do robô. Estas API's são encarregadas de enviar comandos de controle via wireless, necessários para desenvolver os testes experimentais descritos a seguir.

5.3. Posicionamento real do robô

Para realizar os testes experimentais o robô Pioneer 3 é colocado no laboratório de processamento de sinais e análise de sistemas dinâmicos. Em um ambiente com área de 4 m^2 aproximadamente com chão plano. São realizados dois testes utilizando objetos simples no campo visual das câmeras. O robô desloca-se em linha reta em intervalos de 200 mm entre as capturas de dois pares de imagens. O referencial absoluto do sistema de coordenadas é criado na posição inicial do robô. A partir desta posição é estimado o percurso do robô no ambiente.

O robô na posição inicial captura e armazena duas imagens do ambiente a sua frente. Movimenta-se 200 mm até a posição seguinte para capturar o segundo par de imagens. Após obter os pares de imagens de cada posição, é aplicado o método de correlação de imagens unidimensionais extraindo as características principais do ambiente. Utilizando as coordenadas dos pontos correlatos em cada posição é feita a triangulação para estimar a profundidade dos

objetos utilizando a relação epipolar para estimar o movimento entre as duas posições. Se o robô continua se movimentando, as imagens capturadas na última posição, são armazenadas para que se repeta o processo descrito anteriormente.

Ambiente I: Neste primeiro ambiente são utilizados três objetos, um extintor, uma caixa de papelão e um tubo preto, como ilustra a Figura 5.6. O posicionamento do robô está descrito na Tabela 5.1 e as coordenadas dos objetos estão ilustradas na Tabela 5.2. Nesta tabela as coordenadas estão referenciadas desde a origem do robô até o centro dos objetos, sabendo que o raio do extintor e o tubo preto são de 85 e 19 mm respectivamente e a caixa de papelão possui 200x446 mm.

Tabela 5.1 Coordenada de movimento do robô no ambiente I

Posição	Coordenada em X (mm)	Coordenada em Z (mm)
1	0	0
2	0	200

Tabela 5.2 Coordenada dos objetos no ambiente I

Objetos	Coordenada em X (mm)	Coordenada em Z (mm)
Extintor	-415	2485
Caixa de papelão	0	2623
Tubo preto	481	2650



Figura 5.6 Vista do primeiro ambiente

Ambiente II: Neste ambiente são utilizados quatro objetos como ilustrado na Figura 5.7. Os três objetos descritos no ambiente anterior mais um gerador de sinais com dimensão de 450x175 mm. Neste ambiente os objetos são posicionados a uma distância de 2400 mm do robô. Pretende-se analisar os algoritmos durante o movimento do robô em direção aos objetos, sendo adquiridas duas imagens cada 200 mm, como apresentado na Tabela 5.3. As coordenadas dos objetos são ilustradas na Tabela 5.4. Estas coordenadas são descritas do referencial absoluto do robô até o centro dos objetos.

Tabela 5.3 Coordenada de movimento do robô no ambiente II

Posição	Coordenada em X (mm)	Coordenada em Z (mm)
1	0	0
2	0	200
3	0	400
4	0	600
5	0	800
6	0	1000
7	0	1200
8	0	1400
9	0	1600

Tabela 5.4 Coordenada dos objetos no ambiente II

Objetos	Coordenada em X (mm)	Coordenada em Z (mm)
Extintor	125	2825
Caixa de papelão	-300	2623
Tubo preto	-131	2569
Gerador de sinais	575	2487.5



Figura 5.7 Vista do segundo ambiente

5.4. Análise e resultados

A seguir são apresentados os resultados dos testes experimentais de cada ambiente.

Ambiente I:

A partir dos pares de imagens das duas posições é aplicado o método de correlação de pontos. Este método, primeiro procura as bordas correspondentes entre as imagens “CE_PAN” e

“CE_PAT” na Correlação I descrita na Figura 5.8. Os pontos da Correlação I são buscados na imagem “CD_PAN” como ilustra a Correlação II da figura 5.8. Finalmente estes pontos são buscados na imagem “CD_PAT”, a seguir é ilustrado passo a passo este método de correlação.

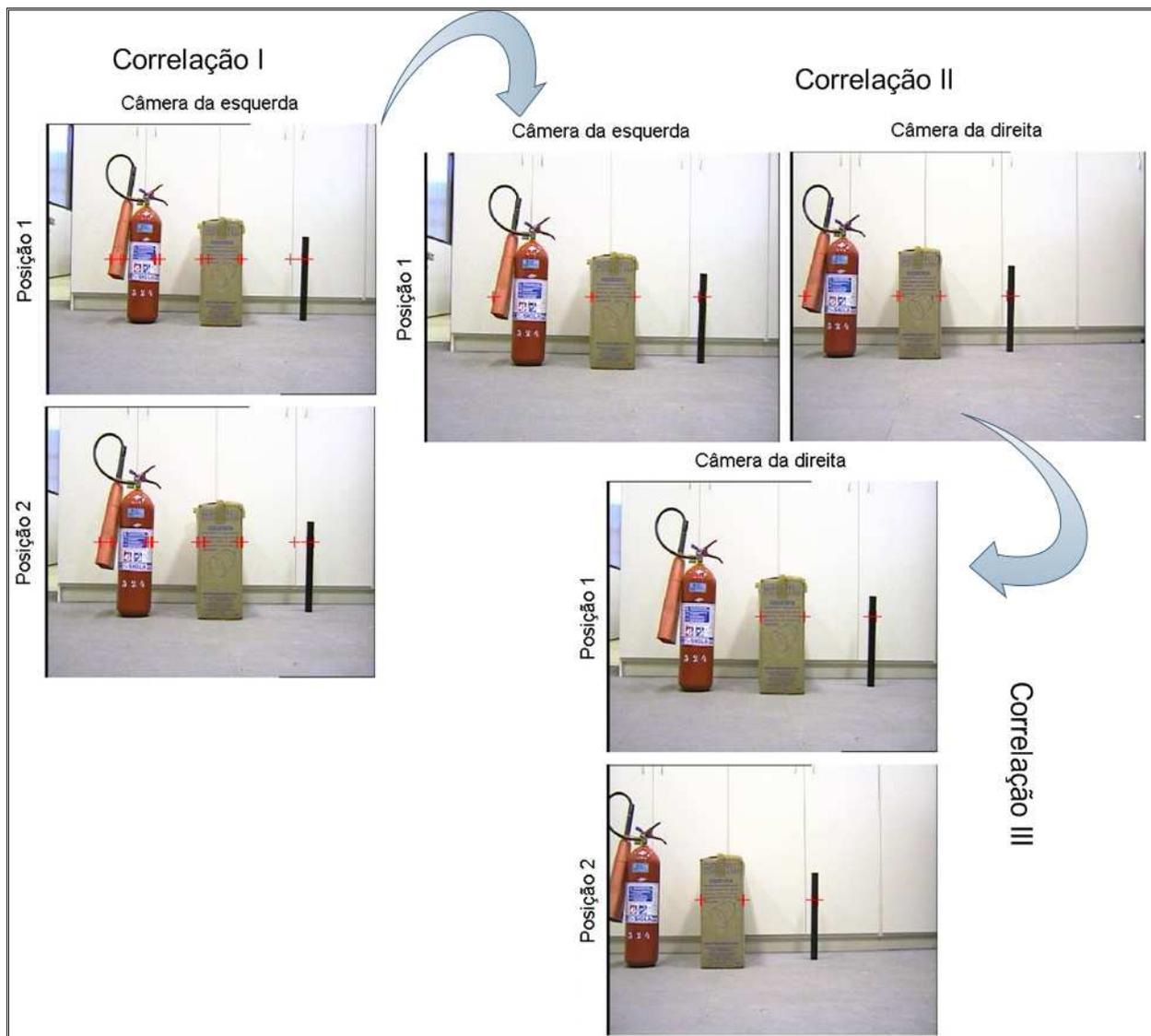


Figura 5.8 Método de correlação passo a passo: Correlação I, correlação entre as imagens “CE_PAN” e “CE_PAT”. Correlação II, correlação entre os pontos obtidos em Correlação I e a imagem “CD_PAN”. Correlação III, correlação entre os pontos obtidos em Correlação II e a imagem “CD_PAT”

Na Figura 5.9 é apresentado o resultado da correlação entre as quatro imagens descritas anteriormente.

