Carla Silva Rocha Aguiar

ESTUDO DA PLATAFORMA MÓVEL OMNI E CALIBRAÇÃO DE CÂMERA PARA LOCALIZAÇÃO 3-D

Monografia apresentada ao Curso de Engenharia Mecatrônica da Universidade de Brasília como requisito parcial para a obtenção do título de Engenheiro de Mecatrônica.

Orientador: Prof. Geovany Araújo Borges (ENE)

Curso de Engenharia Mecatrônica Departamento de Engenharia Elétrica Faculdade de Tecnologia Universidade de Brasília Março, 2006

Dedico este trabalho aos meus pais Nailza e Valter e a minha madrinha Margarete

Agradecimentos

Agradeço primeiramente a Deus, que me deu força, saúde, ânimo e fé para a conclusão deste trabalho.

Aos meus pais e irmãs, pelo amor, apoio, dedicação, paciência e compreensão demonstrados em todo esses anos de convivência. Esta conquista é de vocês também.

Ao meu orientador, professor Geovany Araújo Borges, pela oportunidade da realização do trabalho, pela orientação, apoio e encorajamento dados durante esta fase, pelo exemplo de excelente professor, engenheiro, por sua ética na condução dos projetos e, mais importante, pela amizade. À você Geovany, meu repeito, gratidão e admiração.

À toda minha família, tios, primos e avós, pelo amor, apoio e incentivo depositados em mim. Em especial, o meu muito obrigada à minha madrinha, Margarete, e meus tios, Carlos e Marcos, pelo grande incentivo que vocês me deram ao longo do curso, minhas irmãs Viviane e Juliana, pela companhia e paciência e, especialmente, à você pai, por ser um entusiasta, por ter acreditado em mim, acompanhado e vibrado com tudo o que eu fazia. Minha gratidão à você e meu amor vão ser eternas.

Um muito obrigada especial à todos os técnicos do Departamento da Engenharia Mecânica (bloco SG-9) e do Departamento da Engenharia Elétrica (bloco SG-11) pela disponibilidade, pela ajuda prestada e pela simpatia. Em especial obrigada Marcão e Celius por tornaram este projeto possível.

À todos os professores, sou grata pelos ensinamentos e pelo exemplo de profissionalismo mostrados ao longo dos anos. Especialmente sou grata aos professores Geovany A. Borges, Alexandre Araújo e Vitor Hugo Casanova, pela confiança depositada no início da minha vida acadêmica e aos professores Flávia Sousa Aranha, Marco Antônio Freitas Egito, pelos ensinamentos.

A todos meus amigos, pela companhia e por tornarem suportável os momentos estressantes e por me ajudarem, cada um a sua maneira. Em especial ao Flávio Vidal, Carlos Cascão, Samuel Júnior, Leandro Cotta, Plínio Luís, Alexandre Martins, Gustavo Moisés, que muito me ajudaram para a realização deste trabalho. Além dos meus amigos da banda *Ruah* e da banda *A caminho Pai*, minhas eternas amigas Ilana, Flávia e Glauce e a todos que de alguma forma me fizeram ser uma pessoa melhor. Um obrigada especial ao Will Miranda, que foi o meu apoio e companheiro nesta última fase.

A Etienne Dombre, diretor do *Département Robotique* do LIRMM, e a Philippe Poignet e Jean Triboulet, meus orientadores, ficam meus agradecimentos pela acolhida e pela oportunidade. Obrigada a todos os membros do LIRMM pela colaboração, amizade e simpatia. Em especial, aos meus amigos Christophe Rabaud, Mickaël Sauvee, Thomas pela paciência e amizade e Maria Augusta, Lúcia e Béatrix Reynaud, por terem sido uma família pra mim.

Resumo

Titulo: Estudo da plataforma móvel Omni e Calibração de câmera para localização 3-D Autor: Carla Silva Rocha Aguiar. Orientador: Geovany Araújo Borges. Curso de Graduação em Engenharia Mecatrônica Brasília, março de 2006

Resumo: O manuscrito, apresentado em duas partes, trata do estudo de uma plataforma móvel real, o robô Omni, e da calibração de câmera de vídeo para a localização 3-D. A primeira parte do trabalho trata do estudo da plataforma móvel Omni, um robô móvel omnidirecional usado para pesquisa em robótica móvel. São apresentados sua estrutura física, seus sensores e sua estrutura de software. São implementados projetos para a locomoção do robô, aquisição de imagens de câmeras em configuração estéreo e estimação do posicionamento, que são executados simultaneamente. Futuramente, estes projetos serão usados para realizar o deslocamento autônomo de tal robô a fim de seguir estruturas (cabos) observadas pelas câmeras.

Na segunda parte é abordado o problema da calibração de câmeras de vídeo. Os parâmetros da câmera são apresentados, assim como a influência do procedimento de calibração sobre a estimação de cada parâmetro. É apresentado também o método de estimação de movimento e reconstrução 3-D, utilizando os parâmetros da câmera obtidos na calibração. Este método é usado para fazer a estimação do movimento cardíaco a partir de imagens da superfície do coração. Resultados mostram as características do movimento cardíaco, tais como amplitude e freqüência, e a influência da respiração no sinal cardíaco.

Abstract

Title: Study of the Omni mobile platform and Camera calibration for the 3-D localization Author: Carla Silva Rocha Aguiar Supervisor: Geovany Araújo Borges. Curso de Graduação em Engenharia Mecatrônica Brasília, march, 2006

Abstract: The manuscript, presented in two parts, deals with the study of a real mobile platform, the Omni robot, and of the calibration of video camera for the 3-D localization. The first part of the work deals with the study of the Omni mobile platform, an omnidirecional mobile robot used for research in mobile robotics. Its physical structure, its sensors and its software structure are presented. Projects for the locomotion of the robot, acquisition of images of cameras in stereo configuration and esteem of the positioning are implemented, which are executed simultaneously. In the future, these projects will be used to carry through the independent displacement of such robot in order to follow structures (handles) observed by the cameras.

At the second part is boarded the problem of the calibration of video cameras. The parameters of the camera are presented, as well as the influence of the procedure of calibration on the esteem of each parameter. It is also presented the method of movement esteem and 3-D reconstruction, using the gotten parameters of the camera in the calibration. This method is used to make the esteem of the cardiac movement through images of the surface of the heart. Results show the characteristics of the cardiac movement, such as amplitude and frequency, and the influence of the breath in the cardiac signal.

Conteúdo

1	Intr	rodução	1
	1.1	Descrição	2
		1.1.1 Parte I	2
		1.1.2 Parte II	2
	1.2	Organização do manuscrito	2
Ι	Est	tudo da plataforma móvel Omni	5
2	Roł	bótica Móvel	7
	2.1	Arquitetura de robôs móveis	7
	2.2	Robôs móveis em ambientes externos	8
	2.3	Robôs móveis em ambientes internos	9
	2.4	Plataforma móvel Omni	10
3	Pla	taforma móvel Omni	13
	3.1	Introdução	13

CONTEÚDO

3.2	Arquit	etura da Plataforma móvel Omni	14
	3.2.1	Motores	15
	3.2.2	Sensores Proprioceptivos	16
	3.2.3	Sensores Exteroceptivos	19
3.3	Arquit	tetura lógica da Plataforma móvel Omni	22
	3.3.1	Características do conjunto Windows2000/RTX 5.0	22
Loc	alizaçã	o do Robô Omni	25
4.1	Introd	ução	25
4.2	Sistem	as de coordenadas da Plataforma móvel Omni	26
	4.2.1	Transformação de coordenadas	26
	4.2.2	Posição da referência do ladar em relação ao sistema de referência do robô	27
	4.2.3	Posição da referência da câmera em relação ao sistema de referência do robô	28
4.3	Model	o Cinemático do robô Omni	29
4.4	Odom	etria	33
4.5	Girode	ometria	34
4.6	Model	o de propagação de erro	35
Ger	ação d	e Trajetória	37
5.1	Introd	ução	37
	5.1.1	Definição geométrico da trajetória	38
	5.1.2	Interpolação da trajetória por curva de Bezier	39
	5.1.3	Lei de velocidade curvilínea	40
Des	envolv	imento e Experimentação	43
6.1	Introd	ução	43
6.2	Geraç	ão de Imagens sintéticas	44
	6.2.1	Estimação do espaço de trabalho 3-D visto pelo sistema de coordenadas da câmera	44
	6.2.2	Movimentação do robô e geração da imagem sintética	45
		······································	- 5
	 3.2 3.3 Loc 4.1 4.2 4.3 4.4 4.5 4.6 Ger 5.1 Des 6.1 6.2 	3.2 $3.2.1$ $3.2.2$ $3.2.3$ 3.3 $Arquit 3.3.1 3.3.1 1.1 1.1 4.2 3.3.1 4.2 3.3.1 4.2 3.3.1 4.1 1.1 4.2 3.3.1 4.2 3.3.1 4.2 3.3.1 4.2 3.3.1 4.2 3.3.1 4.2 3.3.1 4.2 3.3.1 4.2 3.3.1 4.2 3.3.1 4.2 3.3.1 4.2 3.3.1 4.2.1 4.2.3 4.3 Model 4.4 Odom 4.5 Girodo 5.1.1 5.1.2 5.1.3 5.1.3 5.1.3 5.1.3 6.2.1 6.2.1$	3.2 Arquitetura da Plataforma móvel Omni 3.2.1 Motores 3.2.2 Sensores Proprioceptivos 3.2.3 Sensores Exteroceptivos 3.3 Arquitetura lógica da Plataforma móvel Omni 3.3.1 Características do conjunto Windows2000/RTX 5.0 Localização do Robô Omni 4.1 Introdução 4.2 Sistemas de coordenadas da Plataforma móvel Omni 4.2.1 Transformação de coordenadas 4.2.2 Posição da referência do ladar em relação ao sistema de referência do robô 4.2.3 Posição da referência da câmera em relação ao sistema de referência do robô 4.3 Modelo Cinemático do robô Omni 4.4 Odometria 4.5 Girodometria 5.1 Introdução 5.1.1 Definição geométrico da trajetória 5.1.2 Interpolação da trajetória por curva de Bezier 5.1.3 Lei de velocidade curvilínea 5.1.3 Lei de velocidade curvilínea 6.2 Geração de Imagens sintéticas 6.2.1 Extimação do espaço de trabalho 3-D visto pelo sistema de coordenadas da câmera 6.2.2 Movimentação do robô e geração da imagem sintética

		6.3.1 Memória compartilhada		46
		6.3.2 OmniDOS:		47
		6.3.3 VisaoStereo:		47
		6.3.4 OmniRTX:		48
	6.4	4 Simulação da Girodometria com rede neural		49
		6.4.1 Estrutura da Rede Neural		50
		6.4.2 Treinamento da rede		50
		6.4.3 Resultados Experimentais		51
7	Cor	onclusões Parte I		55
II do	Ca o mo	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco	a estimação	57
II do 8	Ca o mo Atu	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica	a estimação	57 59
II do 8	Ca o mo Atu 8.1	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica 1 Robótica médica	a estimação	57 59 59
II do 8	Ca mo Atu 8.1 8.2	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica 1 Robótica médica	a estimação 	57 59 59 61
II do 8	Ca mo Atu 8.1 8.2	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica 1 Robótica médica	a estimação	57 59 61 62
II do 8	C: mo Atu 8.1 8.2 8.3	 Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica 1 Robótica médica	a estimação	57 59 61 62 64
II do 8	Ca mo Atu 8.1 8.2 8.3 Cal	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica 1 Robótica médica	a estimação	 57 59 59 61 62 64 67
II do 8	Ca mo Atu 8.1 8.2 8.3 Cal: 9.1	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica 1 Robótica médica	a estimação	 57 59 61 62 64 67 67
II do 8	Ca Matu 8.1 8.2 8.3 Cal 9.1 9.2	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica 1 Robótica médica	a estimação	 57 59 59 61 62 64 67 67 68
II do 8	Ca Matu 8.1 8.2 8.3 Cal 9.1 9.2 9.3	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica 1 Robótica médica 2 Cirurgias robóticas minimamente invasivas 2 Cirurgias robóticas minimamente invasivas 8.2.1 Cirurgias robóticas minimamente invasivas - compensação do causado pela respiração e batimento cardíaco 3 O projeto MAX 4alibração de câmeras 1 Introdução 2 Parâmetros intrínsecos 3 Parâmetros extrínsecos	a estimação	 57 59 59 61 62 64 67 67 68 70
II do 8	Ca Mtu 8.1 8.2 8.3 Cal 9.1 9.2 9.3 9.4	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica 1 Robótica médica	a estimação	 57 59 59 61 62 64 67 67 68 70 71
II do 8	Ca Matu 8.1 8.2 8.3 Cal 9.1 9.2 9.3 9.4 9.5	Calibração de câmeras para a localização 3-D: Aplicação na novimento cardíaco tualidade em Robótica Médica 1 Robótica médica 2 Cirurgias robóticas minimamente invasivas 8.2.1 Cirurgias robóticas minimamente invasivas - compensação do causado pela respiração e batimento cardíaco 3 O projeto MAX 2 Parâmetros intrínsecos 2 Parâmetros extrínsecos 3 Modelo completo da câmera	a estimação	 57 59 59 61 62 64 67 67 68 70 71 74

deformáveis	77
10.1 Introdução	77

CONTEÚDO

	10.2	Marca	dores passivos	78
	10.3	Localiz	zação 3-D	79
	10.4	Rastre	ando marcadores ao longo do tempo	79
		10.4.1	Método de rastreamento: Correlação	80
11	Free		atos em eslibucião de câmenos e localização 2 D	ດາ
11	ъхр 11-1	Inter d	itos em cambração de cameras e localização 5-D	00
	11.1	And Iter	uçao	83
	11.2	Availa	ção do modelo da camera e do procedimento de cambração $\ldots \ldots \ldots \ldots$	84
		11.2.1	Primeiros resultados em calibração	84
		11.2.2	Influência do modelo de distorção radial e tangencial	86
		11.2.3	Análise do desvio padrão e da estabilidade nos parâmetros estimados	87
		11.2.4	Espaço de trabalho	88
		11.2.5	Avaliação da calibração extrínseca	92
	11.3	Fontes	do erro nas medidas	94
		11.3.1	Padrão de calibração	95
		11.3.2	Iluminação	96
		11.3.3	Foco e Íris	98
		11.3.4	Discussão sobre o procedimento de calibração $\ldots \ldots \ldots \ldots \ldots \ldots \ldots$	98
	11.4	Localiz	zação 3-D	99
		11.4.1	Validação do método	100
		11.4.2	Análise do erro usando o robô manipulador <i>PA10</i> e o sistema de localização <i>Polaris</i>	101
		11 4 3	Estimação do movimento cardíaco	101
		11.1.0		100
12	Con	clusõe	s da Parte II	113
Bil	oliog	rafia		115
An	exos	ł		117
Fu	nçõe	s de a	cesso aos componentes da plataforma Omni	119
	A.1	Assrv.	h: Módulo de controle, independente dos dispositivos de E/S	120

CONTEÚDO

A.2	Pnt.h: Módulo de Gestão do ponto			•	 •	 121
A.3	Est.h: Módulo de Estimação da posição			•	 •	 121
A.4	Gyr. h: Módulo do Girômetro a las er $\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots$			•	 •	 122
A.5	Man.h: Módulo do joystick			•	 •	 122
A.6	Ldr.h: Módulo do Ladar a laser			•	 •	 123
A.7	McrRbt.h: Acesso à placa de $\textit{Entrada/Saída}$			•	 •	 123
A.8	Rbt.h:			•	 •	 124
A.9	Art.h: Leitura dos codificadores ópticos incrementais e absolutos			•	 	 126
A.10	St.h: primitivas relativos ao computador central		•	•	 	 127
A.11	Tra.h			•	 •	 128