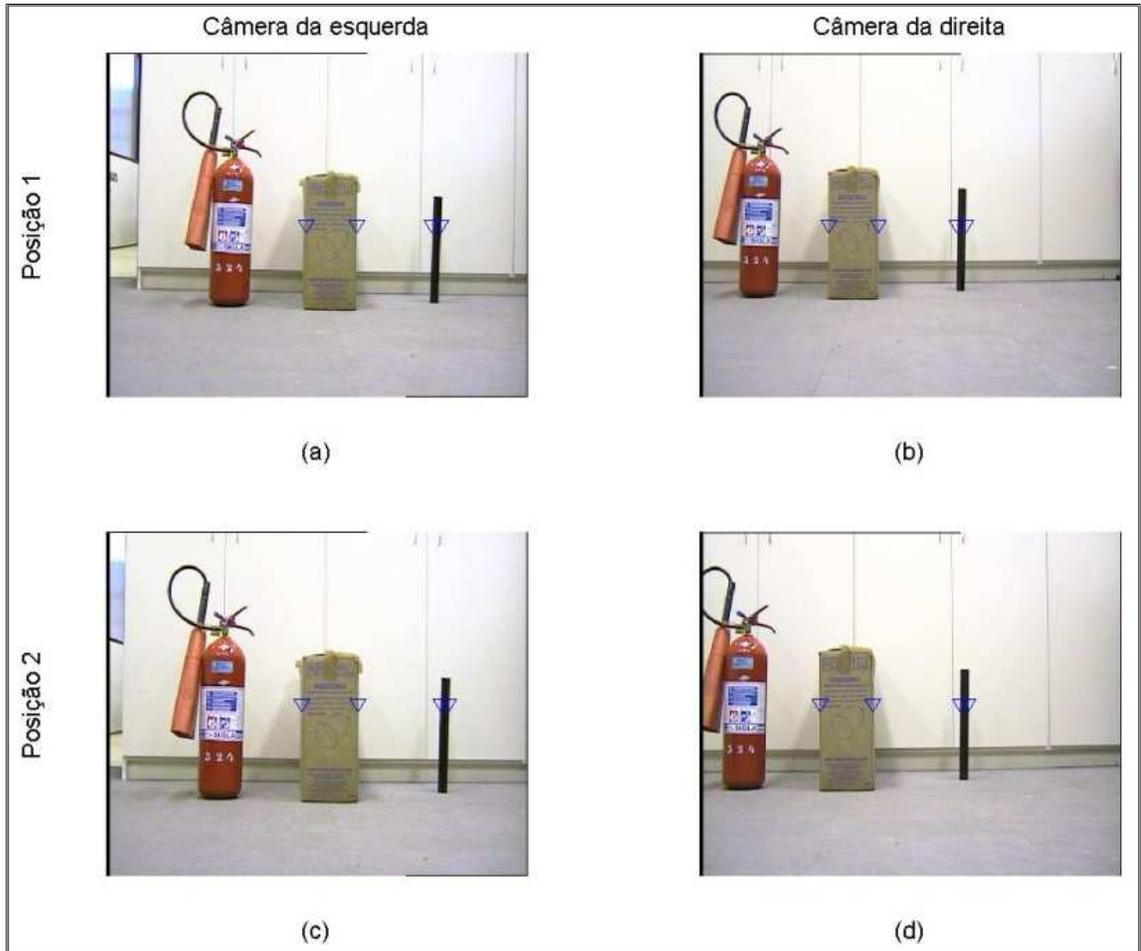


Figura 5.9 Correlação de pontos entre a posição 1 e 2: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

A partir das correlações obtidas anteriormente é estimada a profundidade de cada borda em cada posição. A Figura 5.10 ilustra o ambiente reconstruído à frente do robô.

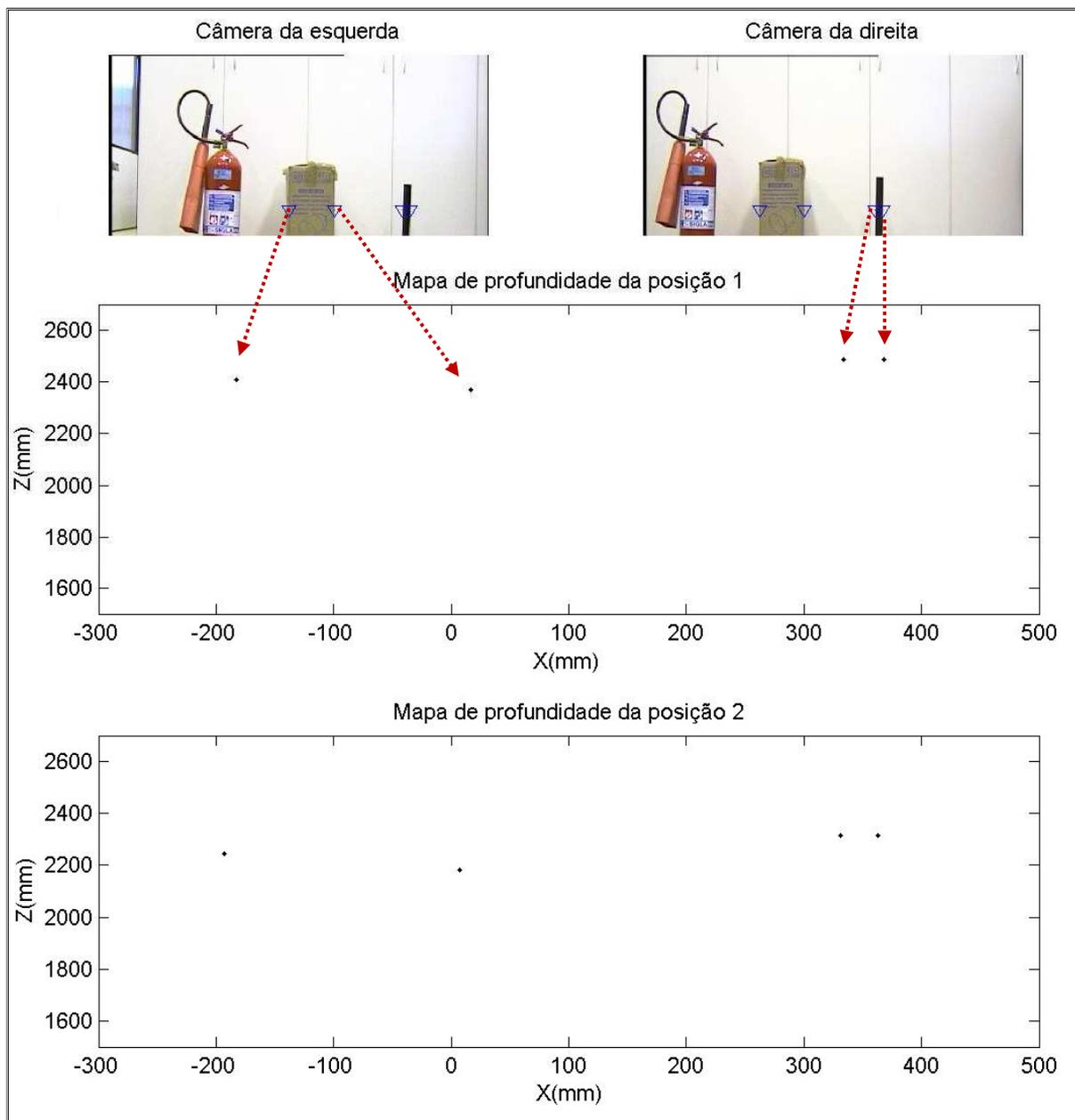


Figura 5.10 Mapa de profundidade de cada posição em vista superior

Os dados de profundidade calculados e reais nas duas posições são mostrados na Tabela a seguir.

Tabela 5.5 Comparação de profundidade

Objeto (posição)	Distância real (mm)	Distância calculada(mm)	Erro (%)
Caixa de papelão (1)	2402	2548	5.7
Caixa de papelão (1)	2402	2501	3.9
Cilindro preto (1)	2690	2648	1.5
Cilindro preto (1)	2696	2652	1.6
Caixa de papelão (2)	2302	2378	3.2
Caixa de papelão (2)	2302	2301	0.005
Cilindro preto (2)	2591	2466	5.06
Cilindro preto (2)	2586	2470	4.6

Analisando-se a média dos valores dos erros em cada posição tem-se 3.23% para o primeiro mapa de profundidade e 3.23% para o segundo mapa de profundidade, apresentando similitude com o ambiente real. Observa-se que a taxa de erro aumentou um pouco com relação as simulações descritas no capítulo 4.

Com os mapas de profundidade é estimada a translação e rotação entre as duas posições. Os resultados de movimento são mostrados na Tabela a seguir.

Tabela 5.6 Comparação do movimento real e estimado

Movimento (mm)	Real	Estimado
X	0	33.37
Z	200	222

A estimativa de movimento é apresentada de forma gráfica na Figura 5.11. O quadro vermelho desta figura representa o movimento real do robô e o quadro preto o seu movimento estimado.

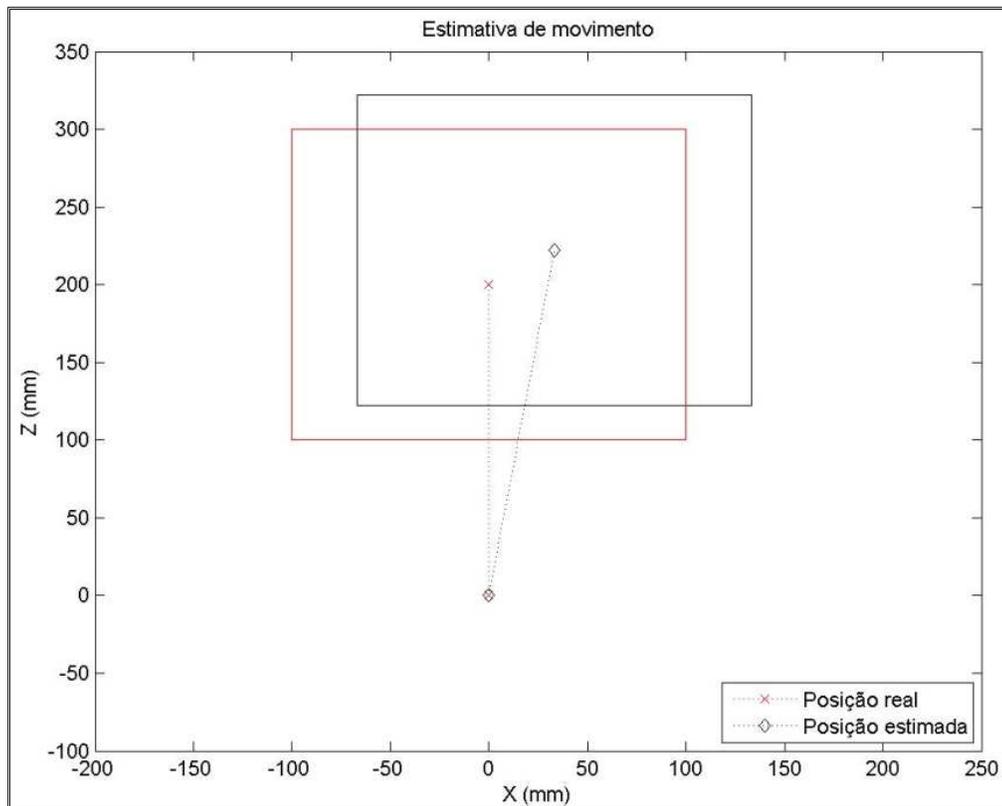


Figura 5.11 Comparação de movimento real e o movimento estimado

Ambiente II:

Neste ambiente o robô percorre o ambiente em direção aos objetos conforme mostrado na Tabela 5.3. Devido ao fato de que o processo é repetitivo, serão apresentadas as imagens correlacionadas e a estimativa de movimento entre cada posição desta tabela, como se ilustra a seguir.

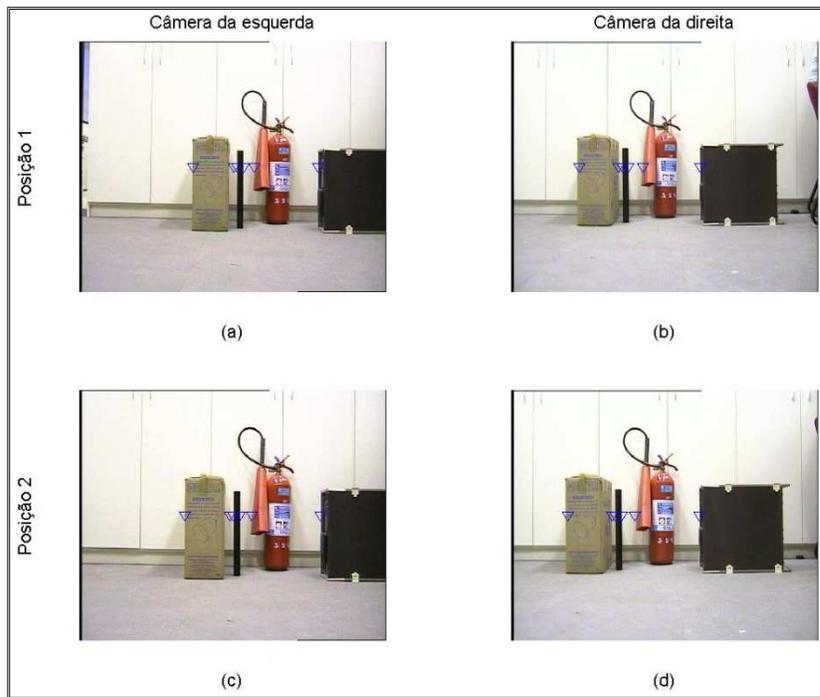


Figura 5.12 Correlação de pontos entre a posição 1 e 2: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT.
(d) CD_PAT

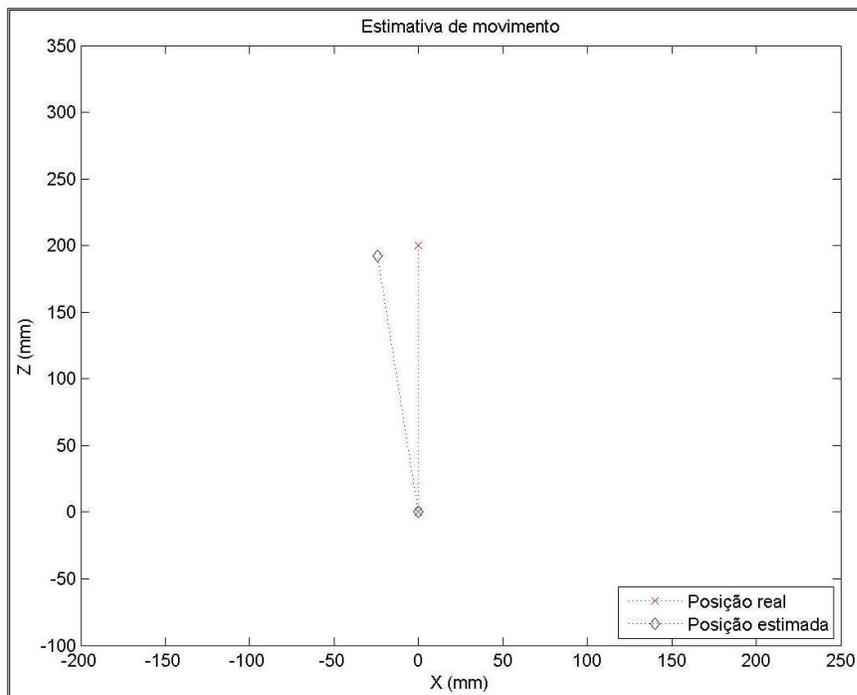


Figura 5.13 Comparação de movimento real e o movimento estimado

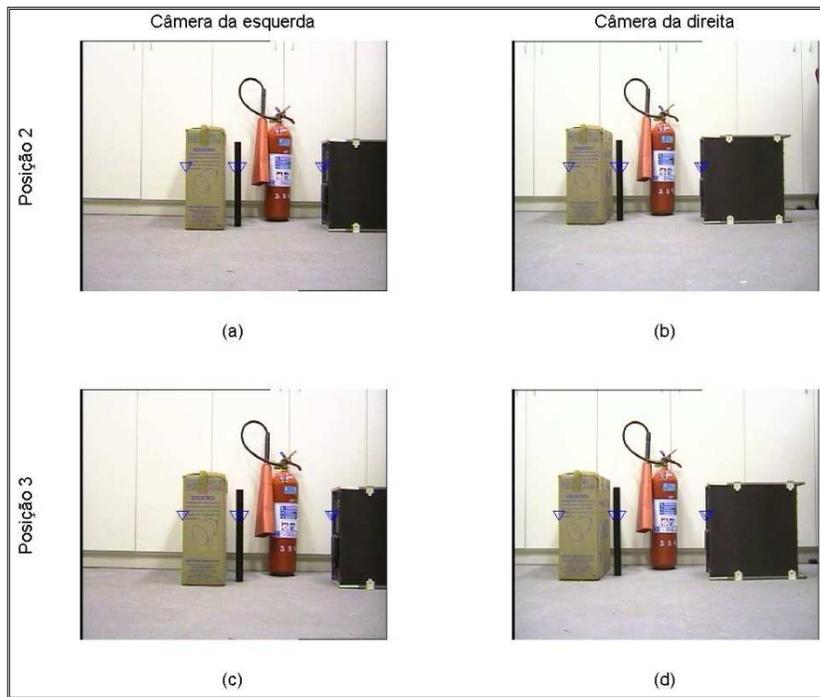


Figura 5.14 Correlação de pontos entre a posição 2 e 3: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT.
(d) CD_PAT