Lista de Figuras

2.1	(a) Robô MSL para exploração de Marte e (b) robô submarino TAIPAN para in-	0
		9
2.2	Expemplos de robôs omnidirecionais: (a) com 4 rodas e (b) com 3 rodas	10
2.3	Mapa de ambiente (cinza) e trajetória do robô (azul) (Borges 2002)	11
2.4	Robô Omni	11
3.1	Plataforma móvel Omni e seus componentes.	14
3.2	Arquitetura física do robô Omni.	16
3.3	Estrutura de um girômetro a laser genérico.	19
3.4	Plano de dispersão do ladar a laser	20
3.5	(a-b) Par de imagens estéreo e, (c) configuração das câmeras e com seus respectivos	
	sistemas de coordenadas	22
3.6	Organização dos módulos que compõem o conjunto Windows 2000/RTX	23
41	Sistemas de referência principais do robô Omni: (a) visão superior e (b) visão frontal	27
1.1	Sistemas de referencia principais do robo Omm. (a) visão superior e (b) visão noma.	21
4.2	Configurações possíveis das câmeras no robô Omni	28
4.3	Variáveis (a) da posição absoluta e (b) do modelo cinemático do robô Omni	30

4.4	Configuração da roda descentralizada e orientável.	30
5.1	Interpolação da trajetória [Aragones,2002]	38
5.2	Tipos de de perfis possíveis de velocidade em um segmento da trajetória $\ .\ .\ .$.	41
6.1	Imagem Sintética	46
6.2	Processos que constituem o projeto.	46
6.3	Estrutura da variáveis que armazena os Dados estimados pela Odometria. \ldots .	49
6.4	Mapeamento utilizando rede neural.	49
6.5	Grafo estrutural da rede perceptron multicamadas utilizada para estimar o ângulo de posição do robô móvel.	51
6.6	Trajetórias usadas no treinamento da rede neural. O posicionamento do robô, in- dicado por um quadrado, foi estimado por girodometria. Em cinza claro estão mostrados pontos de imagens do radar a laser do Omni, superpostas também us- ando girodometria.	52
6.7	Trajetória usada na validação da Girodometria com rede neural.	52
6.8	Comparação entre girodometria (vermelho) e odometria usando os parâmetros ge- ométricos iniciais (azul).	53
6.9	Comparação entre girodometria (vermelho) e girodometria com rede neural com: (a) 9 neurônios na camada oculta e (b) 50 neurônios na camada oculta (azul)	53
8.1	Sistema Zeus: a console de comando é composta do robô mestre e da tela de vídeo.	60
8.2	Exemplos de instrumentos cirúrgicos da $EndoWrist^{TM}$	60
8.3	Configuração dos aparelhos em cirurgias minimamente invasivas.	61
8.4	Sistema robótico usado em cirurgias minimamente invasivas	63
9.1	Formação da imagem a partir do modelo <i>pinhole</i>	68
9.2	O plano focal (x, y) é paralelo ao plano da retina (u, v) e a distância entre esses planos é f , no qual f é a distância focal	69
9.3	Mudança de sistemas de coordenadas no espaço 3-D	70
9.4	Efeito da distorção radial e tangencial, respectivamente. Linhas sólidas: sem dis- torção, linhas tracejadas: distorção positiva, linhas pontilhadas: distorção negativa (radial)	74
10.1	Exemplos dos marcadores passivos planares usados	79

LISTA DE FIGURAS

10.2 Princípio da técnica de correlação.	
11.1 Conjunto de imagens usadas para	calibração
11.2 (a) Erro da projeção - cada cor é a bração; (b) detalhe do ponto real(c	associada a uma das imagens consideras na cali- rruz) e ponto estimado (círculo)
11.3 Configuração do padrão de calibrad milímetro: (a) câmera fixa, (b) qua	ção em relação ao plano da imagem, expressa em adro fixo
11.4 Conjunto de imagens bastante dist	orcidas usadas para calibração 8
11.5 Efeito da distorção sobre cada <i>pix</i> na imagem, (b)efeito da distorção	el da imagem: (a)efeito do modelo da distorção tangencial e (c) efeito da distorção radial
11.6 Influência do número de imagens dos parâmetros e a estabilidade de	usados na calibração sobre os valores estimados tais valores: Parâmetros estimados 9
11.7 Influência do número de imagens dos parâmetros e a estabilidade de estimativas	usados na calibração sobre os valores estimados tais valores: desvio padrão relativo associado as
11.8 Rohô manipulador <i>P410</i>	
 11.9 Efeito do espaço de trabalho de 3, camento relativo estimado e real re trabalho definido na calibração e o entre o deslocamento relativo real o 	5 <i>cm</i> na estimação do deslocamento. (a) Deslo- ealizado em duas configurações. (b) o espaço de o definido no movimento de translação. (c) erro e o estimado.
11.10Efeito do espaço de trabalho de 4 camento relativo estimado e real re trabalho definido na calibração e o entre o deslocamento relativo real	<i>cm</i> na estimação do deslocamento. (a) Deslo- ealizado em duas configurações. (b) o espaço de o definido no movimento de translação. (c) erro e o estimado
11.11Conjunto de imagens usadas para	calibração
11.12Resultados da calibração extrínseca padrão de calibração; (b) erro entr	: (a) deslocamento real e estimado realizado pelo e o deslocamento relativo real e o estimado 9
11.13Exemplos de padrões de calibração	9
 11.14Efeito da iluminação: Vista do pa servado nos eixos x e y: (a) superfí centrado em zero e não tem efeito não paralelas, (d) o erro é menos pr 	drão de calibração e do erro correspondente ob- cie do padrão e da câmera paralelas, (b) o erro é <i>"fall-off"</i> ; (c) Superfície do padrão e da câmera róximo ao centro da imagem e cresce radialmente
quando se aproxima das bordas	
11.15Configuração do plano do padrão e	e da imagem no processo de calibração 9
11.16 Imagem usada para estimação dos	parâmetros extrínsecos

11.17(a) Estimação de movimento usado dois sistemas distintos e o deslocamento relativo real; (b) erro entre o deslocamento relativo real e o estimado	101
11.18Configuração experimental: câmera de vídeo e o sistema de localização NDI Polaris.	102
11.19Sistema de localização <i>NDI Polaris.</i>	102
11.20Campo de visão do sistema de localização NDI Polaris.	103
11.21Medição da posição:(a) Marcadores ativos emitem luz infravermelho que é recebido pelo sensor de posição para estimar a posição da ferramenta . (b) Marcadores passivos refletem luz infravermelho para o sensor de posição, e este estima assim a posição da ferramenta	104
11.22(a) Configuração experimental: padrão de calibração com 15 mm de lado.(b) Deslo- camento relativo estimado. Estimação do erro em (c) milímetros e (d) pixel. Esti- mação 3-D do posicionamento e intervalo de erro	108
11.23Experimento <i>in vivo</i> : (a) Configuração do sistema, (b) padrão de calibração colocado na superfície do coração	109
11.24Resultados da calibração: (a) Distância focal, (b) ponto principal e (c) coeficiente de distorção radial	109
11.25Estimação do movimento da superfície do coração de um porco: (a) padrão de calibração usado; (b) Movimento cardíaco estimado; (c) erro sobre a estimativa, em <i>pixel.</i>	110
11.26(a) Estimação do movimento cardíaco no domínio do tempo. (b) Espectro e ampli- tude para baixas freqüências.	111

Lista de Tabelas

3.1	Tensão de alimentação dos componentes da plataforma móvel Omni	15
3.2	Parâmetros do radar a las er, fornecido pelo fabricante IBEO Las ertechnik $\ .\ .\ .$	21
6.1	Parâmetros intrínsicos estimados da câmera <i>Fire-i400</i>	44
11.1	Parâmetros intrínse cos estimado a partir do conjunto de imagens da Figura 11.1 . $% \left({{{\bf{n}}_{\rm{B}}}} \right)$	85
11.2	Parâmetros intrínse cos estimado com o espaço de trabalho de 3,5 $cm.$ \ldots	89
11.3	Parâmetros intrínse cos estimado com o espaço de trabalho de 4 $cm.$ \ldots \ldots \ldots \ldots	89
11.4	Parâmetros intrínse cos estimado a partir do conjunto de imagens da Figura 11.11. . \ensuremath{R}	92
11.5	Resultados da calibração: efeito do ajuste do foco na estimativa de parâmetros. $\ .$	98
11.6	Estimação dos parâmetros extrínsecos.	100
11.7	Parâmetros intrínse cos estimados para o experimento usando o sistema ${\it Polaris.}$	104
11.8	Parâmetros intrínsecos estimados e usados para estimação do movimento cardíaco (Distância de 48 <i>cm</i> entre a câmera e o padrão de calibração)	106

capítulo 1

Introdução

Os trabalhos apresentados aqui parecem ser, à primeira vista, bastante diferentes: na primeira parte é feito o estudo da plataforma móvel Omni, no contexto da robótica móvel, e a segunda está inserida no contexto de robótica médica, com o objetivo de estimar o movimento cardíaco a partir de imagens de uma câmera calibrada. Porém, ao olhar mais atentamente nos dois trabalhos, pode-se traçar diversos paralelos: em ambos, há grande ênfase no modelamento dos objetos em estudo e no estudo dos parâmetros que o descrevem. Todas as técnicas apresentadas sobre estimação de posição 3-D a partir de imagens 2-D, apresentadas na segunda parte do trabalho, podem ser aproveitadas em robótica móvel para fazer com que a plataforma móvel siga um dado objeto de interesse observado pela câmera.

A primeira parte deste manuscrito se refere ao estudo da plataforma móvel Omni, realizado entre os meses de agosto e dezembro de 2005. O principal objetivo deste trabalho foi estudar esta plataforma, utilizar suas bibliotecas para implementar projetos e realizar simulações de projetos que possam ser implementados em tal plataforma.

Já a segunda parte deste trabalho foi realizado durante um estágio no LIRMM (*Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier*), ligado à faculdade de Montpellier II, na França, entre os meses de Agosto de 2004 e Fevereiro de 2005, estágio possibilitado graças à um acordo entre este laboratório e o LCVC (*Laboratório de Controle e Visão por Computa*dor), na UnB. Neste período, o autor participou das atividades do grupo de pesquisa em robótica médica, na área de visão computacional aplicada ao problema de estimação de movimento cardíaco através de imagens de vídeo.

1.1 Descrição

1.1.1 Parte I

A primeira parte deste trabalho foi apresentada um estudo sobre a plataforma móvel Omni. Esta plataforma foi construída para fins de pesquisa em robótica móvel e existe pouca documentação sobre sua estrutura e sobre os membros que o compõem, por isso a necessidade de documentálo. Serão apresentados os sensores que compõem a plataforma Omni e qual o seu princípio de funcionamento. Em seguida será apresentado o modelo cinemático do robô Omni, assim como a Odometria e a Girodometria.

Experimentos devem ser realizados para que sejam testados a funcionalidade da plataforma Omni, o funcionamento dos sensores e a validação das bibliotecas anteriormente implementadas.

1.1.2 Parte II

Na segunda parte deste trabalho foi apresentado um estudo sobre calibração da câmeras de vídeo a fim de se obter informações 3-D a partir de uma única imagem. Foram definidos os parâmetros que descrevem o funcionamento câmera, os fatores externos que influenciam a estimativa, e qual é a ordem de grandeza dos erros associados à estimativa. A partir destes fatores, é observado o melhor procedimento para realizar a aquisição das imagens para a calibração e todas as considerações devem se basear em dados de resultados experimentais.

É proposta também uma técnica para se extrair informações 3-D a partir de imagens 2-D e para rastrear os pontos de controle em diversas imagens. Com isso é proposta a concepção de um sistema que faça o rastreio dos pontos de controle e estime a posição do objeto desejado em relação à câmera, a partir dos dados obtidos na calibração. Com este sistema, é feito a estimativa do movimento cardíaco em tempo real a partir de imagens adquiridas de uma câmera de vídeo calibrada.

1.2 Organização do manuscrito

Começa-se o manuscrito com uma revisão sobre a arquitetura de plataformas móveis, no Capítulo 2, onde são estudadas as diversas configurações de robôs móveis, assim como a classificação do robô de acordo com sua funcionalidade e estrutura.

Em seguida será apresentado no Capítulo 3 a plataforma móvel Omni, um robô omnidirecional

com três rodas. É detalhado neste capítulo cada sensor que compõe a plataforma e as principais características da arquitetura lógica.

O modelo cinemático do robô Omni é apresentado no Capítulo 4. É exposto também a Odometria e a Girodomotria, técnicas de estimação de posição baseadas no modelo cinemático. São apresentados os sistemas de referências locais na plataforma e a transformação entre os diversos sistemas. No Capítulo 5 é mostrado como é feito a criação da trajetória a partir da curva de Bezier quando o robô está em uma missão autônoma.

Finalmente, no Capítulo 6 são abordados os experimentos realizados e no Capítulo 7 são apresentadas as conclusões da primeira parte.

A segunda parte começa com a apresentação do estado-da-arte em robótica médica, no Capítulo 8, onde serão descritas as características e aplicações dos principais tipos de sistemas robóticos usados em aplicações médicas. Serão mostradas as soluções de mercado existentes e as principais pesquisas na área.

Em seguida é apresentado o modelo da câmera e o princípio da formação da imagem, no Capítulo 9. São expostos os parâmetros que descrevem as características físicas da câmera, que são chamados de parâmetros intrínsecos, e os parâmetros que permitem obter informações do sistema de coordenadas da câmera em relação ao sistema de coordenadas do objeto observado, chamados de parâmetros extrínsecos.

É então apresentada a técnica de rastreio utilizada para localizar os pontos de controle em diversas imagens diferentes e a técnica usada para extrair informações 3-D a partir de uma única imagem 2-D, dos pontos de controle e dos parâmetros da câmera, no Capítulo 10.

Por fim, o Capítulo 11 é reservado para mostrar os resultados experimentais. Ao fim deste capítulo é apresentado os resultados obtidos do experimento *in vivo*, no qual foi estimado o batimento cardíaco de um porco a partir de imagens de uma câmera calibrada. As conclusões e perspectivas deste trabalho são apresentadas no Capítulo 12.

Parte I

Estudo da plataforma móvel Omni

capítulo 2

Robótica Móvel

2.1 Arquitetura de robôs móveis

Diferentemente de robôs manipuladores industriais, que evoluíram em ambientes protegidos e estruturados em linhas de montagem, os robôs móveis são conhecidos por intervir em ambientes que mudam de configuração continuamente, devida particularmente à presença humana. Sua finalidade é ter autonomia suficiente para realizar a missão sem que haja intervenção humana.

A robótica móvel tem por objetivo criar veículos autônomos que sejam capazes de tomarem decisões, a partir do conhecimento do ambiente no qual se encontram. Isso só é possível se as plataformas de tais robôs possuírem capacidade operacional, que é resultado da concepção eletromecânica e do comando do robô.

O conjunto composto pela estrutura eletromecânica e pelos algoritmos de controle e de tomada de decisão de um dado robô móvel define sua arquitetura. Estas características dependem principalmente do tipo de ambiente no qual o robô está imerso, seja estruturados ou não estruturados, seja internos ou externos, e do tipo de tarefa tal robô deve executar.

A definição de ambiente estruturado e não estruturado é muito importante na elaboração de algoritmos para a construção de mapas do ambiente no qual o robô está imerso, além de ser um fator importante no projeto da estrutura mecânica do mesmo. O ambiente estruturado é aquele no qual existem restrições geométricas que podem ser utilizados na construção de modelos, tais

como retas e planos. No caso do não estruturado, não existem restrições geométricas que possam ser parametrizadas, ou utilizadas, para construção de mapas. Além disso, a estrutura mecânica deve ser bastante flexível para permitir ao robô se deslocar em ambientes a priori desconhecidos.

Um outro tipo de classificação de ambiente é se este é interno ou externo, podendo ser estruturado ou não. Ambiente interno corresponde a salas, corredores, ou seja, lugares que se pode obter modelos. Já os ambientes externos não podem ser modelados ou parametrizados, tendo como exemplos regiões ao ar livre e o fundo do mar.