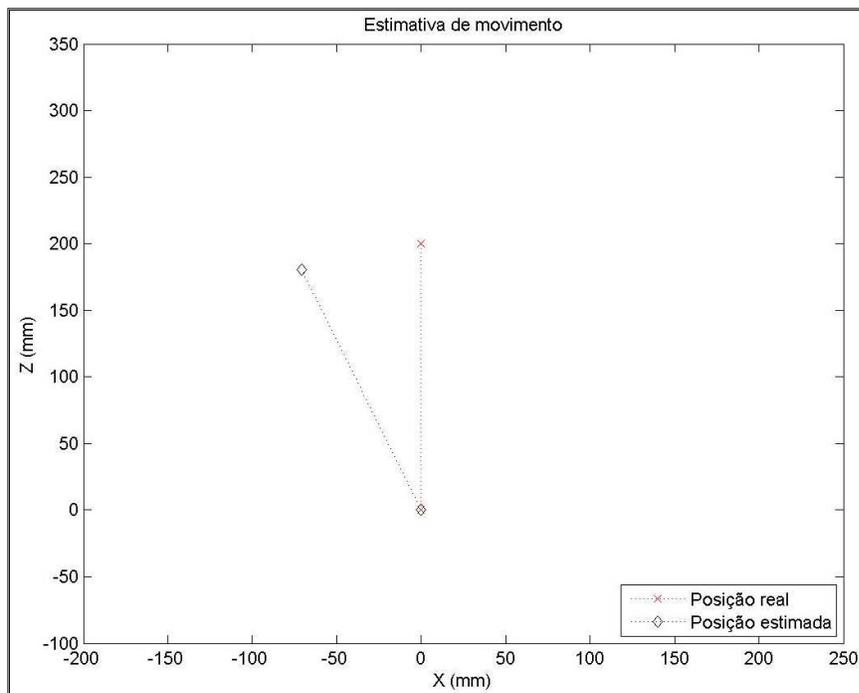


Figura 5.15 Comparação de movimento real e o movimento estimado

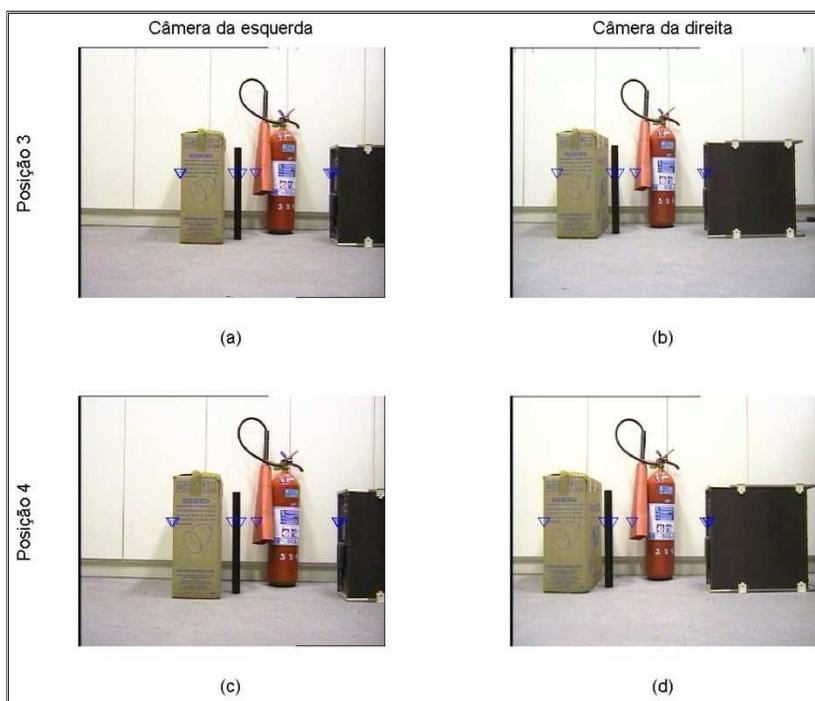


Figura 5.16 Correlação de pontos entre a posição 3 e 4: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

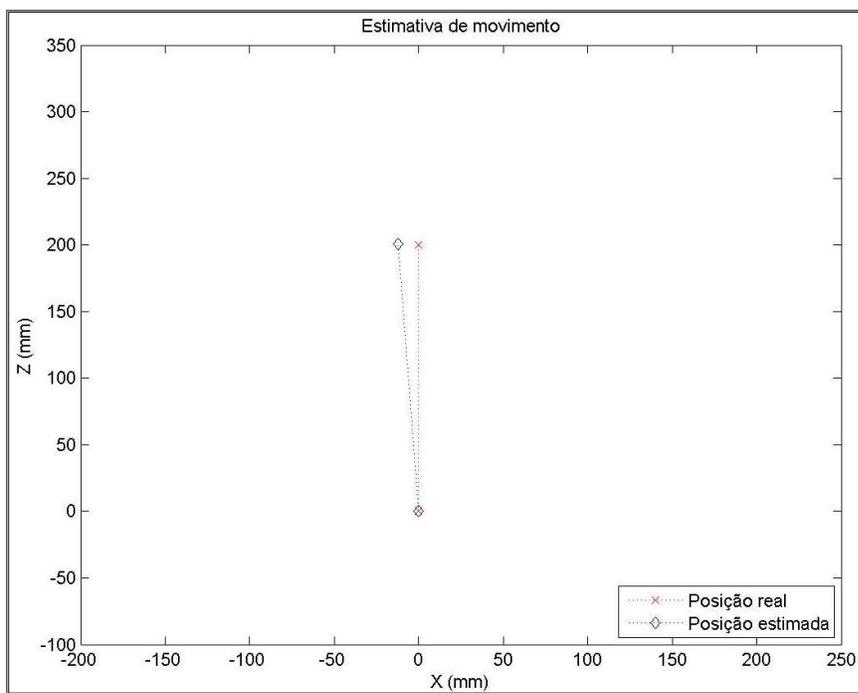


Figura 5.17 Comparação de movimento real e o movimento estimado

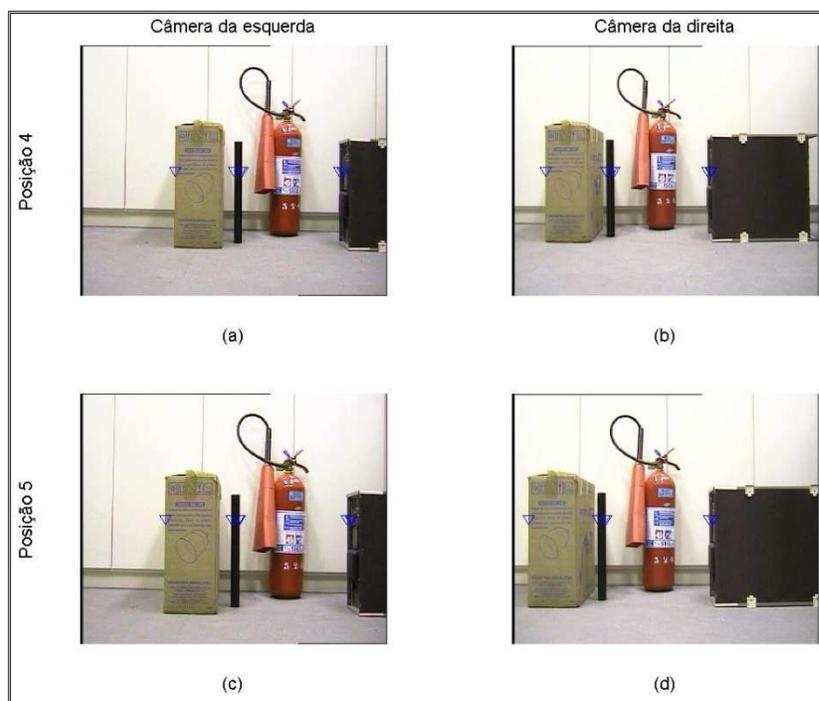


Figura 5.18 Correlação de pontos entre a posição 4 e 5: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT.
(d) CD_PAT

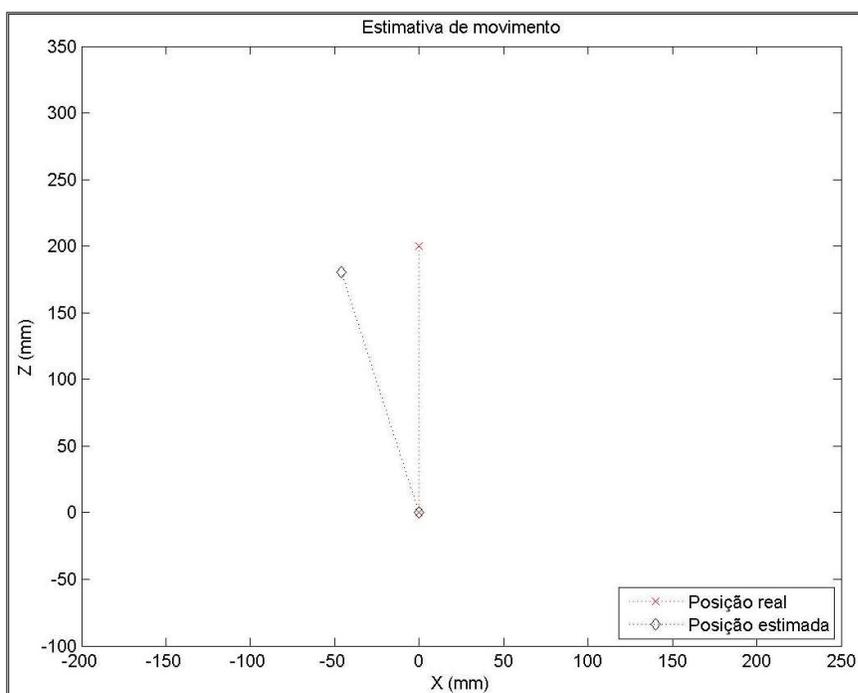


Figura 5.19 Comparação de movimento real e o movimento estimado

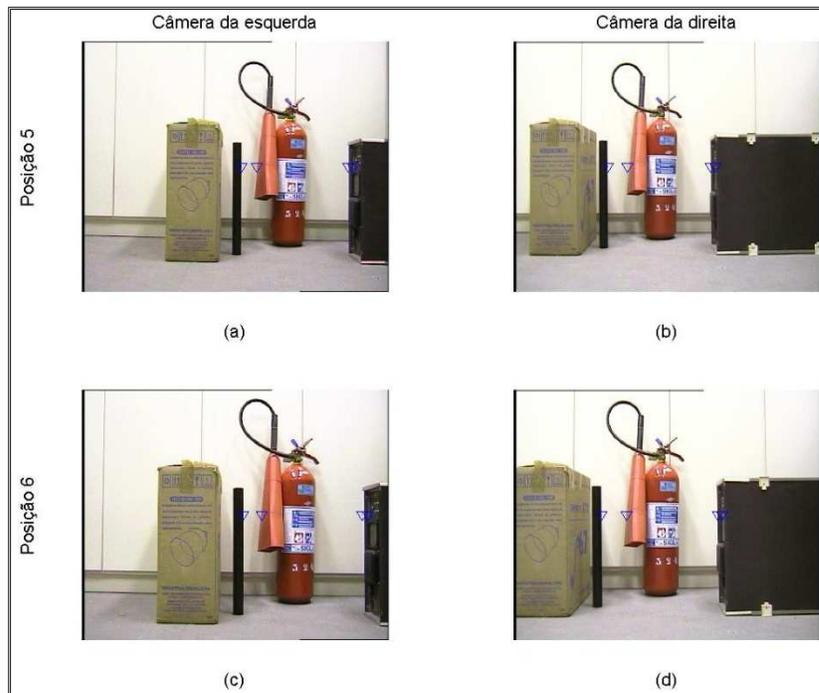


Figura 5.20 Correlação de pontos entre a posição 5 e 6: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT.
(d) CD_PAT

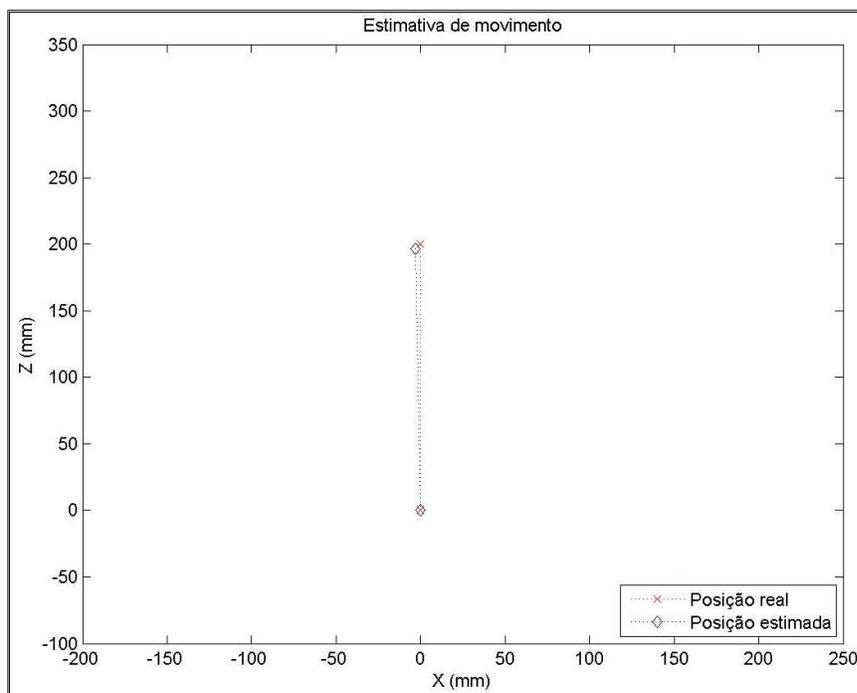


Figura 5.21 Comparação de movimento real e o movimento estimado

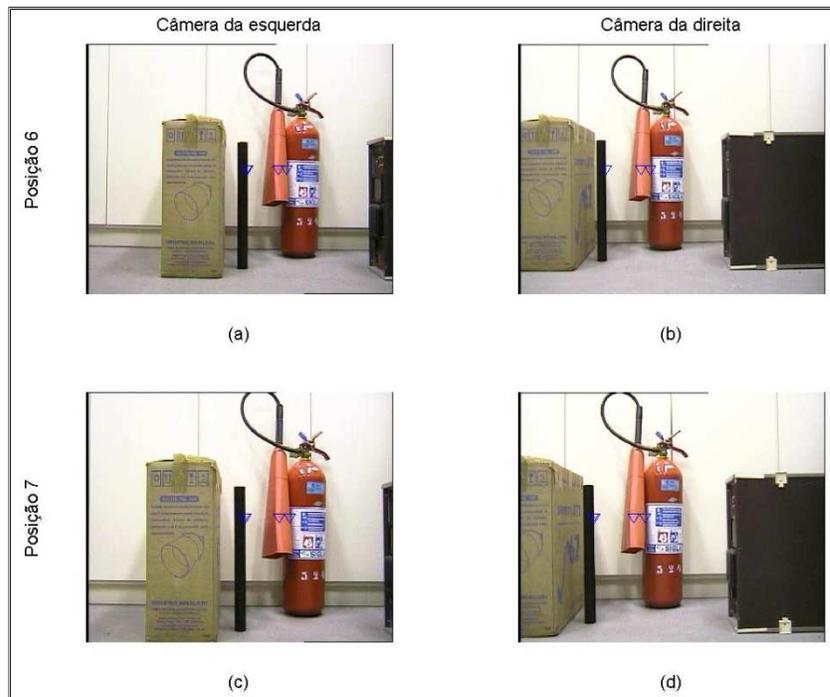


Figura 5.22 Correlação de pontos entre a posição 6 e 7: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT.
(d) CD_PAT