2.2 Robôs móveis em ambientes externos

Como já mencionado na seção 2.1, os ambientes externos são definidos como lugares que não se pode modelar. Robôs móveis neste tipo de ambiente tendem a se deslocarem por grandes distâncias realizando tarefas relativamente complexas em terrenos irregulares, com obstáculos e podem ser inclinados em certos pontos. O sistema de localização é normalmente formado por estimativas absolutas com incerteza associada altas, uma vez que estimativas locais, a partir de leitura de codificadores ópticos são inviáveis devido à natureza do terreno, que proporciona deslizamento das rodas, causando leituras erradas nos sensores. Sensores típicos utilizados na localização e no mapeamento local do ambiente são GPS (Global Positioning System), câmeras de vídeo e sensores de ultra-som.

A estrutura física de robôs móveis para ambientes externos é bastante complexa. Estes robôs tem que possuir muitos graus de liberdade (normalmente mais de três graus de liberdade) para permitir uma boa mobilidade em terrenos irregulares e para realizar tarefas complexas em tais terrenos. Além disso, sua carenagem deve ser robusta e vedada, devido à natureza agressiva de, devido a condições climáticas diversas, por exemplo.

Quanto aos algoritmos de controle, estes fazem tanto a fusão entre os diversos sensores para a estimação de posicionamento e controle da trajetória, quanto criar mapa local do ambiente e gerar novos pontos de referência da trajetória a fim de desviar de obstáculos. A parte de comunicação entre o robô móvel e a base para supervisionamento está também presente no algoritmo de controle.

Um ambiente externo ainda pode ser classificado como estruturado e não estruturado. Como exemplo de navegação em ambientes externos estruturados, pode-se citar a navegação de carros autônomos que navegam em estradas. Como exemplo de navegação em ambientes externos não estruturados, tem-se o robô MSL ¹ desenvolvido pela NASA para a exploração do planeta Marte (figura 2.1a) e o protótipo de veículo autônomo submarino TAIPAN ², desenvolvido no LIRMM para inspecionar tubulações de petróleo no fundo do mar (figura 2.1b).

¹http://www.nasa.gov

²http://www.lirmm.fr



Figura 2.1: (a) Robô MSL para exploração de Marte e (b) robô submarino TAIPAN para inspeção de tubulações de petróleo.

2.3 Robôs móveis em ambientes internos

Ambientes internos, como salas, corredores e laboratórios, podem ser considerados como estruturados. Em sua maioria há retas e planos, que podem ser identificados para criação de um modelo representativo do ambiente. Robôs que navegam em tais ambientes geralmente percorrem pequenas distâncias em ambientes que possuem o terreno regular e com nenhuma ou pouca inclinação. Tarefas em tais ambientes tendem a ser mais minuciosas e requerem uma grande precisão no deslocamento do robô, uma vez que este navega em pequenos espaços, tais como corredores, e que mudam continuamente, devido principalmente à intervenção humana. Por isso, o sistema de localização deve ser confiável e preciso, e pode ser auxiliado por meio da construção de mapas geométricos da região ou sensores que detectam obstáculos.

Quanto à estrutura mecânica de plataformas móveis que navegam em ambientes internos, esta tem que proporcionar mobilidade total no plano. Geralmente dois graus de liberdade são suficientes para navegar no plano bidimensional. Todavia, os robôs omnidirecionais, com três graus de liberdade, são largamente utilizados em tais ambientes uma vez que estes podem se mover em qualquer direção sem que seja necessário mudar a direção das rodas. Vários tipos de robôs omnidirecionais já foram propostos com diversos tipos e número de rodas em diferentes configurações. Pode-se citar os robôs omnidirecional com rodas universais, rodas não-centradas, e rodas holonômicas, ou suecas, entre as mais populares. A Figura 2.2 mostra alguns robôs móveis omnidirecionais em configurações de rodas distintas.

Os sensores que compõem plataformas móveis que navegam e ambientes internos auxiliam tanto no sistema de localização relativo e absoluto quanto na construção de mapas de ambiente. Dentre os sensores utilizados no sistema de localização relativa estão os codificadores ópticos incrementais



Figura 2.2: Expemplos de robôs omnidirecionais: (a) com 4 rodas e (b) com 3 rodas.

e absolutos, que para este tipo de aplicação devem possuir uma alta precisão em suas leituras, os girômetros, sensores que detectam a mundança de postura (orientação) da plataforma, e acelerômetros. Já no sistema de localização absoluta, pode-se usar sensores adicionais como leitores de balisas (não muito utilizados atualmente) e câmeras de vídeo. Na construção de mapas de ambientes, os dados providos dos sensores já citados acima são usados em conjunto com a leitura provida de outras fontes tais como *ladar* a laser, além de sensores de ultra-som, sonares e radares.

O controle de robôs que navegam em ambientes internos, tal como aqueles que navegam em ambientes externos, é responsável pela leitura dos dados provenientes de todos os sensores e pela manipulação destes dados de modo a estimar a localização do robô. O mapa do ambiente deve ser construído baseado na fusão de dados vindos de sensores tais como câmera de vídeo e ladar a laser, a partir de uma varredura minuciosa do ambiente e da comparação deste com o mapa construído *off-line* do mesmo ambiente, obtendo assim uma estimativa de posição do robô mais precisa. Por fim, para que o robô realize com sucesso sua missão, este deve ser capaz de gerar novos pontos de controle da trajetória, evitando assim obstáculos, trajetórias complexas e variação brusca de velocidade. A figura 2.3 mostra um mapa de ambiente construído durante a execução de uma missão.

2.4 Plataforma móvel Omni

A plataforma móvel Omni vem sendo desenvolvida desde 1998 para fins de pesquisa em robótica móvel em ambientes internos estruturados. Sua estrutura mecânica foi projetada e construída na tése de Le Corre [le Corre 1998], no LIRMM (*Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier*), em Montpellier, França. Neste período foi também concebida



Figura 2.3: Mapa de ambiente (cinza) e trajetória do robô (azul) (Borges 2002).

os circuitos de interface entre o computador central e alguns sensores, tal como codificadores ópticos, além de bibliotecas para acesso aos sensores para configurá-los e fazer a leitura dos dados, e controle dos motores por meio do computador central. Desde então, a plataforma vem sofrendo modificações tanto na sua arquitetura de *hardware* quanto de *software* a fim de satisfazer os requisitos de diversos trabalhos.

Quanto à sua estrutura mecânica, o robô Omni é um robô omnidirecional com três rodas nãocentradas orientáveis e tracionáveis, resultando seis eixos articulados, como mostrado na Figura 2.4³. Esta configuração lhe permite navegar facilmente em ambientes internos de difícil acesso, uma vez que cada eixo articular é controlado por um motor de corrente contínua.



Figura 2.4: Robô Omni.

Os sensores que compõem a plataforma móvel Omni são: codificadores ópticos incrementais e absolutos, leitor de balisas, girômetro a laser, *ladar* a laser e câmeras em configuração estéreo.

 $^{^{3}} http://www.lirmm.fr/\tilde{w}3rob/SiteWeb/detail_topic.php?num_topic=3$

Estes sensores são usados tanto para a estimação de posicionamento do robô quanto para mapear o ambiente em que se encontra. Circuitos fazem a interface entre o computador central e os sensores.

Diversos estudos em robótica móvel já foram realizados ou estão em andamento na plataforma Omni. Algoritmos de geração e controle de trajetória foram desenvolvidos por Aragones em [Aragones 2002]. O objetivo deste trabalho foi fazer com que o robô seguisse uma trajetória definida por uma curva suave de Bézier utilizando os pontos de controle à uma velocidade contínua [Aragones e Fournier 2002, Aragones, Borges e Fournier 2002]. O método desenvolvido permite considerar a incerteza estimada da posição absoluta do robô e considerar tais dados para corrigir a trajetória, preservando a continuidade do movimento. Em [Aragones, Borges e Fournier 2002] foi também realizado um estudo sobre meios de melhorar a precisão do posicionamento do robô Omni. Foram realizados estudos de melhoria da odometria por identificação de parâmetros e por minimização dos esforços articulares do robô, que podem aumentar a derrapagem do mesmo.

O problema do mapeamento bidimensional do ambiente e localização robusta foi abordado por Borges em [Borges 2002]. Neste trabalho, foi proposto uma nova arquitetura de localização dinâmica a partir de um sistema multi-sensorial (câmera de vídeo e ladar a laser). A arquitetura proposta foi implementada no robô Omni, permitindo a este robô se localizar, construir o mapa local do ambiente e se deslocar ao mesmo tempo.

Diversos trabalhos são atualmente desenvolvidos na plataforma móvel Omni. Estão sendo acoplado diversos sensores de ultra-som em torno da plataforma direcionadas para o chão, a fim de detectar obstáculos. Estudos com visão estéreo estão sendo feitos, além de tratamento de imagens para diversos tipos de aplicação. Tem-se também a implementação de algoritmos de geração de trajetória *on-line* a partir do mapa local do ambiente.

capítulo 3

Plataforma móvel Omni

3.1 Introdução

O robô Omni foi desenvolvido no LIRMM (*Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier*), em Montpellier, França, para fins de pesquisa em robótica móvel. Sua concepção mecânica foi feita na tese de Le Corre [le Corre 1998], no LIRMM. Desde então, esta plataforma sofreu mudanças em sua estrutura mecânica e eletrônica, até chegar na configuração atual. A Figura 3.1 mostra uma idéia da geometria do robô Omni. Suas dimensões físicas são: 860 mm de comprimento, 725 mm de largura e 760 mm de altura distribuídos em 350 Kg.

As três rodas orientáveis e tracionáveis de forma independente, resultando em seis eixos articulares, classificam o Omni como robô omnidirecional, como mostrado na Figura 3.1. Três atuadores são geralmente suficientes para proporcionar ao robô três graus de liberdade, mas no caso do robô Omni cada eixo articular é controlado por um motor de corrente contínua, fazendo com que o sistema seja sobre atuado. A redundância de atuadores e a configuração mecânica de suas rodas (descentralizadas e orientáveis ou do tipo "Castor") fazem com que menos restrições sejam impostas na planificação de trajetórias, quando comparadas a uma plataforma clássica, composta por duas rodas diferenciais, uma vez que esta primeira configuração permite plena mobilidade em um plano.

O controle do robô é feito por um micro computador IBM-PC Pentium TM II 300 MHz e placas



Figura 3.1: Plataforma móvel Omni e seus componentes.

transputer que servem de interface entre o computador e certos sensores . Sua arquitetura lógica foi desenvolvida no Windows $2000^{\odot 1}$ com extensão tempo real $RTX^{\odot 2}$. Periféricos, tais como tela LCD, teclado, mouse e joystick, permitem a interação entre o usuário e o Omni.

Vários sensores equipam a plataforma móvel. Estes sensores são utilizados ou para estimação relativa do veículo por odometria, denominados sensores proprioceptivos ou para localização absoluta do veículo no ambiente, denominados sensores exteroceptivos.

Este capítulo destina-se a apresentar todos os componentes que equipam o robô Omni, assim como a característica e funcionalidade de cada componente.

3.2 Arquitetura da Plataforma móvel Omni

A Figura 3.2 mostra como é a configuração de cada componente da plataforma e por que meio é feita a transferência de dados para o computador central e o envio do controle deste para os componentes. Todos os sensores, atuadores, circuitos e computador central são alimentados por quatro baterias de 12 V conectadas em série, fornecendo 4 8V ao sistema. Como nem todos os equipamentos são alimentados por esta tensão, conversores DC - DC são necessários para fornecer a tensão necessária para cada equipamento. A Tabela 3.1 mostra a tensão de alimentação dos diversos componentes da plataforma.

Como mostrado na Figura 3.2, o computador central é conectado a diferentes sensores por

 $^{^1 \}rm Windows$ 2000 é uma marca registrada da Microsoft, Inc

²RTX é uma marca registrada da Venturcom, Inc
Sensor	Tensão de Alimentação
Computador central	48 V
Joystick	30 V
Girômetro a laser	30 V
Ladar a laser	24 V
Motor	24 V
Leitor de balisas	24 V
codificadores ópticos	5 V

Tabela 3.1: Tensão de alimentação dos componentes da plataforma móvel Omni.

meio dos barramentos PCI e ISA. Conectado ao barramento PCI há uma placa de Entrada/Saída desenvolvida pelo seviço técnico do LIRMM, no qual estão presentes, entre outros, conversores analógico/digital e digital/analógico, conversores de padrão de comunicação serial de RS485 para RS232. Essa placa permite ler o estado do joystick, os dados dos codificadores ópticos e de enviar as tensões de referência dos motores dos eixos por meio das placas de controle de velocidade/corrente. Além disso, duas saídas seriais assíncronas permitem configurar o girômetro a laser e o goniômetro. É também conectado ao barramento PCI uma placa para transferência de dados, contendo três entradas *FireWire*, para aquisição dos dados vindos das câmeras de vídeo.

O barramento ISA é utilizado para conectar três placas de comunicação serial *INMOS*, interligados à placa *transputers*. As placas *transputer* fazem a aquisição dos dados do radar a laser, do girômetro a laser e do goniôemtro a laser. A utilização destas placas é herança de versões antigas desenvolvidas no Omni e é apresentada em [Borges 2002].

3.2.1 Motores

Como já mencionado na seção 3.1, cada eixo articular é acionado por um motor de corrente contínua, ou seja, cada roda é controlada por dois motores: um que controla a tração da roda e outro que controla o ângulo de direção da mesma. Cada motor é alimentado por 24 V providos da saída do conversor DC - DC e fornece um torque na faixa de 6, 5Kg - cm. O acionamento dos motores é feito por meio das placas de controle de velocidade/corrente, que são amplificadores de tensão ou corrente, controlados pelo computador central.

O sinal de referência é enviada pelo computador central às placas de controle de velocidade/corrente a partir de uma tensão na saída do conversor digital/analógico da placa de En-trada/Saida. Esta tensão varia de -10 V à +10 V e é proporcional à velocidade (voltagem) ou torque (corrente) de referência. O sinal de saída das placas de controle de velocidade/corrente é uma tensão em modulação por largura de pulso (PWM_Pulse Width Modulation) que é dependente do tipo de controle escolhido e do valor do sinal de entrada. E é esta tensão que aciona o motor e movimenta a roda.



Figura 3.2: Arquitetura física do robô Omni.

3.2.2 Sensores Proprioceptivos

Sensores proprioceptivos são sensores que fornecem informações relativas às variáveis do sistema em suas medidas. Em robótica móvel, os sensores proprioceptivos fornecem informações a respeito da movimentação relativa do veículo. Os sensores proprioceptivos que compõem o Omni são:

I. Seis Codificadores ópticos incrementais:

Codificadores ópticos incrementais são sensores que convertem variação de movimentação angular em uma série de pulsos elétricos. O codificador óptico é incremental quando não há um ponto de refência inicial do sistema, ou seja, a informação sobre o ângulo de rotação do eixo no qual se encontra tal sensor é referente ao ângulo no qual o sistema foi inicializado. A função deste sensor é medir a variação angular relativa das rodas e, mais importante, estimar a velocidade de rotação de cada eixo articula. A resolução de um codificador óptico incremental é dada pela relação *pulsos/revolução* e depende tanto de fatores mecânicos, como número de divisões que há no disco do sensor, configuração dos sensores ópticos, quanto de fatores elétricos.

No robô Omni, há dois codificadores ópticos incrementais em quadratura em cada um dos seis motores que controlam as rodas e eles são montados no eixo de cada motor. A saída do codificador óptico incremental é um pulso a cada variação do ângulo de rotação fixa. A quantidades de *ticks* por revolução destes sensores são de $1000K_{c_{ij}}$, no qual $K_{c_{ij}}$ é o ganho do *j-ésimo* codificador na *i-ésima* roda, sendo que esse ganho depende da configuração como é lido o sinal de saída do sensor, e pode assumir os valores 1, 2 e 4. Na configuração atual, todos os codificadores ópticos estão com o ganho de 4, uma vez que são considerados tanto as transições positivas quanto as negativas do sinal saída de cada um dos dois canais do codificador óptico.