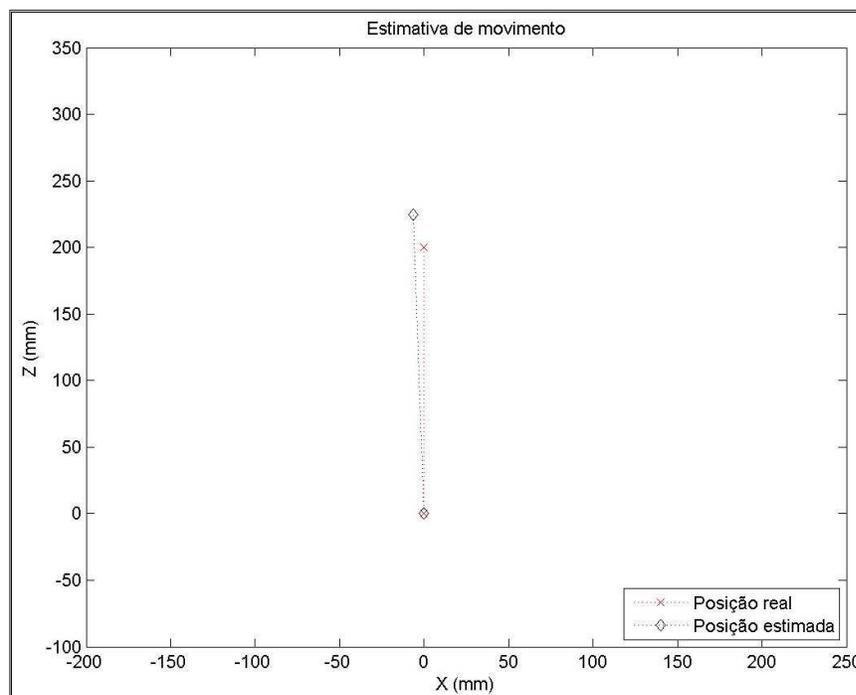


Figura 5.23 Comparação de movimento real e o movimento estimado

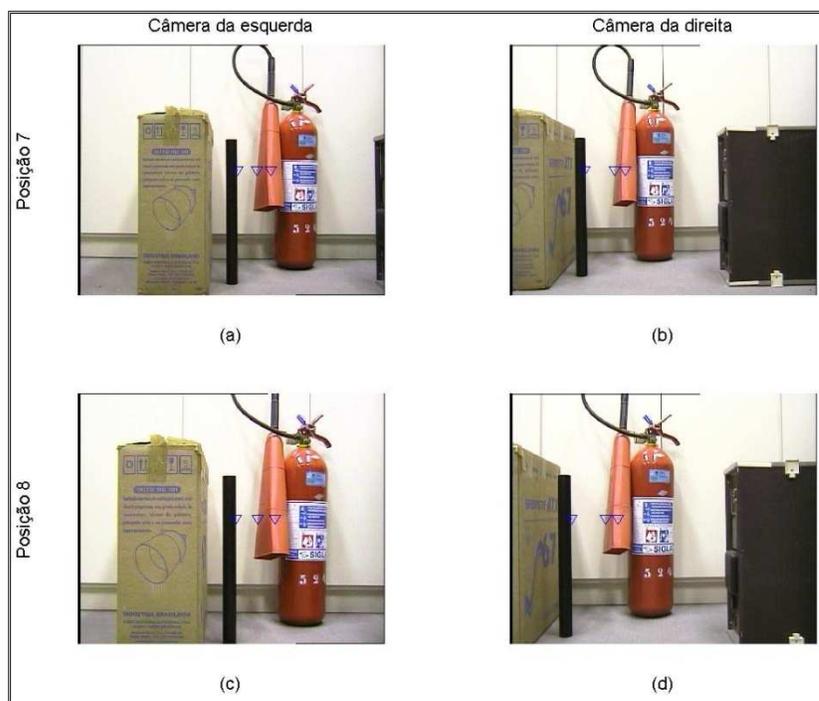


Figura 5.24 Correlação de pontos entre a posição 7 e 8: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT.
(d) CD_PAT

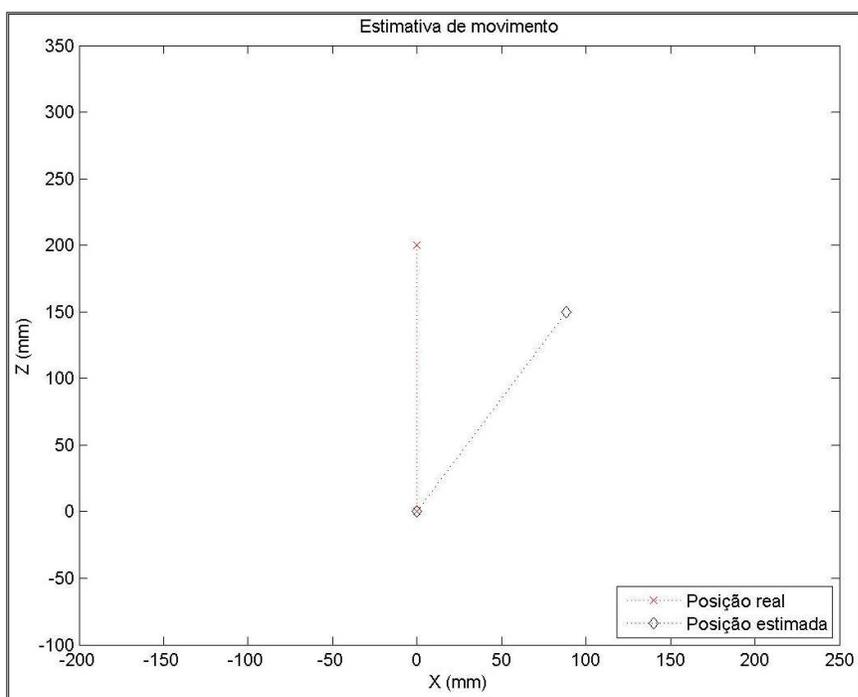


Figura 5.25 Comparação de movimento real e o movimento estimado

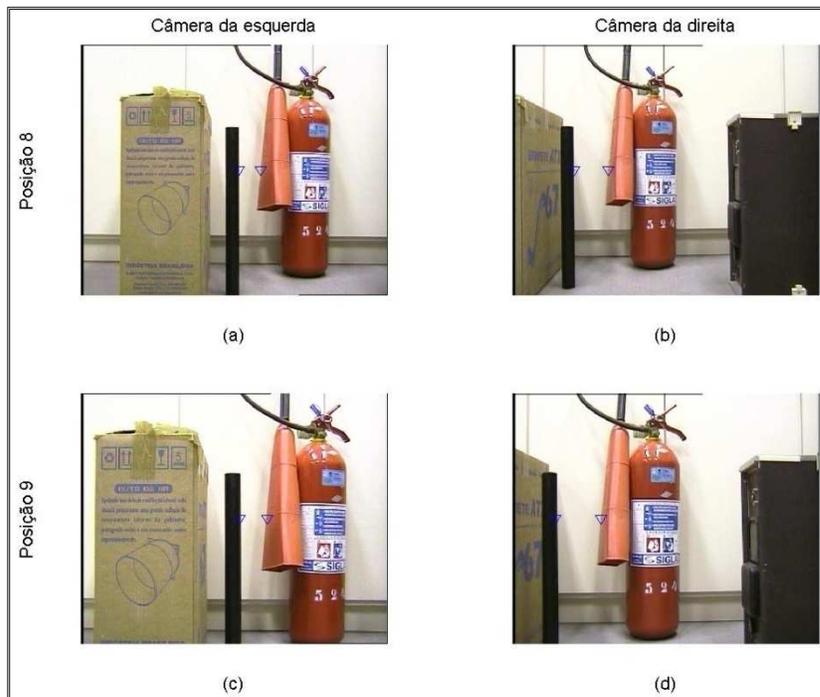


Figura 5.26 Correlação de pontos entre a posição 8 e 9: (a) CE_PAN. (b) CD_PAN. (c) CE_PAT. (d) CD_PAT

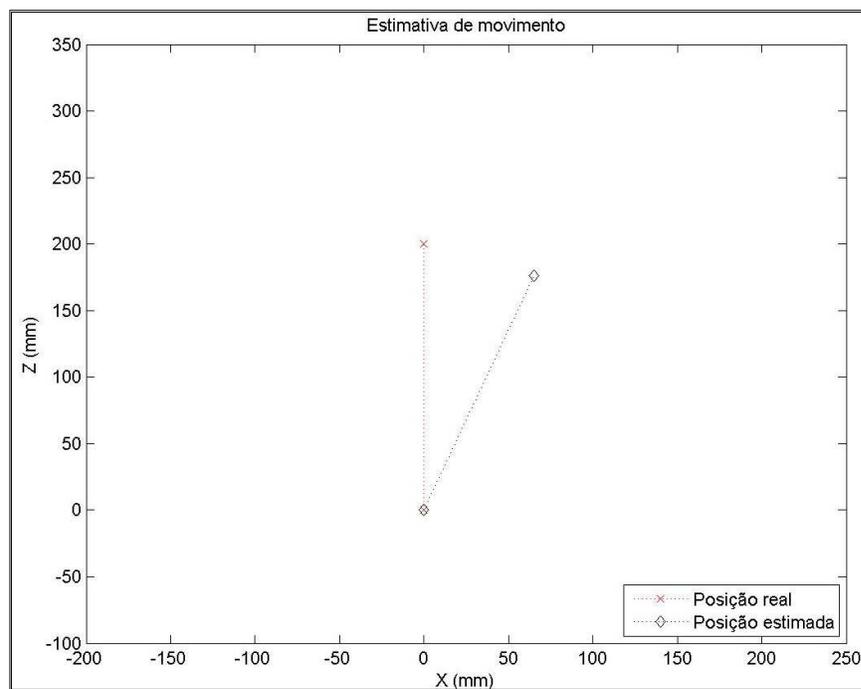


Figura 5.27 Comparação de movimento real e o movimento estimado

A tabela 5.7 apresenta as coordenadas do movimento calculado a partir de duas posições durante toda a trajetória. Na Figura 5.28 é apresentado o percurso do robô. Esta figura é a vista de cima do ambiente. O deslocamento real do robô realizado desde a primeira posição até a última é representado com pontos azuis. O percurso estimado do robô, considerando erros constantes durante a trajetória, é representado com estrelas azuis.