Os codificadores ópticos que fazem medições nos motores que controlam o ângulo de direção das rodas estão fixados no eixo do motor antes da redução de 29, logo a conversão de *ticks* para graus é de $1000 \times 4 \times 29$ *ticks* por revolução ou 11600 *ticks* a cada 360° . Ele é usado tanto para estimar a velocidade de rotação do eixo, $\dot{\beta}$, como para estimar a posição absoluta do mesmo, β . Isso só é possível porque há um codificador óptico absoluto no mesmo eixo que faz a leitura do ângulo absoluto do ângulo inicial da roda e com este valor tem como estimar o ângulo absoluta do eixo a partir do decremento e incremento de rotações relativas. Esta forma de estimar o ângulo absoluto foi escolhida pois o codificador óptico incremental possui uma resolução maior que o absoluto.

Já os codificadores ópticos que fazem medições nos motores que controlam a rotação da roda são fixados no eixo do motor antes das reduções de 7, 2 e 38/15. A conversão de *ticks* para graus correspondente é de 1000 × 4 × 7.2 × 38/15 ticks a cada 360°. Este sensor mede a velocidade de rotação da roda, $\dot{\varphi}$, e o deslocamento relativo da mesma, $\Delta \varphi$.

A saída do encorder incremental é conectada a placa de *Entrada/Saída* (Figura 3.2), no qual há um *chip* que acumula o número de interrupções gerada pelas transições positivas e negativas de cada canal do codificador óptico. A cada período de amostragem ΔT_a , este *chip* entrega ao processador o valor acumulado. A velocidade e deslocamento angular média é estimada da seguinte forma:

$$\Delta \varphi = \frac{n_{ticks} \cdot 2\pi}{1000 \cdot 4 \cdot 7, 2 \cdot \frac{38}{15}} [rad] \tag{3.1}$$

$$\dot{\varphi} = \frac{n_{ticks} \cdot 2\pi}{\left(1000 \cdot 4 \cdot 7, 2 \cdot \frac{38}{15}\right) \cdot \Delta T_a} \left[\frac{rad}{seg}\right] \tag{3.2}$$

$$\Delta\beta = \frac{n_{ticks} \cdot 2\pi}{1000 \cdot 4 \cdot 29} \left[rad \right] \tag{3.3}$$

$$\dot{\beta} = \frac{n_{ticks} \cdot 2\pi}{(1000 \cdot 4 \cdot 29) \cdot \Delta T_a} \left[\frac{rad}{seg} \right]$$
(3.4)

II. Três codificadores ópticos absolutos:

Codificadores ópticos absolutos convertem variação da posição angular em pulsos elétricos, assim como o codificador óptico incremental, porém o sinal de saída do primeiro é um valor proporcional ao ângulo absoluto do eixo no qual está fixado. O elemento básico do codificador óptico absoluto é o disco de leitura estampado por um padrão de trilhas concêntricas. Esta trilha pode ser estampada com qualquer tipo de código binário, de modo que a saída do sensor corresponda a uma posição absoluta do eixo no qual o este está fixado. A codificação mais usada para estampar as trilhas é o codigo Gray, uma vez que a saída digital entre duas palavras de código consecutivos difere em apenas um bit.

Há um codificador óptico absoluto em cada um dos três motores que controla o ângulo de posição das rodas. Os codificadores ópticos são montados no eixo de direção da roda e mecanicamente eles são montados invertidos e é aplicado um *offset* digital. Isso significa que o valor de saída do sensor cresce no sentido negativo do ângulo de direção e o valor 0V não corresponde a 0^o do ângulo de posição. O sinal de saída do codificador óptico é proporcional ao ângulo de posição da roda e dado em 10 bits em código Gray, fazendo com que haja 2^{10} , ou 1024, *ticks* por revolução.

A saída do codificador óptico absoluto é conectada à placa *Entrada/Saída*. Na inicilização do sistema é enviado ao processador o sinal de saída do codificador óptico absoluto. Um *offset* digital é somado a leitura de modo que o valor binário 0 corresponda a 0^{o} . O resultado é então convertido para o código binário natural a partir de uma tabela de conversão e este valor binário é modificado para que este cresça quando o ângulo no eixo crescer. Somente então a leitura é convertida para o seu correspondente decimal, que indica o ângulo no qual o eixo se encontra.

III. Girômetro a laser:

O girômetro é um sensor que detecta o momento angular, sendo assim seu sinal de saída é proporcional à variação da orientação, ou velocidade angular, no eixo perpendicular à base do sensor. Em robótica móvel, este tipo de sensor é usado para medir a variação da orientação do robô, representado pela variável ω .

O princípio de funcionamento do girômetro a laser é baseado no *efeito Sagnac*. A diferença do comprimento óptico de dois feixes lasers viajando no mesmo circuito óptico fechado é proporcional à velocidade angular do circuito em coordenadas inerciais [Komoriya e Oyama 1992]. A estrutura de um girômetro a laser genérico é mostrado na Figura 3.3. A fonte luminosa de tal sensor é um diodo semicondutor laser. O feixe do laser é divido em dois. Uma detector óptico detecta a diferença do comprimento óptico dos dois feixes lasers que é obtido pela diferença entre o tempo em que os dois feixes percorrem a trajetória, sendo este tempo proporcional à velocidade angular.

O sinal de saída do girômetro presente na plataforma Omni é ΔT , proporcional à velocidade angular do robô, e esta relação é dada por:



Figura 3.3: Estrutura de um girômetro a laser genérico.

$$\omega = \Delta T \cdot \frac{\pi}{180 \cdot 60 \cdot 60 \cdot 10} \left[\frac{rad}{seg/10} \right]$$
(3.5)

Pela relação mostrado acima, tem-se que a precisão do girômetro a laser usado na plataforma. Omni é de $5 \cdot 10^{-7}$ radianos. A deriva deste mesmo sensor é inferior a $45^{o}/h$, de acordo com o fabricante. O sensor é montado na plataforma móvel de tal forma que o sinal de saída deste sensor seja a medida de velocidade angular no eixo perpendicular ao plano de movimento do robô, sendo que o deslocamento angular pode ser obtido sabendo o período de amostragem ΔT_a .

Por meio de uma interface serial síncrona RS-232 é feita a configuração do girômetro a laser pelo computador central. Os dados de saída do sensor são enviados para a placa transputer, a partir de uma interface síncrona RS-485, e são armazenados até serem enviados para o processador a cada período de amostragem ΔT_a .

3.2.3 Sensores Exteroceptivos

Os sensores exteroceptivos são aqueles que fornecem informações absolutas em suas medidas sobre o ambiente no qual o sistema está imerso. Sensores exteroceptivos são sensíveis tanto a mudanças do ambiente no qual está imerso quanto a mudança nas variáveis referentes ao próprio sistema. Somente sensores exteroceptivos são suficientes para detectar variações na configuração do sistema e no ambiente, teoricamente. Em robótica móvel, normalmente sensores proprioceptivo são usados em conjunto com sensores exteroceptivos e é feita a fusão de dados entre esses sensores para se obter uma melhor estimativa do posicionamento dos veículos. Dentre os sensores exteroceptivos existentes, os abaixo estão presentes na plataforma móvel Omni:

I. Radar a laser:

O radar a laser fornece uma imagem de profundidade 2-D que representa o contorno dos obstáculos ao seu redor, dentro de um *plano de dispersão*. Esta imagem é formada por medidas discretas, chamados *pontos de dispersão*. O sistema de medidas de distância é baseado na emissão de um trem de impulsos laser que, ao encontrar um obstáculo, é refletido na direção de emissão. A tempo de *ida-volta* dos impulsos são medidos por um relógio de alta freqüência. Teoricamente, a distância r do obstáculo ao sensor é diretamente proporcional ao tempo de vôo do trem de pulsos. Na prática, a medida sofre influência da luz ambiente, sendo este uma fonte de incerteza a ser considerada.



Figura 3.4: Plano de dispersão do ladar a laser.

A dispersão em um plano se faz com o auxílio de um espelho rotativo inclinado à 45° que gira a uma velocidade constante no sentido anti-horário em torno do eixo ortogonal ao plano de dispersão $\mathcal{X}^L \times \mathcal{Y}^L$. Um codificador óptico absoluto permite medir de forma precisa o ângulo correspondente de cada medida do sensor. Seja ϕ_n o ângulo de dispersão correspondente a *n-ésima* medida da distância r_n . A primeira medida é obtida na direção $\phi_{mín}$ e as medidas consecutivas são realizadas a cada $\Delta \phi$ até a última medida, na direção de $\phi_{máx}$. O conjunto de medidas do radar a laser forma uma imagem de profundidade 2-D com a seguinte notação:

$$\mathcal{L} = \{ (r_n, \phi_n) | n = 1, \dots, N; \phi_1 = \phi_{min}; \phi_N = \phi_{max}; \phi_n - \phi_{n-1} = \Delta \phi \}$$
(3.6)

O campo angular do sensor é de $\phi_{m\acute{a}x} - \phi_{m\acute{n}n}$. Cada par (r_n, ϕ_n) corresponde à coordenada polar do ponto de intersecção do feixe laser com a superfície do obstáculo na direção ϕ_n . Logo, 3.6 corresponde a representação polar de uma imagem de profundidade. A tabela 3.2 mostra os parâmetros do radar a laser, assim como os seus valores nominais.

II. Duas Câmeras em configuração estéreo:

A plataforma móvel Omni é equipada com duas câmeras montadas na configuração estéreo, como mostrado na Figura 3.1. Cada uma das câmeras fornece imagens de intensidade de cinza (fig. 3.5), com a notação \mathcal{I} , tal como:

Parâmetros	Valor do fabricante	Descrição
$r_{m\acute{a}x}$	30 m	Alcance máximo medido pelo ladar
$\phi_{ m min}$	-135^{o}	Ângulo mínimo de dispersão (programável)
$\phi_{ m max}$	135^{o}	Ângulo máximo de dispersão (programável)
$\Delta \phi$	$0, 6^{o}$	Resolução angular de ϕ (programável)
T	$0,125 \ s$	Período de uma volta completa de 360°
σ_r	0,03m	Desvio padrão da medida da distância \boldsymbol{r}

Tabela 3.2: Parâmetros do radar a laser, fornecido pelo fabricante IBEO Lasertechnik

$$\mathcal{I} = \{ p(u, v) | u = 1, \dots, M; v = 1, \dots, N \}.$$
(3.7)

Em \mathcal{I} , p(u, v) é uma função discreta monotônica e crescente da energia luminosa recebida pela célula de coordenadas, chamado também de *pixel*, (u, v) no *plano da imagem* durante o *período de integração* do sensor. As coordenadas métricas de um *pixel* estão relacionadas à sua coordenada na imagem (u, v) por meio de constantes multiplicativas h_u e h_v . Logo, o *pixel* (u, v) na imagem \mathcal{I} tem como coordenadas métricas (x^I, y^I) , com $x^I = h_u \cdot u$ e $y^I = h_v \cdot v$, no espaço euclidiano de \mathcal{R}^I , no sistema de coordenadas $\mathcal{R}^I : \mathcal{X}^I \times \mathcal{Y}^I$.

O modelo geométrico considerado é do tipo simplificado *pin-hole*. Supõe-se que o eixo óptico da câmera intercepta ortogonalmente o plano da imagem no ponto (x_0^I, y_0^I) . A distância entre o centro da imagem (x_0^I, y_0^I) e o foco do sistema ótico é a *distância focal*, representado por f. O foco é a origem do sistema de coordenadas 3-D da câmera: $\mathcal{R}^C : \mathcal{X}^C \times \mathcal{Y}^C \times \mathcal{Z}^C$. A Figura 3.5 mostra o sistema de coordenadas métrica fixo à câmera \mathcal{R}^C . Tanto os eixos \mathcal{X}^C e \mathcal{X}^I quanto \mathcal{Y}^C e \mathcal{Y}^I são paralelos.

A interface entre as câmeras e o computador central é feita pelo barramento IEEE 1394, conhecido como *FireWire*. O *FireWire* é um barramento serial de altíssimo desempenho que suporta uma taxa de transmissão de 400 Mbps (com expansão prevista para até 1 Gbps) [Dantas 2002]. O envio dos dados referentes a uma dada imagem é feito diretamente da câmera para o computador central. Por isso, movimentos rápidos com a câmera geram imagens distorcidas, pois as informações adquiridas pelo computador central são fragmentos referentes a imagens distintas obtidas pela câmera . A conexão e desconexão das câmeras pode ser feita de maneira dinâmica, ou seja, sem a necessidade de efetuar *shutdown* no computador.

III. Goniômetro a laser:

O robô Omni possui um goniômetro a laser em sua estrutura. A função de tal sensor é determinar a posição absoluta do robô a partir da detecção e reconhecimento de balisas codificadas e fixas no ambiente de navegação. Este sensor não é atualmente usado, pois para





a sua utilização, o meio no qual o robô navega deve ser modificado com a inserção de balisas em pontos de interesse, havendo assim interferência no espaço de trabalho do robô móvel.

3.3 Arquitetura lógica da Plataforma móvel Omni

3.3.1 Características do conjunto Windows2000/RTX 5.0

O Windows 2000 é um sistema operacional criado com a tecnologia do Windows NT. Ele é um sistema multitarefa/multi-usuários que utiliza o escalonamento por alternância circular (Round-robin) preemptivo de prioridade dinâmica para controlar a execução dos processos no processador. Este algoritmo de escalonamento busca maximizar a quantidade de tarefas ativas. RTX (de Real-Time eXtension), como o próprio nome diz, é uma extensão tempo real do Windows 2000, que interage com este último de modo como ilustrado na Figura 3.6. De fato, estendendo o sistema operacional Windows 2000, o RTX habilita componentes de aplicativos ou módulos que necessitam



de repostas rápidas e determinísticas ao longo do tempo, juntamente com aplicações não tempo real, trabalhando juntos em comum em um sistema Windows.

Figura 3.6: Organização dos módulos que compõem o conjunto Windows 2000/RTX.

As características principais de um sistema Windows 2000/RTX são:

- O kernel do Windows 2000 é tratado como uma tarefa de fundo do RTX. Ele se executa sempre que não há tarefa RTX ativa, por mínima que seja sua prioridade. Como a interface gráfica é completamente controlado pelo Windows 2000, as funções de impressão de caracteres como printf são consideradas não determinísticas, uma vez que sua execução é gerenciada pelo Windows 2000. Porém, RTX contém a biblioteca RTAPI_32, que inclui certas funções da API (Application Interface) do Windows 2000 que foram reescritas de forma que sejam determinísticas, tal como RtPrintf;
- RTX inclui uma extensão da HAL (*Hardware Abstraction Layer*) permitindo habilitações em tempo real de dispositivos de entrada e saída.
- As operações em ponto flutuante são autorizados em uma tarefa *RTX*, o que não é possível em programas sob o controle do *Windows 200*.

O RTX foi desenvolvido para promover o desenvolvimento de aplicações em tempo real. Intencionalmente, não foi incluído funções *Win32*, como chamadas a interface gráfica, que normalmente são usados em aplicações que não tem funções tempo-real. Para que possa haver comunicação entre qualquer tipo de processo, foram implementados alguns objetos de comunicação entre processos, tais como semáforos, mutex, memória compartilhada, além de temporizadores e relógios, a gerência de interrupção e barramentos, tarefas periódicas.

capítulo 4

Localização do Robô Omni

4.1 Introdução

Uma das características principais de robôs móveis é possuírem espaço de trabalho teoricamente infinito. Como estes veículos estão sujeitos a deslizamento das rodas, mesmo se o terreno em que se encontram é plano, rígido e sem deformações, para a realização de tarefas de navegação faz-se necessário um sistema de localização confiável, com incerteza limitada. As restrições cinemáticas dos robôs a rodas são caracterizadas pelo seu modelo cinemático, escrito na forma diferencial de uma equação diferencial vetorial. Para o caso de robôs a rodas com tração diferencial, esses modelos foram bastante explorados e estudados em [Campion, Bastin e D'Andréa-Novel 1996].

O modelo cinemático direto é de extrema importância para localização de robôs móveis. Na verdade, a quase totalidade dos sistemas de localização por mapas de ambiente utiliza o modelo cinemático para obter uma predição do deslocamento relativo do veículo. Isto é feito por meio da integração do modelo cinemático, que é alimentado por dados provenientes de sensores proprioceptivos. Os sensores proprioceptivos, tal como codificadores ópticos incrementais, fornecem medições de direcionamento e velocidade das rodas, obtendo-se então a chamada odometria. A inserção do girômetro no sistema permite que, ao invés de estimar o ângulo da orientação do robô, medir esta variável por meio do sensor, tendo-se assim a chamada girodometria.

Neste capítulo será mostrado o modelo cinemático, a odometria e a girodometria do robô

Omni. Por causa da sua configuração mecânica e dos inúmeros sistemas de referência relativos aos sensores, a estrutura da plataforma móvel Omni é muito complexa. Com isso é apresentado os sistemas de coordenadas 3-D definidos e as transformações entre esses sistemas e o sistema de referência global do Omni. Devido a simplificações no modelo cinemático e à imprecisão do modelo geométrico, o erro na estimação na odometria aumenta indefinidamente com o deslocamento do robô. Logo, a incerteza associado à estimação do deslocamento do robô deve ser considerada.

4.2 Sistemas de coordenadas da Plataforma móvel Omni

De acordo com a notação aplicada, um sistema de coordenadas é representado por \mathcal{R} . Os sistemas de coordenadas que compõem a plataforma móvel Omni são:

- \mathcal{R}^{G} é o sistema de coordenadas absoluta, que representa a referência absoluta adotada;
- \mathcal{R}^R é o sistema de coordenadas local, ou referente ao robô, fixo no centro geométrico dos eixos de direção das rodas;
- \mathcal{R}^L é o sistema de coordenadas do ladar a laser;
- \mathcal{R}^{C_E} é o sistema de coordenadas relativo à câmera esquerda, quando esta se encontra na configuração estéreo;
- \mathcal{R}^{C_D} é o sistema de coordenadas relativo à câmera direita, quando esta se encontra na configuração estéreo;
- $\mathcal{R}^{\mathcal{C}_{\mathcal{M}}}$ é o sistema de coordenadas relativo à câmera, quando esta se encontra na configuração mono.

4.2.1 Transformação de coordenadas

Dados dois sistemas de coordenadas $\mathcal{X}^{\mathcal{A}} \in \mathcal{X}^{\mathcal{B}}$ no espaço euclidiano 3-D, a relação entre as coordenadas (x^{A}, y^{A}, z^{A}) no espaço $\mathcal{X}^{A} \in (x^{B}, y^{B}, z^{B})$ no espaço \mathcal{X}^{B} de todos os pontos estáticos do espaço 3-D é representado por:

$$\begin{pmatrix} x^{B} \\ y^{B} \\ z^{B} \end{pmatrix} = \mathbf{R}_{z} \left(\theta_{z}\right) \cdot \mathbf{R}_{y} \left(\theta_{y}\right) \cdot \mathbf{R}_{x} \left(\theta_{x}\right) \cdot \left\{ \begin{pmatrix} x^{A} \\ y^{A} \\ z^{A} \end{pmatrix} - \mathbf{T} \right\}$$
(4.1)

no qual **T** é o vetor de translação representado por $(x, y, z)^T$ e representa a coordenada da origem de $\mathcal{X}^{\mathcal{B}}$ no sistema $\mathcal{X}^{\mathcal{A}}$. $\mathbf{R}_i(\theta_i)$ é a matriz de rotação em torno do *i-ésimo* eixo do sistema de coordenadas $\mathcal{X}^{\mathcal{A}}$.



Figura 4.1: Sistemas de referência principais do robô Omni: (a) visão superior e (b) visão frontal.

4.2.2 Posição da referência do ladar em relação ao sistema de referência do robô

A referência local do robô \mathcal{R}^R e o sistema de referência do ladar a laser \mathcal{R}^L são ligados pela matriz $Z_L^R = \{\mathbf{z}_L^R, \mathbf{0}_3\}$, no qual \mathbf{z}_L^R define a posição de \mathcal{R}^L em relação a \mathcal{R}^R . A matriz de covariância $\mathbf{0}_3$ mostra que não foi considerada a incerteza sobre a medida \mathbf{z}_L^R . As componentes de $\mathbf{z}_L^R = (x_L^R, y_L^R, z_L^R, \theta_{L_x}^R, \theta_{L_y}^R, \theta_{L_z}^R)^T$ foram determinadas experimentalmente a partir de medidas simples com régua. O ângulo $\theta_{L_z}^R$ é o ângulo de rotação em torno do eixo \mathcal{Z}^L , para que os eixos $\mathcal{X}^L, \mathcal{Y}^L$ e \mathcal{Z}^L fiquem paralelos aos eixos $\mathcal{X}^R, \mathcal{Y}^R$ e \mathcal{Z}^R , respectivamente. Os valores obtidos foram:

$$x_L^R = 0,3214m \quad y_L^R = 0,0730m \quad z_L^R = 0,480m \quad \theta_{L_x}^R = 0^o \quad \theta_{L_y}^R = 0^o \quad \theta_{L_z}^R = 0^o \tag{4.2}$$

Tem-se então que um ponto com a coordenada (x^L, y^L, z^L) no sistema de referência \mathcal{R}^L possui as coordenadas (x^R, y^R, z^R) no sistema de referência \mathcal{R}^R dada pela seguinte relação:

$$\begin{pmatrix} x^{R} \\ y^{R} \\ z^{R} \end{pmatrix} = \mathbf{I}_{3\times3} \cdot \left\{ \begin{pmatrix} x^{L} \\ y^{L} \\ z^{L} \end{pmatrix} - \begin{pmatrix} x^{R}_{L} \\ y^{R}_{L} \\ z^{R}_{L} \end{pmatrix} \right\}$$
(4.3)

4.2.3 Posição da referência da câmera em relação ao sistema de referência do robô

Na plataforma móvel Omni, há a possibilidade de fixar três câmeras em sua estrutura: duas câmeras são fixadas em configuração estéreo e uma terceira câmera é fixada com um ângulo de 90° em relação à primeira configuração, como mostrado na figura 4.2.



Figura 4.2: Configurações possíveis das câmeras no robô Omni.

A referência local do robô \mathcal{R}^R e o sistema de referência de cada uma das câmeras \mathcal{R}^{C_E} , \mathcal{R}^{C_D} e \mathcal{R}^{C_M} são ligadas as matrizes $\mathbf{z}_{C_E}^R$, $\mathbf{z}_{C_D}^R$ e $\mathbf{z}_{C_M}^R$, respectivamente, que possui os seguintes valores:

$$\begin{aligned} x^R_{C_E} &= 0,3214m \quad y^R_{C_E} = 0,0730m \quad z^R_{C_E} = 0,6700m \quad \theta^R_{C_{Ex}} = 90^o \quad \theta^R_{C_{EY}} = 0^o \quad \theta^R_{C_{Ez}} = -90^o \\ x^R_{C_D} &= 0,2214m \quad y^R_{C_D} = 0,0730m \quad z^R_{C_D} = 0,6700m \quad \theta^R_{C_{Dx}} = 90^o \quad \theta^R_{C_{DY}} = 0^o \quad \theta^R_{C_{Dz}} = -90^o \\ x^R_{C_M} &= 0,1714m \quad y^R_{C_M} = 0,0730m \quad z^R_{C_M} = 0,6250m \quad \theta^R_{C_{Mx}} = 0^o \quad \theta^R_{C_{MY}} = 0^o \quad \theta^R_{C_{Mz}} = -90^o \\ \end{aligned}$$

Com isso têm-se as seguintes equações de transformações

• Câmera Esquerda:

$$\begin{pmatrix} x^{R} \\ y^{R} \\ z^{R} \end{pmatrix} = \mathbf{R}_{z} \left(\theta^{R}_{C_{Ez}} \right) \cdot \mathbf{I}_{3 \times 3} \cdot \mathbf{R}_{x} \left(\theta^{R}_{C_{Ex}} \right) \cdot \left\{ \begin{pmatrix} x^{C_{E}} \\ y^{C_{E}} \\ z^{C_{E}} \end{pmatrix} - \begin{pmatrix} x^{R}_{C_{E}} \\ y^{R}_{C_{E}} \\ z^{R} \end{pmatrix} \right\}$$
(4.4)
$$\operatorname{com} \mathbf{R}_{z} \left(\theta^{R}_{C_{Ez}} \right) = \begin{pmatrix} \cos \left(\theta^{R}_{C_{Ez}} \right) & -\sin \left(\theta^{R}_{C_{Ez}} \right) & 0 \\ \sin \left(\theta^{R}_{C_{Ez}} \right) & \cos \left(\theta^{R}_{C_{Ez}} \right) & 0 \\ 0 & 0 & 1 \end{pmatrix} e \mathbf{R}_{x} \left(\theta^{R}_{C_{Ex}} \right) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \left(\theta^{R}_{C_{Ex}} \right) & -\sin \left(\theta^{R}_{C_{Ex}} \right) \\ 0 & \sin \left(\theta^{R}_{C_{Ex}} \right) & \cos \left(\theta^{R}_{C_{Ex}} \right) \end{pmatrix}$$

• Câmera Direita

$$\begin{pmatrix} x^{R} \\ y^{R} \\ z^{R} \end{pmatrix} = \mathbf{R}_{z} \left(\theta^{R}_{C_{Dz}}\right) \cdot \mathbf{I}_{3 \times 3} \cdot \mathbf{R}_{x} \left(\theta^{R}_{C_{Dx}}\right) \cdot \left\{ \begin{pmatrix} x^{C_{D}} \\ y^{C_{D}} \\ z^{C_{D}} \end{pmatrix} - \begin{pmatrix} x^{R}_{C_{D}} \\ y^{R}_{C_{D}} \\ z^{R} \end{pmatrix} \right\}$$
(4.5)
$$\operatorname{com} \mathbf{R}_{z} \left(\theta^{R}_{C_{Dz}}\right) = \begin{pmatrix} \cos\left(\theta^{R}_{C_{Dz}}\right) & -\sin\left(\theta^{R}_{C_{Dz}}\right) & 0 \\ \sin\left(\theta^{R}_{C_{Dz}}\right) & \cos\left(\theta^{R}_{C_{Dz}}\right) & 0 \\ 0 & 0 & 1 \end{pmatrix} e$$
$$\mathbf{R}_{x} \left(\theta^{R}_{C_{Dx}}\right) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\left(\theta^{R}_{C_{Dx}}\right) & -\sin\left(\theta^{R}_{C_{Dx}}\right) \\ 0 & \sin\left(\theta^{R}_{C_{Dx}}\right) & \cos\left(\theta^{R}_{C_{Dx}}\right) \end{pmatrix}$$

• Câmera Avulsa

$$\begin{pmatrix} x^{R} \\ y^{R} \\ z^{R} \end{pmatrix} = \mathbf{R}_{z} \left(\theta^{R}_{C_{Mz}} \right) \cdot \mathbf{I}_{3 \times 3} \cdot \mathbf{I}_{3 \times 3} \cdot \left\{ \begin{pmatrix} x^{C_{M}} \\ y^{C_{M}} \\ z^{C_{M}} \end{pmatrix} - \begin{pmatrix} x^{R}_{C_{M}} \\ y^{R}_{C_{M}} \\ z^{R}_{C_{M}} \end{pmatrix} \right\}$$
(4.6)
$$\operatorname{com} \mathbf{R}_{z} \left(\theta^{R}_{C_{Mz}} \right) = \begin{pmatrix} \operatorname{cos} \left(\theta^{R}_{C_{Mz}} \right) & -\operatorname{sin} \left(\theta^{R}_{C_{Mz}} \right) & 0 \\ \operatorname{sin} \left(\theta^{R}_{C_{Mz}} \right) & \operatorname{cos} \left(\theta^{R}_{C_{Mz}} \right) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

4.3 Modelo Cinemático do robô Omni

A modelagem cinemática permite identificar as relações existentes entre variações das variáveis de configuração \mathbf{q} no espaço articular do robô, e variações de sua pose $\mathbf{z} = (x, y, \theta)$ dentro do espaço cartesiano de trabalho. Na posição absoluta \mathbf{z} , $x \in y$ são as coordenadas do centro geométrico G do suporte das rodas no sistema de coordenadas absoluta, e θ é o ângulo entre o eixo X' e o eixo X. Para modelar as equações cinemáticas que descrevem o robô Omni, considera-se que há somente a rolagem das rodas, ou seja, desconsidera-se o deslizamento na região de contato entre a roda e o solo. Os parâmetros geométricos envolvidos no modelo cinemático são mostrados na Figura 4.3.

Os eixos XY representam o sistema de coordenadas absoluto, enquanto os eixos X'Y' representam o sistema de coordenadas do robô. A plataforam móvel Omni possui três rodas não centradas. Para a *i-ésima* roda, tem-se os seguintes parâmetros geométricos:

• r_i : seu raio (em metros);



Figura 4.3: Variáveis (a) da posição absoluta e (b) do modelo cinemático do robô Omni.



Figura 4.4: Configuração da roda descentralizada e orientável.

- e_i : a distância entre o eixo de orientação do suporte da roda (Ponto A_i no sistema de coordenadas X'Y') e seu centro de rotação. O eixo de orientação passa pelo ponto de contato do suporte da roda e a estrutura do robô (figura 4.4);
- $x_{ai} \in y_{ai}$: coordenadas do ponto A_i do eixo de orientação do suporte da roda no sistema de coordenadas do robô;
- (l_i, α_i) : coordenadas polares do ponto A_i no sistema de coordenadas X'Y', com $\alpha_1 = \pi/6, \alpha_2 = 5\pi/6$ e $\alpha_3 = 9\pi/6$. Com isso tem-se as seguintes relações:

$$x_{ai} = l_i \cos\left(\alpha_i\right) \tag{4.7}$$

$$y_{ai} = l_i \sin\left(\alpha_i\right) \tag{4.8}$$

Os parâmtros geométricos do robô Omni medidos com régua e paquímetro são:

$$\lambda_0 = \begin{cases} r_1 = 0, 1m & e_1 = 0, 05m & x_{a1} = 0, 325m & y_{a1} = 0, 188m \\ r_2 = 0, 1m & e_2 = 0, 05m & x_{a2} = -0, 325m & y_{a2} = 0, 188m \\ r_3 = 0, 1m & e_3 = 0, 05m & x_{a1} = 0m & y_{a3} = -0, 376m \end{cases}$$
(4.9)

Considerando $\mathbf{r} = (r_1, r_2, r_3)^T$, $\mathbf{e} = (e_1, e_2, e_3)^T$, $\mathbf{x}_a = (x_{a1}, x_{a2}, x_{a3})^T$ e $\mathbf{y}_a = (y_{a1}, y_{a2}, y_{a3})^T$, os parâmetros geométricos são representados por $\lambda = (\mathbf{r}^T, \mathbf{e}^T, \mathbf{x}_a^T, \mathbf{y}_a^T)$. As variáveis de configuração são representadas pelo seguinte vetor:

$$q = \left(\begin{array}{cccc} \beta_1 & \beta_2 & \beta_3 & \varphi_1 & \varphi_2 & \varphi_3 \end{array}\right)^T \tag{4.10}$$

em que β_i é o ângulo de direção da roda i e φ_i sua posição angular. A velocidade angular $\dot{\varphi_i}$ e o ângulo de direção de cada roda são comandados independentemente pelos motores de corrente contínua.