Tabela 5.7 Coordenada da estimativa de movimento

Posição	X (mm)	Z(mm)
1-2	-24.27	192.23
2-3	-70.76	180.54
3-4	-12.44	200.51
4-5	-46.15	180.35
5-6	-3.03	196.25
6-7	-6.62	224.83
7-8	88.42	150.07
8-9	64.94	176.16

Das figuras anteriores e da tabela anterior, observe-se que embora as bordas correlatas entre as imagens sejam correspondentes, a estimativa de movimento tem uma grande variação no eixo X. À medida que o robô se aproxima aos objetos, o erro no eixo Z vai aumentando.

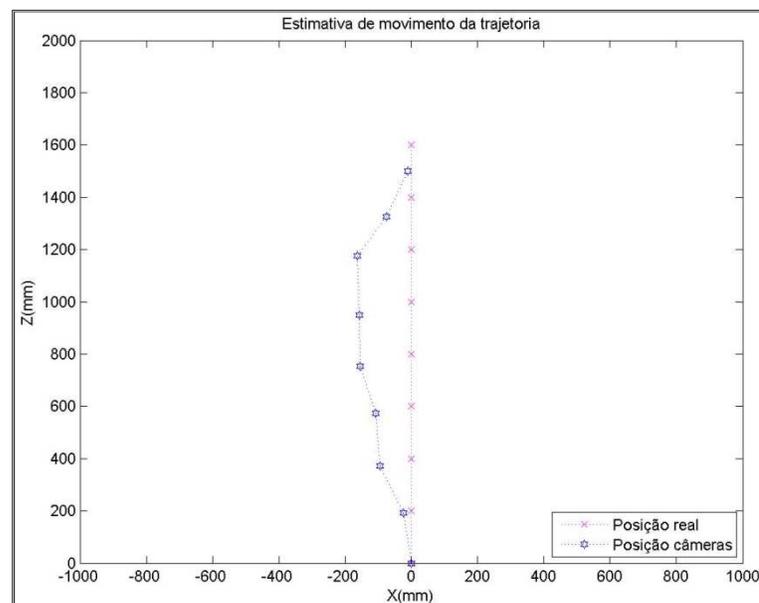


Figura 5.28 Percurso do robô no ambiente

- Análise geral dos experimentos

O método de correlação de imagens unidimensionais foi testado com imagens reais apresentando um desempenho satisfatório, como foi observado na Correlação I da Figura 5.8. A amostragem obtida na Correlação I apresenta uma quantidade considerável de pontos correlatos para reconstruir o ambiente. Um problema surge na busca desses pontos nas outras duas imagens, já que a distância entre as câmeras é relativamente grande, causando variações nas perspectivas das imagens. Apesar disso, são obtidos mais de dois pontos entre os pares de imagens, necessários para aplicar o método de estimativa de movimento.

A partir das correlações, observa-se que a reconstrução do ambiente apresenta erro médio em torno de 3%. Esta reconstrução pode ser considerada útil na navegação de robôs, se o objetivo é navegar desviando de objetos até chegar a um alvo, ou se a fusão sensorial é empregada, através da utilização de outros sensores.

Conforme observado em todos os experimentos anteriores a estimativa de movimento do robô através da análise de pontos da imagem, apresenta sempre algum desvio. Estes desvios são causados pelo desalinhamento das câmeras, distorções das lentes e principalmente pelas incertezas associadas aos procedimentos de correlação de pontos na diferença de imagens. Estes erros, quando somados no gráfico da estimativa da trajetória do robô, propagam-se de forma acumulativa.

6. Conclusões e sugestões para trabalhos futuros

6.1. Conclusões

Os estudos desenvolvidos para utilizar câmeras como sensor na navegação de robôs autônomos, possibilita utilizar correlação de imagens simplificadas à unidimensional como ferramentas de navegação em ambientes de interiores reais. Observa-se que o método de correlação de imagens unidimensionais utilizado para duas imagens, identifica várias características principais nos objetos, embora a amostragem de pontos vá diminuindo à medida que são procuradas essas mesmas características nas outras duas imagens da cena. De outro lado, ao serem triangulados os pontos correlacionados nos pares de imagens, permitem reconstruir ambientes com grande similitude ao ambiente real. Resultados obtidos em ambientes virtuais e reais apresentam um erro médio em torno de 1% e 3% correspondentemente na estimativa de profundidade dos objetos, indicando eficiência e robustez do método de correlação de imagens unidimensional e na determinação do mapa de profundidade das simulações com testes experimentais.

A metodologia proposta neste trabalho permite obter mapas para a navegação robótica, identificando e calculando a profundidade de objetos, para assim estimar o movimento que houve entre duas posições, por meio do processamento das imagens.

Os resultados obtidos na estimativa de movimento bidimensional utilizando a relação epipolar, permitem concluir que os resultados não se ajustam ao esperado, implicando que este método deve ser aperfeiçoado. Embora as técnicas propostas para estimar o movimento bidimensional do robô, contribuem para corrigir a orientação dada pela odometria. Pode-se ainda

integrar por meio de fusão sensorial, os dados oriundos da visão computacional estéreo com aqueles obtidos pela odometria.

A ferramenta matemática Matlab permite processar imagens, extraindo informação do ambiente para obter soluções a problemas reais e facilita a interação da informação dos ambientes virtuais gerados pelo software Blender.

O fato de processar somente o vetor de pixel correspondente à linha horizontal média da imagem permite que o processamento de dados seja rápido e eficiente. Além disso, possibilita a implementação do método de correlação de imagens extraindo mais de dois vetores de pixel da imagem para ter estimativas do ambiente com maior precisão. Neste trabalho foram realizados testes experimentais aproveitando a informação proveniente de 5 linhas horizontais de cada imagem da cena, obtendo resultados similares aos apresentados na estimativa de movimento do capítulo 5.

Os estudos realizados neste trabalho permitem guiar e situar um robô móvel utilizando informação proveniente de duas câmeras. Além disso, os métodos utilizados na navegação autônoma do robô permitem extrair informação da distância dos objetos, informação útil que pode ser aplicada em estratégias de controle em paralelo aos métodos, para obter maior informação com respeito a rotas e caminhos utilizados no movimento do robô para desviar-se dos obstáculos.

6.2. Sugestões para trabalhos futuros

O desenvolvimento deste trabalho possibilita aperfeiçoar e realizar novas aplicações dando continuidade a este trabalho segundo o descrito a seguir.

- Inicialmente sugere-se tratar as distorções das lentes para solucionar o mesmo problema.
- Aperfeiçoar o método de estimativa de movimento ou estudar métodos alternativos para estimar o movimento do robô entre duas posições.
- Realizar um estudo aprofundado, onde seja comparada a qualidade dos dados, robustez dos métodos, erros obtidos, velocidade de processamento, vantagem e desvantagem, custo, etc. Com outros sensores para estimar a viabilidade de utilizar câmeras como sensor.
- Uma próxima aplicação é fundir a informação das câmeras com outro sensor, como os encoders, para complementar a estimação de movimento, introduzindo a teoria de probabilidade para resolver problemas de robótica SLAM.

REFERÊNCIAS

- [1] AXIS 213 PTZ Network Camera User's Manual. [cited 2010 Junho 21]; Available from: http://www.axis.com/es/products/cam_213/index.htm
- [2] Begum, M., Mann, G., and Gosine, R. *Integrated Fuzzy Logic and Genetic Algorithmic Approach for Simultaneous Localization and Mapping of Mobile Robots*. Applied Soft Computing, n. 8, pp. 150-165, 2006.
- [3] Bianchi, R., Simões, A., e Costa, A. *Comportamentos Reativos para Seguir Pistas em um Robô Móvel Guiado por Visão*. Anais SBAI'2001 - Simpósio Brasileiro de Automação Inteligente, Canela, RS, 2001.
- [4] Brian, G., Jain, Kasturi, Ramesh, Rangachar, Schunck. *Machine Vision*. McGraw-Hill, 1995.
- [5] Caballero, F., Merino, L., Ferruz, J., and Ollero, A. *Vision-Based odometry and SLAM for Medium and High Altitude Flying UAVs*. Springer Science, 2008.
- [6] Cardozo, E., Guimarães, E., Rocha, L., Souza, R., Paolieri, F., and Pinho, F. *A Platform for Networked Robotics*. IROS 2010, 2010.
- [7] Clar, M., Ramos R. *A State-space Approach for Measuring Regional Manufacturing Production Indices*. Papers in Regional Science, n. 80, pp. 357-369, 2001.