Campion, em [Campion, Bastin e D'Andréa-Novel 1996], apresentou as equações de modelos cinemáticos e dinâmicos generalizados para diferentes classes de robôs móveis.

$$\left(-\sin\left(\alpha_{i}+\beta_{i}^{\prime}\right) \cos\left(\alpha_{i}+\beta_{i}^{\prime}\right) \quad l_{i}\cos\left(\beta_{i}^{\prime}\right)\right)\mathbf{R}_{h}\left(\theta\right)\dot{\mathbf{z}}-r_{i}\dot{\varphi}_{i}^{\prime}=0,\tag{4.11}$$

$$\left(\cos\left(\alpha_{i}+\beta_{i}'\right) \quad \sin\left(\alpha_{i}+\beta_{i}'\right) \quad e_{i}+l_{i}\sin\left(\beta_{i}'\right) \quad \mathbf{R}_{h}\left(\theta\right)\dot{\mathbf{z}}+e_{i}\dot{\beta}_{i}'=0, \quad (4.12)$$

no qual

$$\mathbf{R}_{h}(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0\\ -\sin(\theta) & \cos(\theta) & 0\\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}(\theta) & \mathbf{0}_{2\times 1}\\ \mathbf{0}_{1\times 2} & 1 \end{pmatrix}$$

é a transformação homogênea constituída de uma rotação θ e representada pela matriz $\mathbf{R}_h(\theta)$. Esta matriz é quadrada e ortogonal. Portanto, $\mathbf{R}_h^T(\theta) = \mathbf{R}_h^{-1}(\theta)$.

As equações 4.11 e 4.12 utilizam uma notação diferente da adotada por Campion, no qual os ângulos $\dot{\beta}'_i$ e $\dot{\varphi}'_i$ possuem uma referência diferente daquela utilizada no robô Omni. No caso do robô Omni, os ângulos $\dot{\beta}'_i$ e $\dot{\varphi}'_i$ se relacionam com os ângulos β'_i e φ'_i da seguinte maneira:

$$\beta_i = \left(\beta'_i + \alpha_i\right) + \frac{\pi}{2},\tag{4.13}$$

$$\varphi_i = -\varphi_i' \tag{4.14}$$

A mudança na referência em relação aos trabalhos de Campion em [Campion, Bastin e D'Andréa-Novel 1996 é devido à restrições no projeto mecânico da plataforma móvel Omni. Com esta nova representação, tem-se que para *i-ésima* roda, as seguintes restrições:

$$\left(\begin{array}{cc}\cos\left(\beta_{i}\right) & \sin\left(\beta_{i}\right) & l_{i}\sin\left(\beta_{i}-\alpha_{i}\right)\end{array}\right)\mathbf{R}_{h}\left(\theta\right)\dot{\mathbf{z}}-r_{i}\dot{\varphi}_{i}^{\prime}=0$$
(4.15)

$$\left(\sin(\beta_i) - \cos(\beta_i) \quad l_i \cos(\beta_i - \alpha_i) \right) \mathbf{R}_h(\theta) \, \dot{\mathbf{z}} - e_i \dot{\beta}'_i = 0 \tag{4.16}$$

no qual $\dot{\beta}_i = \dot{\beta}'_i \in \dot{\phi}_i = \dot{\phi}'_i$.

Logo, para o robô Omni, composto por três rodas diretrizes e motrizes não-centrada, o modelo cinemático inverso é dado por[Borges 2002]:

$$\dot{\mathbf{q}} = \mathbf{J} \left(\mathbf{q}, \lambda, \theta \right) \cdot \dot{\mathbf{z}} \tag{4.17}$$

Na equação 4.17, $\mathbf{J}(\mathbf{q},\lambda,\theta)$ é a matriz jacobiana de dimensão 6×3 dado por:

$$\mathbf{J}(\mathbf{q},\lambda,\theta) = \mathbf{S}(\mathbf{q},\lambda) \cdot \mathbf{R}_{h}(\theta)$$
(4.18)

 com

$$\mathbf{S}(\mathbf{q},\lambda) = \begin{pmatrix} \left(\begin{array}{ccc} \frac{1}{e_{1}} & 0 & 0\\ 0 & \frac{1}{e_{2}} & 0\\ 0 & 0 & \frac{1}{e_{3}} \end{array}\right) \cdot \left(\begin{array}{ccc} -\sin\left(\beta_{1}\right) & \cos\left(\beta_{1}\right) & -e_{1} + x_{a1}\cos\left(\beta_{1}\right) + y_{a1}\sin\left(\beta_{1}\right)\\ -\sin\left(\beta_{2}\right) & \cos\left(\beta_{2}\right) & -e_{2} + x_{a2}\cos\left(\beta_{2}\right) + y_{a2}\sin\left(\beta_{2}\right)\\ -\sin\left(\beta_{3}\right) & \cos\left(\beta_{3}\right) & -e_{3} + x_{a3}\cos\left(\beta_{3}\right) + y_{a3}\sin\left(\beta_{3}\right) \end{pmatrix} \\ \left(\begin{array}{c} \frac{1}{r_{1}} & 0 & 0\\ 0 & \frac{1}{r_{2}} & 0\\ 0 & 0 & \frac{1}{r_{3}} \end{array}\right) \cdot \left(\begin{array}{c} \cos\left(\beta_{1}\right) & \sin\left(\beta_{1}\right) & x_{a1}\sin\left(\beta_{1}\right) - y_{a1}\cos\left(\beta_{1}\right)\\ \cos\left(\beta_{2}\right) & \sin\left(\beta_{2}\right) & x_{a2}\sin\left(\beta_{2}\right) - y_{a2}\cos\left(\beta_{2}\right)\\ \cos\left(\beta_{3}\right) & \sin\left(\beta_{3}\right) & x_{a3}\sin\left(\beta_{3}\right) - y_{a3}\cos\left(\beta_{3}\right) \end{array}\right) \end{pmatrix}\right).$$

O modelo cinemático direto dá a variação de posição do robô em função das variáveis cinemáticas. Este modelo é obtido a partir do modelo cinemático inverso mostrado na equação 4.17, e dado por:

$$\dot{\mathbf{z}} = \mathbf{J}^{\dagger} \left(\mathbf{q}, \lambda, \theta \right) \dot{\mathbf{q}} \tag{4.19}$$

no qual $\mathbf{J}^{\dagger}(\mathbf{q},\lambda,\theta)$ é pseudo-inversa de Moore-Penrose de $\mathbf{J}(\mathbf{q},\lambda,\theta)$:

$$\mathbf{J}^{\dagger}(\mathbf{q},\lambda,\theta) = \left(\mathbf{J}^{T}(\mathbf{q},\lambda,\theta) \cdot \mathbf{J}(\mathbf{q},\lambda,\theta)\right)^{-1} \cdot \mathbf{J}^{T}(\mathbf{q},\lambda,\theta)$$
(4.20)

Dado a propriedade de inversão da matriz $\mathbf{R}_h(\theta)$, tem-se que a transposta da matriz jacobiana é dado por:

$$\mathbf{J}^{\dagger}(\mathbf{q},\lambda,\theta) = \mathbf{R}_{h}^{T}(\theta) \cdot \mathbf{S}^{\dagger}(\mathbf{q},\lambda)$$
(4.21)

A posição $\dot{\mathbf{z}}$ do robô pode ser estimado como a solução dos mínimos quadrados de um sistema de 6 equações 3 três incógnitas (eq. 4.17). O teorema abaixo, ilustra esta solução ([Borges 2002]):

Theorem 1 $\dot{\mathbf{z}} = \mathbf{J}^{\dagger}(\mathbf{q},\lambda,\theta) \dot{\mathbf{q}}$ *é a solução da mínima norma de* $\mathbf{J}(\mathbf{q},\lambda,\theta) \cdot \dot{\mathbf{z}} = \dot{\mathbf{q}}$ *em mínimos quadrados, isto é, a solução de* $\dot{\mathbf{z}}$ *que minimaliza*

$$V_{MC} = \|\mathbf{\dot{q}} - \mathbf{J}(\mathbf{q},\lambda,\theta) \cdot \mathbf{\dot{z}}\|^2.$$
(4.22)

Para determinar $\dot{\mathbf{z}}$, decessita-se calcular $\mathbf{J}^{\dagger}(\mathbf{q},\lambda,\theta)$. Campion em [Campion, Bastin e D'Andréa-Novel 1996] mostrou que o rank de $\mathbf{J}(\mathbf{q},\lambda,\theta)$ é sempre igual a três para robôs omnidirecionais. Logo, $\mathbf{J}^{\dagger}(\mathbf{q},\lambda,\theta)$ é sempre calculado pela equação 4.20

4.4 Odometria

No robô Omni, o modelo cinemático é empregado sob a forma discretizada para atualizar a posição do veículo em função da variação das variáveis de comando das rodas dentro do perído de amostragem. Isto é feito discretizando a equação 4.19, utilizando a aproximação de Euler de primeira ordem:

$$\hat{\mathbf{z}}(k) = \hat{\mathbf{z}}(k-1) + \Delta \hat{\mathbf{z}}(k)$$
(4.23)

$$\Delta \hat{\mathbf{z}}(k) = \mathbf{J}^{\dagger} \left(\mathbf{q}(k-1), \lambda, \hat{\theta}(k-1) \right) \cdot \Delta \mathbf{q}(k)$$

$$\Delta \mathbf{q}(k) = \mathbf{q}(k) - \mathbf{q}(k-1)$$
(4.24)

Que pode ser simplificado na seguinte equação:

$$\hat{\mathbf{z}}(k) = \hat{\mathbf{z}}(k-1) + \mathbf{J}^{\dagger} \left(\mathbf{q}(k-1), \lambda, \hat{\theta}(k-1) \right) \cdot \Delta \mathbf{q}(k)$$
(4.25)

Estas equações mostram como é feita a estimação de posição do robô no instante discreto ka partir do seu estado no instante k-1 e a leitura dos encoders no intantes k e k-1. Nota-se que a estimação de deslocamente do robô entre os instantes k e k-1 é obtida por uma aproximação de primeira ordem, e os termos de ordem superior são negligenciados. Isso porque, espera-se que movimentação do robô respeitem as hipóteses supostas abaixo:

- I. A configuração geométrica do robô é exatamente como mostrado na figura 4.3;
- II. A discretização considera nulas as componentes de ordem superior do modelo cinemático;
- III. As rodas não derrapam durante o deslocamento do robô, sendo o seu contato considerado pontual;
- IV. O solo no qual o robô navega é perfeitamente plano;

V. Os valores dos parâmetros λ são exatamente conhecidos.

O não cumprimento destas hipóteses contribui para um aumento sem limite do erro de posicionamento por odometria. Infelizmente, em sistemas reais, todas essas hipóteses não são satisfeitas.

4.5 Girodometria

Uma variável que contribui de forma importante para o erro crescente na odometria é a orientação do robô θ . De fato, como as componentes de $\mathbf{J}^{\dagger} \left(\mathbf{q}_{k-1}, \lambda, \hat{\theta}_{k-1} \right)$ são fortemente não-lineares e dependentes da matriz \mathbf{R}_h , erros na estimação de $\hat{\theta}_{k-1}$ são re-integrados na estimativa $\hat{\theta}_k$. Uma forma de reduzir este erro consiste em empregar um sensor independente para medir esta variável, como é o caso dos giroscópios. E se o giroscópio for a laser, como no caso do robô Omni, a incerteza associada a $\hat{\theta}_k$ será enormemente reduzida. Com um giroscópio, $\hat{\theta}_k$ seria obtida por

$$\hat{\theta}_k = \hat{\theta}_{k-1} + T_s \omega_k \tag{4.26}$$

com ω_k sendo a medida de velocidade angular medida com o giroscópio no eixo perpendicular ao plano de movimento do robô, e T_s é o período de amostragem.

Dispondo de (4.26), as relações de odometria podem ser manipuladas, de forma que $\hat{\theta}_k$ seja considerada como uma entrada, tal como \mathbf{q}_{k-1} . A técnica resultante é chamada de Girodometria. Pode-se observar que o modelo cinemático inverso (4.19) pode ser escrito como:

$$\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q}, \lambda) \cdot \mathbf{R}_{\mathbf{h}}(\theta) \cdot \dot{\mathbf{z}}$$
(4.27)

$$= \mathbf{S}_{xy}(\mathbf{q},\lambda) \cdot \mathbf{R}\left(\theta\right) \cdot \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} + \mathbf{S}_{\theta}(\mathbf{q},\lambda) \cdot \dot{\theta}$$
(4.28)

com

$$\mathbf{S}_{xy}(\mathbf{q},\lambda) = \begin{pmatrix} \left(\begin{array}{ccc} \frac{1}{e_1} & 0 & 0\\ 0 & \frac{1}{e_2} & 0\\ 0 & 0 & \frac{1}{e_3} \\ \frac{1}{r_1} & 0 & 0\\ 0 & \frac{1}{r_2} & 0\\ 0 & 0 & \frac{1}{r_3} \end{array} \right) & \begin{pmatrix} -\sin\left(\beta_1\right) & \cos\left(\beta_1\right)\\ -\sin\left(\beta_2\right) & \cos\left(\beta_2\right)\\ -\sin\left(\beta_3\right) & \cos\left(\beta_3\right) \end{pmatrix} \\ \begin{pmatrix} \cos\left(\beta_1\right) & \sin\left(\beta_1\right)\\ \cos\left(\beta_2\right) & \sin\left(\beta_2\right)\\ \cos\left(\beta_3\right) & \sin\left(\beta_3\right) \end{pmatrix} \end{pmatrix}$$
(4.29)

$$\mathbf{S}_{\theta}(\mathbf{q},\lambda) = \begin{pmatrix} \left(\begin{array}{ccc} \frac{1}{e_{1}} & 0 & 0\\ 0 & \frac{1}{e_{2}} & 0\\ 0 & 0 & \frac{1}{e_{3}} \\ \frac{1}{r_{1}} & 0 & 0\\ 0 & \frac{1}{r_{2}} & 0\\ 0 & 0 & \frac{1}{r_{3}} \end{array} \right) & \begin{pmatrix} -e_{1} + x_{a1}\cos\left(\beta_{1}\right) + y_{a1}\sin\left(\beta_{1}\right) \\ -e_{2} + x_{a2}\cos\left(\beta_{2}\right) + y_{a2}\sin\left(\beta_{2}\right) \\ -e_{3} + x_{a3}\cos\left(\beta_{3}\right) + y_{a3}\sin\left(\beta_{3}\right) \end{pmatrix} \\ -e_{3} + x_{a3}\cos\left(\beta_{3}\right) + y_{a3}\sin\left(\beta_{3}\right) \end{pmatrix} \\ \begin{pmatrix} x_{a1}\sin\left(\beta_{1}\right) - y_{a1}\cos\left(\beta_{1}\right) \\ x_{a2}\sin\left(\beta_{2}\right) - y_{a2}\cos\left(\beta_{2}\right) \\ x_{a3}\sin\left(\beta_{3}\right) - y_{a3}\cos\left(\beta_{3}\right) \end{pmatrix} \end{pmatrix}$$
(4.30)

Por um processo de linearização semelhante à 4.25, obtém-se a seguiten fórmula para a girodometria:

$$\hat{\mathbf{z}}_{k} = \hat{\mathbf{z}}_{k-1} + \begin{pmatrix} \left(\mathbf{S}_{xy}(\mathbf{q}_{k-1}, \lambda) \cdot \mathbf{R} \begin{pmatrix} \hat{\theta}_{k-1} \end{pmatrix} \right)^{\dagger} & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 6} & 1 \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{q}_{k} - \mathbf{S}_{\theta}(\mathbf{q}_{k}, \lambda) \cdot \boldsymbol{\Delta} \theta_{k} \\ \Delta \theta_{k} \end{pmatrix}, \quad (4.31)$$

No qual $\Delta \theta_k$ é a medida da variação do ângulo de postura do robô Omni obtido pelo Girometro laser entre os instantes k-1 e k. A utilização do girometro laser preciso e com uma pequena deriva (no caso do Omni, $45^o/h$) permite minimizar o erro na estimação da posição do robô.

4.6 Modelo de propagação de erro

Para criar o modelo da incerteza associado à incerteza na estimãção da posição do robô Omni por Odometria ou Girodometria, supõe-se que todos os erros seguem uma distribuição multivariável com média nula e uma matriz de covariância associada. Logo, a estimativa da posição em um instante discreto k pode ser representado como $\mathcal{N}(\hat{\mathbf{z}}_k, \Lambda_{\hat{z}_k})$ e a incerteza é representada pela matriz $\Lambda_{\hat{z}_k}$.

As fórmulas de atualização da estimativa de $\hat{\mathbf{z}}_k$ por Odometria (equação 4.25) ou por Girodometria (4.31) podem ser formuladas da seguintes forma:

$$\hat{\mathbf{z}}_{k} = \hat{\mathbf{z}}_{k-1} + \Delta \hat{\mathbf{z}}_{k} \left(\hat{\theta}_{k-1}, \mathbf{q}_{k-1}, \Delta \mathbf{q}_{k}, \Delta \theta_{k}, \lambda \right)$$
(4.32)

$$= \mathbf{f} \left(\hat{\mathbf{z}}_{k-1}, \mathbf{q}_{k-1}, \Delta \mathbf{q}_k, \Delta \theta_k, \lambda \right), \qquad (4.33)$$

no qual o termo $\Delta \theta_k$ não é considerado quando a girodometria é utilizada. Para a propagação da incerteza, tem-se que as dispersões são representadas por:

- $\Lambda_{\hat{z}_{k-1}}$: matriz de covariância de \hat{z}_{k-1} , calculado a partir da atualização da posição anterior;
- $\sigma^2 \Delta \theta_k$: variância de $\Delta \theta_k$, dado pelo fabricante ¹

¹Somente para a girodometria

• σ_{ri}^2 : variância da *i-ésima* componente de $\mathbf{r}_k = \Delta \mathbf{q}_k - \mathbf{J} \left(\mathbf{q}_{k-1}, \lambda, \hat{\theta}_{k-1} \right) \cdot (\hat{\mathbf{z}}_k - \hat{\mathbf{z}}_{k-1})^2$. De acordo com o teorema 1, \mathbf{r}_k é o vetor dos resíduos da estimação dentro do espaço de $\Delta \mathbf{q}_k$. Se $\Delta \mathbf{q}_k$ não é coerente com o modelo, esses resíduos serão importantes (por exemplo, na derrapagem de uma roda). Supondo as componentes de r_k independentes, a variância empírica de qualquer uma delas é dada por:

$$\sigma_{ri}^2 = \frac{1}{5} \cdot \mathbf{r}_k^T \cdot \mathbf{r}_k. \tag{4.34}$$

• Λ_{λ} : a matriz de covariância de λ . Ela foi obtida intuitivamente em [Borges 2002] e seu valor foi de:

$$\Lambda_{\lambda} = (0, 03m)^2 \,\mathbf{I}_{12} \tag{4.35}$$

Esta matrix de covariância corresponde a um desvio padrão de 3 cm para cada componente de $\lambda.$

Se forem considerados como fontes de incertezas somente aquelas citadas acima, a matriz de covariância $\Lambda_{\hat{z}_k}$ é determinado pela propogação direta:

$$\Lambda_{\hat{\mathbf{z}}_{k}} = \left(\bigtriangledown_{\hat{\mathbf{z}}_{k-1}} \mathbf{f} \right) \cdot \Lambda_{\hat{\mathbf{z}}_{k-1}} \cdot \left(\bigtriangledown_{\hat{\mathbf{z}}_{k-1}} \mathbf{f} \right)^{T} + \left(\bigtriangledown_{\Delta \theta_{k}} \mathbf{f} \right) \cdot \sigma^{2} \Delta \theta_{k} \cdot \left(\bigtriangledown_{\Delta \theta_{k}} \mathbf{f} \right)^{T} \\ + \left(\bigtriangledown_{r_{k}} \mathbf{f} \right) \cdot \mathbf{Diag} \left(\sigma_{r_{1}}^{2}, \dots, \sigma_{r_{6}}^{2} \right) \cdot \left(\bigtriangledown_{r_{k}} \mathbf{f} \right)^{T} + \mathbf{\Gamma} \cdot \left(\bigtriangledown_{\lambda} \mathbf{f} \right) \cdot \Lambda_{\lambda} \cdot \left(\bigtriangledown_{\lambda} \mathbf{f} \right)^{T} \cdot \mathbf{\Gamma}^{T}$$

no qual Γ é uma matriz diagonal que permite aumentar artificialmente a porção da incertreza relacionada a λ . De fato, se $\Gamma = \text{Diag}(\gamma_x, \gamma_y, \gamma_\theta)$, pode-se entao amplificar independentemente os eixos de incerteza sobre $x, y \in \theta$. Logo, Γ deve ser determinado com os dados experimentais, de modo que $\Lambda_{\mathbf{z}_k}$ seja uma estimativa consistente, seja pessimista sobre a incerteza sobre \hat{z}_k . Em robótica móvel, este tipo de ajuste empírico da incerteza sobre uma estimativa é muito utilizada. Os resultados obtidos por Borges em [Borges 2002] do vetor Γ foram:

$$Para \ a \ Odometria \quad : \quad \gamma_x = 10, \quad \gamma_y = 10, \quad \gamma_\theta = 5, \tag{4.36}$$

$$Para \ a \ Girodometria \quad : \quad \gamma_x = 2, \quad \gamma_y = 2, \quad \gamma_\theta = 0, \tag{4.37}$$

 $\gamma_{\theta} = 0$ para a Girodometria significa que a incerteza sobre a medida de $\Delta \theta_k$ do girômetro já é suficiente para a estimativa consistente de $\hat{\theta}_k$.

²Somente para a Odometria

capítulo 5

Geração de Trajetória

5.1 Introdução

Para que um robô móvel possa ser autônomo, ele deve ser equipado de um planificador de trajetória, para definir um ou diversos caminhos possíveis entre o ponto de saída e do de chegada. Além disso é necessário um gerador de trajetória, para construir o movimento contínuo e compatível com as limitações do veículo e do caminho escolhido, uma vez que o número de pontos intermediários gerado pelo planificador é pequeno. Por fim, é necessária a lei de comando gerar a movimentação do robô, a partir do caminho planificado.

O gerador de trajetória funciona a partir dos pontos de passagem fornecidos pelo algoritmo de planificação da trajetória. O planificador fornece os pontos de passagem, que são representados pelo vetor $\mathbf{P_i} = (x_i, y_i, \theta_i, V_{pto}, V_{máx})^T$, no qual (x_i, y_i, θ_i) representa o ponto de passagem e $(V_{pto}, V_{máx})$ representam a velocidade no ponto e a velocidade máxima admitida entre o (i-1)-ésimo e o *i*-ésimo ponto de passagem. Com esses dados, o gerador da trajetóra gera os pontos intermediários e a velocidade de cada ponto a fim de o robô realizar um movimento contínuo e suave. O algoritmo de planejamento de trajetórias utilizando o Algoritmo de Mapa de Rotas Probabilístico está sendo desenvolvido por Adôrno [Adôrno, Aguiar e Borges 2005].

O robô Omni utiliza o gerador de trajetória baseado em curva de Bezier desenvolvido por Aragones [Aragones 2002]. Ao final da geração da trajetória global usando o planejador local, é feita a suavização usando um interpolador de trajetórias baseado em curvas de Bezier. Esse gerador é executado em tempo-real no Omni, servindo para gerar referência de configuração operacional e determinar as velocidades de referência de modo a possibilitar um movimento contínuo com derivadas finitas em todos os pontos do trajeto, resultando assim em um movimento suave.

5.1.1 Definição geométrico da trajetória

Uma vez obtido os pontos de passagem pelo planificador de trajetória dentro do espaço da postura do robô $(x, y, \theta)^T$, o gerador de trajetória gera os pontos intermediários a fim do robô realizar uma trajetória suave e contínua. Para que as distâncias euclidianas sejam coerentes, os componentes do vetor de postura devem ser homogêneas e devem estar na mesma unidade, no caso $x \in y$ em metros e θ em radianos, além de V_{pto} e $V_{máx}$ estarem em radianos por segundo.

A trajetória será composta de uma sucessão de *curvas*, que são segmentos da trajetória entre dois pontos de passagem consecutivos, como mostrado na figura 5.1. Na contrução da curva, são definidos para cada ponto, sua tangente e sua curvatura de modo a manter a continuidade da trajetória, em relação aos pontos vizinhos. Já para o ponto inicial e final, a tangente e a curvatura são definidos para ter continuide com o ponto posterior e anterior, repectivamente, sendo estas curvas um arco de círculo.

À excessão do ponto inicial e final, que possuem condições de continuidade particulares, quatro pontos de passagem intervém da geometria de uma curva. Por simplificação, a curva atual (P_{i-1}, P_i) tem a notação de (P_0, P_1) e não está situada nas extremidades da trajetória. O segmento da trajetória é calculado enquanto o robô se encontra sobre o primeiro ponto de passagem. Dado três pontos consecutivos $(\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2)$ no espaço \mathbf{R}^3 , *i.e.*, coordenadas (x, y, θ) , traça-se um único círculo pelos três pontos. A tangente e a cuvartura do ponto intermediário \mathbf{P}_1 são definidos pela sua tangente e pelo raio do centro do círculo a este ponto. Cada novo ponto de passagem permite contruir um novo círculo e então calcular suas tangentes e curvaturas ao ponto final de uma nova curva.



Figura 5.1: Interpolação da trajetória [Aragones,2002].

5.1.2 Interpolação da trajetória por curva de Bezier

O interpolador de trajetória do robô Omni se baseia nas seguintes relações: $\mathbf{V}_0 = \mathbf{P}_0 \mathbf{P}_1 = \mathbf{P}_1 - \mathbf{P}_0$

 $\mathbf{V}_1 = \mathbf{P}_1 \mathbf{P}_2 = \mathbf{P}_2 - \mathbf{P}_1$ $\mathbf{V} = \mathbf{P}_0 \mathbf{\Omega}_1 = \mathbf{\Omega}_1 - \mathbf{P}_0$, no qual $\mathbf{\Omega}$ é o centro do círculo C_i passando por \mathbf{P}_0 , $\mathbf{P}_1 \in \mathbf{P}_2$. \mathbf{T}_i é o vetor tangente ao círculo em \mathbf{P}_i

 \mathbf{C}_i é o vetor normal ao círculo em \mathbf{P}_i .

$$S^{2} = \mathbf{V}_{0}^{2} \cdot \mathbf{V}_{1}^{2} - (\mathbf{V}_{0} \cdot \mathbf{V}_{1})^{2}$$

$$\mathbf{V} = \frac{\left[\mathbf{V}_{1}^{2} \cdot \left(\mathbf{V}_{0}^{2} - \mathbf{V}_{0} \cdot \mathbf{V}_{1}\right) \cdot \mathbf{V}_{0} + \mathbf{V}_{0}^{2} \cdot \left(\mathbf{V}_{1}^{2} - \mathbf{V}_{0} \cdot \mathbf{V}_{1}\right) \cdot \mathbf{V}_{1}\right]}{2S^{2}}$$

$$Em \mathbf{P}_{0} : \mathbf{C}_{0} = \frac{\mathbf{V}}{\mathbf{V}^{2}}$$

$$\mathbf{T}_{0} = \mathbf{V}_{1}^{2} \cdot \mathbf{V}_{0} - \mathbf{V}_{0}^{2} \cdot \mathbf{V}_{1}$$

$$Em \mathbf{P}_{1} : \mathbf{C}_{1} = \frac{(\mathbf{V} - \mathbf{V}_{0})}{\mathbf{V}^{2}}$$

$$\mathbf{T}_{1} = \left(\mathbf{V}_{1}^{2} - 2\mathbf{V}_{0} \cdot \mathbf{V}_{1}\right) \cdot \mathbf{V}_{0} + \mathbf{V}_{0}^{2} \cdot \mathbf{V}_{1}$$

$$Em \mathbf{P}_{2} : \mathbf{C}_{2} = \frac{(\mathbf{V} - \mathbf{V}_{1})}{\mathbf{V}^{2}}$$

$$\mathbf{T}_{0} = \left(\mathbf{V}_{0}^{2} - 2\mathbf{V}_{0} \cdot \mathbf{V}_{1}\right) \cdot \mathbf{V}_{1} + \mathbf{V}_{1}^{2} \cdot \mathbf{V}_{0}$$

Os pontos de controle da curva de Bezier são expressos em função dos pontos de passagem e das tangentes e curvaturas dos pontos considerados, gerando assim os pontos de controle N_i (i = 1, 2..5) no espaço \mathbb{R}^3 dado por:

$$\mathbf{N}_{0} = \mathbf{P}_{0}
\mathbf{N}_{1} = \mathbf{P}_{0} + \frac{\mathbf{T}_{0}}{5}
\mathbf{N}_{2} = \mathbf{P}_{0} + \left(\frac{\mathbf{T}_{0}^{2}}{20}\right) \mathbf{C}_{0} + \left(\frac{2}{5} - \left(\frac{\mathbf{T}_{0}}{20}\right)^{2} \frac{\mathbf{C}_{0} \cdot \mathbf{V}_{1}}{\mathbf{V}_{1}^{2}}\right) \mathbf{T}_{0}
\mathbf{N}_{3} = \mathbf{P}_{1} + \left(\frac{\mathbf{T}_{1}^{2}}{20}\right) \mathbf{C}_{1} + \left(\frac{2}{5} + \left(\frac{\mathbf{T}_{1}}{20}\right)^{2} \frac{\mathbf{C}_{1} \cdot \mathbf{V}_{2}}{\mathbf{V}_{2}^{2}}\right) \mathbf{T}_{1}
\mathbf{N}_{4} = \mathbf{P}_{1} - \frac{\mathbf{T}_{1}}{5}
\mathbf{N}_{5} = \mathbf{P}_{1}$$
(5.1)

Logo, a trajetória entre os pontos $\mathbf{P}_0 \in \mathbf{P}_1$ é interpolada por um polinômio do quinto grau dado por:

$$\mathbf{P}(u) = (1-u)^{5} \cdot \mathbf{N}_{0} + 5 \cdot u \cdot (1-u)^{4} \cdot \mathbf{N}_{1} + 10 \cdot u^{2} \cdot (1-u)^{3} \cdot \mathbf{N}_{2} + 10 \cdot u^{3} \cdot (1-u)^{2} \cdot \mathbf{N}_{3} + 5 \cdot u^{4} \cdot (1-u) \cdot \mathbf{N}_{4} + u^{5} \cdot \mathbf{N}_{5}$$
(5.2)

A equação 5.2 mostra a expressão vetorial de $\mathbf{P}(u)$ e a curva é definida ao fazer u variar no intervalo de [0, 1], sabendo-se então a postura de referência do robô em função do parâmetro u.

Para definir completamente o movimento, uma *lei de velocidade curvilínea* é aplicada a cada curva.

5.1.3 Lei de velocidade curvilínea

A finalidade de se aplicar uma lei de velocidade ao sistema após a geração dos pontos da trajetória é de obter a continuidade C_2 ao movimento do robô, ou continuidade da segunda derivada. Isso significa que o ponto de interseção entre duas curvas consecutivas possuem a mesma primeira e segunda derivadas, garantindo assim continuidade do movimento.

Uma curva da trajetória é dividida em uma fase de aceleração, uma fase de velocidade constante e uma fase de desaceleração, ou lei de velocidade do tipo rampa senoidal positiva. No caso de curvas isoladas, a velocidade do veículo no ponto incial e final são nulas. Se a curva é longa, o movimento compreende em três fases: aceleração, velocidade constante e desaceleração. Senão, o movimento será composto somente por uma fase de aceleração e outro de desaceleração.