- [8] Costa, E., Gomes, M., e Bianchi, R. *Um mini Robô Móvel seguidor de Pistas Guiado por Visão Local* Anais do VI Simpósio Brasileiro de Automação Inteligente, Bauru São Paulo: Sociedade Brasileira de Automática, v. 1, pp. 710-715, 2003.
- [9] Davison, A., Kita, N. *3D Simultaneous Localization and Map-Building Using Active Vision for a Robot Moving on Undulation Terrain*. IEEE, pp. 384-391, 2001.
- [10] Diogenes, L. *Uso de Filtro de Kalman e Visão Computacional para a Correção de Incertezas de Navegação de Robôs Autônomos*. 2008.[Doutorado]. Universidade Estadual de Campinas, SP - BRASIL.
- [11] Drews, P., Botelho, S. *Uma Abordagem Baseada em Visão Computacional para Localização e Mapeamento Simultâneos de Robôs Subaquáticos*. Anais do XXVIII Congresso da SBC, 2008.
- [12] Duckett, T., Saffiotti, A. *Building Globally Consistent Gridmaps from Topologies*, in: *Proceedings of the Sixth International IFAC Symposium on Robot Control (SYROCO)*. 2000.
- [13] Durrant-Whyte, H. *Simultaneous Localisation and Mapping (SLAM): The Essential Algorithms*. IEEE, 2006.
- [14] Florczyk, S. *Robot Vision. Video-based Indoor Exploration with Autonomous and Mobile Robots*. WILEY-VCH, 2005. p. 36-45.
- [15] Forsyth, D., Ponce, J. *Computer Vision: a Modern Approach*. Hall, P., 2003. p. 3-11, 234-250.
- [16] Gasper, J. *Visão para Robótica Móvel: Detecção de Obstáculos sobre Pavimento Plano*. 1994.[Mestrado]. Universidade Técnica de Lisboa, Lisboa - PORTUGAL.

- [17] Hebert, M., Kanade, T. *3-D vision for outdoor navigation by an autonomous vehicle*. Proc Image Understanding Workshop, San Mateo, CA, pp. 593-601, 1998.
- [18] Heipke, C. *Automation of Interior, Relative, and Absolute Orientation*. ELSEVIER Photogrammetry & Remote Sensing, 1996.
- [19] Huang, S., Wang, Z., and Dissanayake, G. *Time Optimal Robot Motion Control in Simultaneous Localization and Map Building (SLAM) Problem*. IEEE/RSJ International Conference on Intelligent Robots and System, pp. 3110-3115, 2004.
- [20] Koller, D., Wegbreit, B., Montemerlo, M., and Thrun, S. *FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges*. IJCAI, 2003.
- [21] Koller, D., Wegbreit, B., Montemerlo, M., and Thrun, S. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem*. in Proc AAAI, n. 2, pp. 593-598, 2002.
- [22] Kriegman, D., Triendl, E., and Binford, T. *Stereo Vision and Navigation in Building for Mobile Robots*. IEEE Transactions on Robotics and Automation, v. 5, n. 6, pp. 792-803, 1989.
- [23] Kurka, P., Rudek, M. *Three-Dimensional Volume and Position Recovering Using a Virtual Reference Box*. IEEE Transactions on Image Processing, 2006.
- [24] Longuet-Higgins, H. *The Reconstruction of a Scene from two Projections-Configuration that Defeat the 8-point Algorithm*. Artificial Intelligence Application. Denver, CO; 1984. p. 395-397.
- [25] Manual Blender 2.49. [cited 2009 Outubro 15]; Available from: http://wiki.blender.org/index.php/Main_Page

- [26] Marques, F. *Reconstrução Tridimensional por Visão Estéreo e Detecção de Oclusões*. 2007.[Doutorado]. Universidade Estadual de Campinas, SP - BRASIL.
- [27] MATH WORKS Inc. *MatLab*. [Programa de computador], v. R2009a, 2009.
- [28] Menezes, L. *Processamento de Imagens na Análise Dinâmica de Risers de Produção de Petróleo com Modelo de Escala Reduzida em Ambiente de Laboratório*. 2008.[Doutorado]. Universidade Estadual de Campinas, SP - BRASIL.
- [29] Nogueira, F. *Reconstrução Tridimensional por Visão Estéreo e Detecção de Oclusões*. 2007.[Doutorado]. Universidade Estadual de Campinas, SP - BRASIL.
- [30] Pajares, G., Cruz, J., and Aranda, J. *Stereo Matching Based on the Self-organizing Feature-mapping Algorithm*. *Pattern Recognition Letters*, v. 19, n. 3-4, pp. 319-330, 1998.
- [31] Paz, L. *Divide and Conquer: EKF SLAM in O(n)*. 2008.[Doctoral]. Universidad de Zaragoza, Zaragoza.
- [32] Peter S., M. *Stochastic Models, Estimation and Control*. Press, A., 1999. p. 3-16.
- [33] Pinto, A., Reali, A. *Relaxação do Modelo Oculto de Markov para o Problema de Mapeamento e Localização Simultâneos*. *Anais do XVI congresso Brasileiro de Automática CBA*, 2006.
- [34] Pioneer 3 Operations Manual. [cited 2009 Novembro 9]; Available from: <http://143.106.50.145:8080/Cursos/IA368N/02-09/P3OpMan5.pdf>
- [35] Rabiner, I. *A Tutorial on Hidden Markov Model and Selected Application in Speech Recognition*. *IEEE*, v. 77, n. 2, pp. 257-286, 1989.

- [36] Rafikova, E. *Controle de Robô móvel Via Realimentação de Estados Utilizando Visão Estereoscópica*. 2009.[Doutorado]. Universidade Estadual de Campinas, SP - BRASIL.
- [37] Rapoport, I., and Oshman, Y. *Fault-Tolerant Particle Filtering by Using Interacting Multiple Model-Based Rao-Blackwellization*. Journal of Guidance, Control, and Dynamics, v. 28, n. 6, pp. 1171-1177, 2005.
- [38] Rudek, M. *Método de Posicionamento e Dimensionamento 3D Baseado em Imagens Digitais*. 2006.[Doutorado]. Universidade Estadual de Campinas, SP - BRASIL.
- [39] Thorpe, C., Hebert, M., Kanade, T., and Shafer, S. *Vision and Navigation for the Carnegie-Mellon Navlab*. IEEE Trans Pattern Anal Mach Intell, v. 10, n. 3, pp. 362-373, 1988.
- [40] Thrum, S. *Exploring Artificial Intelligence in the New Millennium*. São Francisco, USA: Thiuri, E., 2003. p. 1-34.
- [41] Tovar, B., Gómez, L. Cid, R., Miranda, M., Monroy, R., and Hutchinson, S. *Planning Exploration Strategies for Simultaneous Localization*. Robotics and Autonomous System, n. 54, pp. 314-331, 2006.
- [42] Turk, M., Morgenthaler, D., Gremabn, K., and Marra, M. *Vision System for Autonomous Land Vehicle Navigation*. IEEE Trans Pattern Anal Mach Intell, v. 3, pp. 342-361, 1988.
- [43] Wang, M., Cen, Y., Hu, X., Yu, X., Xie, N., Wu, Y., Xu, P., and Xu, D. *A Weighting Windows Applied to the Digital Image Correlation Method*. Optics & Laser Technology, n. 41, pp. 154-158, 2009.
- [44] Waxman, A. *A Visual Navigation System for Autonomous Land Vehicles*. IEEE J Robot Autom, v. RA-3, n.º. 2, pp. 124-141, 1987.

[45] Zhang, Z., Deriche, R. *A Robust Technique for Matching two Uncalibrated Image Through the Recovery of the Unknown Epipolar Geometry*. *Artificial Intelligence*, v. 78, n. 1-2, pp. 87-119, 1995.

[46] Zhou, J., Shi, J. *A Robust Algorithm for Feature Point Matching*. *Computers & Graphics*, n. 26, pp. 429-436, 2002.