Para que o movimento seja realizável, a curva deve ter um comprimento suficiente para atingir a velocidade imposta no final a partir da velocidade do ponto inicial. Caso esta curva seja curta, implicará em uma aceleração (ou desaceleração) muito elevada e muitas vezes incompatível com as limitações do robô.

Considerando s'(t) a velocidade curvilínea no instante t, a lei de geração de movimento do tipo rampa senoidal positiva é definida pelas seguintes fases:

- I. Fase de aceleração: para $0 \le t \le \Delta t_1$ $s'(t) = \frac{V1 - Vm}{2} \cos\left(\frac{\pi \cdot t}{t_1}\right) + \frac{V1 + Vm}{2}.$
- II. Fase de velocidade constante:para $\Delta t_1 \leq t \leq \Delta t_1 + \Delta t_2$ s'(t) = Vm.
- III. Fase de desaceleração: para $\Delta t_1 + \Delta t_2 \le t \le \Delta t_1 + \Delta t_2 + \Delta t_3$ $s'(t) = Vm - \left\{ \frac{V2 - Vm}{2} \cos\left(\frac{\pi(t - \Delta t_1 - \Delta t_3)}{\Delta t_2}\right) + \frac{Vm - V2}{2} \right\}$

A duração de cada fase depende da velocidade imposta nos pontos de passagem e a aceleração máxima possível alacançada pelo robô e é dada pelas seguintes expressões:

$$\Delta t_1 = \frac{\pi}{2} \cdot \frac{|Vm-V1|}{Am}$$

$$\Delta t_2 = \frac{1}{Vm} \cdot \left\{ L_t - \frac{Vm+V1}{2} \cdot \Delta t_1 - \frac{Vm+V2}{2} \cdot \Delta t_3 \right\}$$

$$\Delta t_3 = \frac{\pi}{2} \cdot \frac{|Vm-V2|}{Am}$$
(5.3)

no qual $Am \in Vm$ são a aceleração e a velocidade máxima permitida em um segmento da trajetória e L_t é o comprimento do segmento no instante t. V1 e V2 são as velocidades no ponto inicial e

final, respectivamente, da curva. A figura 5.2 mostra os diversos tipos de perfis de velocidade possíveis.



Figura 5.2: Tipos de de perfis possíveis de velocidade em um segmento da trajetória

CAPÍTULO Ó

Desenvolvimento e Experimentação

6.1 Introdução

A realização dos experimentos teve como finalidade melhor compreender a plataforma Omni, suas características e suas limitações. Primeiramente a câmera de vídeo foi estudada, e um simulador para geração de imagens sintéticas foi projetado e implementado. Nesta implementação foi considerada tanto o modelamento da câmera quanto os sistema de coordenadas de toda a plataforma Omni, de modo que fosse considerado a movimentação do robô na geração das imagens.

A fim de melhor conhecer as bibliotecas de controle presentes na plataforma móvel Omni, um projeto para a movimentação do robô, a partir de comandos pelo joystick, para a aquisição de imagens das câmeras em configuração estéreo, para leitura dos dados dos sensores e apresentação de dados na tela foi implementado.

Por fim, foi testado uma forma alternativa de estimar o posicionamento do robô Omni. O uso de uma rede neural para imitar o funcionamento do giroscópio é testada por simulação, utilizando dados reais para treinar a rede, obtendo então a girodometria com rede neural.

6.2 Geração de Imagens sintéticas

Este experimento consistiu em projetar e implementar um gerador de imagens sintéticas a partir das configurações reais do robô Omni e da câmera *Fire-i400*. O simulador foi implementado em um programa no ambiente Matlab. Este simulador gera imagens sintéticas de uma câmera fixada no robô Omni, quando este se movimenta no espaço (x, y, θ) . O objetivo desta simulação é estudar o funcionamento da câmera, o espaço de trabalho 3-D observado por esta e o processo de geração de imagem 2-D (apresentados na seção 3.2.3). Isso acoplado à movimentação do robô e estimação do deslocamento (apresentado no capítulo 4) e à mudança de coordenadas entre diversos sistemas de referência (apresentado na seção 4.2).

6.2.1 Estimação do espaço de trabalho 3-D visto pelo sistema de coordenadas da câmera

A primeira tarefa a ser realizada consiste em estimar o espaço de trabalho 3-D observado pela câmera em seu sistema de coordenadas. O espaço de trabalho observado depende exclusivamente das características internas da câmera, representada pelos parâmetros intrínsecos, que são a distância focal vezes os coeficientes de mudança de escala de metros para pixel, $(f \cdot D_u, f \cdot D_v)$, a coordenada em pixel do centro da imagem, (u_0, v_0) , e o coeficiente de distorção radial, k. Esses parâmetros são obtidos pela calibração da câmera, que foi realizada utilizando a toolbox de Calibração de Câmeras no ambiente Matlab, desenvolvido por Jean-Yves Bouguet ¹. Os parâmetros intrínsecos encontrado para câmera *Fire-i400* são mostrados na tabela 6.1

Parâmetros	Valor Estimado	Erro asociado à estimativa
$f \cdot D_u$	1450.93191	6.08074
$f \cdot D_v$	1453.43086	6.15036
u_0	273.77180	8.17728
v_0	157.90635	9.03598
k	-0.32323	0.03862

Tabela 6.1: Parâmetros intrínsicos estimados da câmera Fire-i400.

Uma vez estimada os parâmetros intrínsecos da câmera, pode-se estimar o espaço 3-D observado pela câmera. Na simulação, supõem-se que a câmera esteja na configuração da Figura 4.2 voltada para o teto, sendo então a distância z_t^c observada (do "target") fixa. Logo, a relação entre os parâmetros intrínsecos e o espaço de trabalho observado pela câmera é mostrado na equação 6.1. O espaço de trabalho 3-D é obtido fazendo-se variar u no intervalo $[1, n_{colunas}]$ e v no intervalo $[1, n_{linhas}]$. Para cada par (u, v) tem-se a coordenada 3-D do ponto observado (x_t^C, y_t^C, z_t^C) , em relação ao sistema de coordenadas fixo na câmera, \mathcal{R}^C .

 $^{^{1}}http://www.vision.caltech.edu/bouguetj/calib_doc/$

$$\begin{pmatrix} x_t^C \\ y_t^C \end{pmatrix} = \begin{pmatrix} -f \cdot D_u & 0 \\ 0 & f \cdot D_v \end{pmatrix}^{-1} \cdot \begin{pmatrix} u - u_0 \cdot z_t^C \\ v - v_0 \cdot z_t^C \end{pmatrix}$$
(6.1)

Para obter o espaço 3-D observado pela câmera tendo o sistema de coordenadas do robô como referência, basta somente fazer a mudança de coordenadas entre o sistema de coordenadas fixo na câmera, \mathcal{R}^C , e o fixo no robô, \mathcal{R}^R , como apresentado na equação 4.6 e reescrito na equação 6.2.

$$\begin{pmatrix} x_t^R \\ y_t^R \\ z_t^R \end{pmatrix} = \begin{pmatrix} \cos\left(\theta_{C_z}^R\right) & -\sin\left(\theta_{C_z}^R\right) & 0 \\ \sin\left(\theta_{C_z}^R\right) & \cos\left(\theta_{C_z}^R\right) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \left\{ \begin{pmatrix} x_t^C \\ y_t^C \\ z_t^C \end{pmatrix} - \begin{pmatrix} x_c^R \\ y_C^R \\ z_c^R \end{pmatrix} \right\}$$
(6.2)

6.2.2 Movimentação do robô e geração da imagem sintética

Definido o espaço de trabalho 3-D da câmera em relação ao sistema de referência do robô, que não muda com a movimentação do mesmo, pode-se criar a imagem 2-D sintética a cada deslocamento (x, y, θ) do robô.

Primeiramente, obtém-se a transformação do espaço de trabalho 3-D da referência fixa no robô, \mathcal{R}^R , para o sistema de coordenadas global, \mathcal{R}^G , a partir das estimativas do posicionamento obtidas pela odometria, como mostrado na equação 6.3.

$$\begin{pmatrix} x_t^G \\ y_t^G \\ z_t^G \end{pmatrix} = \begin{pmatrix} \cos\left(\theta\right) & -\sin\left(\theta\right) & 0 \\ \sin\left(\theta\right) & \cos\left(\theta\right) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \left\{ \begin{pmatrix} x_t^R \\ y_t^R \\ z_t^R \end{pmatrix} - \begin{pmatrix} x_R^G \\ y_R^G \\ z_R^G \end{pmatrix} \right\}$$
(6.3)

no qual $x = x_R^G$, $y = y_R^G$ e z_R^G é a distância entre os dois sistemas de referência.

Uma vez obtido o espaço de trabalho 3-D observado pela câmera em coordenadas absolutas, percorre-se cada ponto na imagem, (u, v), e compara-se o seu correspondente em 3-D (x_t^G, y_t^G, z_t^G) com as coordenadas da curva observada $(x_{curva}, y_{curva}, z_{curva})$,no qual $z_t^G = z_{curva}$. Os pixels são marcados quando a distância entre dois pontos é menor que um limite determinado. A Figura 6.1 mostra a imagem gerada quando o robô realiza um deslocamento de 9,471764.10⁻¹ metros na direção x, 7,075707.10⁻² metros na direção y e 5.580611.10⁻¹ (cerca de $\pi/5$) radianos em θ . A curva observada está a 1,5 metros de distância da câmera e corresponde a uma senóide de comprimento 15, amplitude de 0,2 metros e freqüência de 2π radianos.

6.3 Controle e aquisição de imagens do robô Omni

A fim de estudar as funcionalidades da plataforma móvel Omni, foi desenvolvido um projeto usando as funções presentes nas bibliotecas de controle (Anexo 12) de tal plataforma. A trajetória deveria



Figura 6.1: Imagem Sintética.

ser controlada pelo usuário, por meio do *joystick*, e uma interface gráfica deve mostrar os dados a respeito da navegação do robô, tais como o posicionamento estimado pela odometria e as imagens capturadas pelas câmeras em configuração estéreo.

Devido às características do conjunto Windows2000/RTX 5.0, a programação do projeto de controle do robô Omni foi divido em três processos que são lançados juntos. Esses processos e suas tarefas associadas são ilustradas na Figura 6.2.



Figura 6.2: Processos que constituem o projeto.

6.3.1 Memória compartilhada

A memória compartilhada é uma das funcionalidades integradas no RTX (seção 3.3) que permite a comunicação entre diversos processos. Neste projeto ela é usada para permitir a comunicação entre processos controlados pelo RTX e os processos controlados pelo Windows 2000. Ela primeiramente precisa ser criada e/ou aberta pelos processos do projeto. Quando um processo precisa acessar a memória compartilhada para manipular as variáveis, este deve pedir o acesso exclusivo à memória

compartilhada, e somente após ser concedido o acesso exclusivo, o processo pode manipular as variáveis. Isso porque o acesso à memória compartilhada é considerada uma seção crítica, pois diversos processos podem necessitar acessar a mesma variável "simultaneamente".

6.3.2 OmniDOS:

Este processo é gerado e controlado pelo Windows 2000 utilizando funções tempo real a partir RTAPI_W32. Suas funções principais são servir de interface com o usuário, por meio da amostragem das imagens adquiridas na tela, armazenar os dados referentes à odometria em um arquivo Matlab, além de salvar as imagens capturadas, para posterior tratamento de dados off-line. Todas essas tarefas são realizadas por uma só função principal, Main, que é executada à cada período de aproximadamente 200 ms. Em sua inicialização, os outros processos são lançados a memória compartilhada e o mutex para acesso a esta seção crítica são criados.

A cada período são lidos da memória compartilhada os dados providos dos sensores, as estimativas de posicionamento dadas pela odometria e o último par de imagens capturado pelas câmeras em configuração estéreo. As imagens são salvas em arquivo a cada período de amostragem e os dados são armazenados na memória, até que sejam salvos no final da execução do processo.

Como neste processo não há nenhuma tarefa crítica sendo executada, é dada a prioridade mínima a este processo (RT_PRIORITY_MIN), para que a sua execução não interfira nos processos com tarefas crítica, como a tarefa de controle.

6.3.3 VisaoStereo:

Esse processo é controlado pelo Windows 2000 utilizando funções tempo real a partir RTAPI_W32, uma vez em que a biblioteca utilizada não permite aquisição de imagens em tempo real. Sua função é fazer a aquisição de um par de imagens das câmeras em configuração estéreo e converter a imagem para um formato compatível com a biblioteca OpenCV, no qual suas funções devem ser utilizadas para o tratamento das imagens, em trabalhos futuros. A aquisição das imagens é feita de forma seqüencial e o tempo para aquisição de cada imagem é cerca de 35 ms. Para o caso no qual o robô está se movimentando lentamente, pode-se considerar que o par de imagens são capturadas simultaneamente, que é o esperado. O procedimento total, de aquisição do par de imagens e conversão destas para o formato compatível com o OpenCV, é realizado em cerca de 150 ms.

O processo *VisaoStereo* é composto por apenas uma função principal, *Main*, com uma tarefa que acessa a cada 10 *ms* a memória compartilhada a fim de obter informação a respeito da requisição de imagens por outros processos. Por ser um processo controlado pelo *Windows 2000*, o tempo de aquisição de imagens é não determinístico e pode variar bastante em torno da média, que é de 150 *ms*. Por isso, é associado prioridade máxima a este processo (RT_PRIORITY_MAX) como uma

tentativa de minimizar o efeito do não determinismo de tal tarefa.

6.3.4 OmniRTX:

Esse processo é controlado pelo RTX e é o processo mais importante, uma vez que as tarefas submetidas à restrições tempo-real são executadas aqui, tais como controle e requisição de imagens. As tarefas que compõem este processo são:

 Tarefa main: é a tarefa que se ocupa da inicialização da plataforma Omni e da abertura da memória compartilhada e do Mutex, para seu posterior acesso. Em tal implementação, são usados o joystick, para a geração de movimento, e dos codificadores ópticos absolutos e incrementais, além do girômetro laser, para a estimação do posicionamento por odometria. Todos esses sensores são inicializados a partir da inicialização do módulo de controle (AssrvInit()). Então, foi determinado o modo de navegação como sendo manual(AssrvSetMode(ModeOperateurVeh)), ou seja, controlada pelo joystick. São inicializadas também, todas as variáveis globais usadas nesse processo, tais como as variáveis que armazenam a estimação do posicionamento do robô por Odometria.

De fato, *main* é uma tarefa de fundo, com prioridade baixa (RT_PRIORITY_MIN-1) que acorda a cada cerca de 5 *ms* para fazer a leitura do comando vindo pelo *joystick* e os enviar aos motores do robô Omni.

• Tarefa *TaskOdomUpdate*: é a tarefa tempo-real de mais alta prioridade (RT_PRIORITY_MAX-1). Essa tarefa controla a execução dos procedimentos de comando baixo nível dos motores do robô, a aquisição de dados do Girômetro laser e dos codificadores ópticos, e realiza a estimação do posicionamento. Ela se executa a cada 5 *ms*.

A principal variável desta tarefa é a variável Pose do tipo POSITION_2D, que armazena os dados estimados pela Odometria sobre o posicionamento do robô, além da estimação do erro associado à esta estimação. Após a estimação, estes dados são alocados na memória compartilhada, em uma fila circular. A estrutura POSITION_2D é mostrada na Figura 6.3. A estimação por Odometria foi feita pelo método apresentado na seção 4.4, por meio do comando *Odometry_UpdatePoseGyrometer* do módulo **Odometry.h**.

Como a variável *Pose* é uma variável global, ela pode ser acessada por qualquer tarefa e assim ser modificada por qualquer tarefa. Para impedir que duas tarefas modifiquem esta variável ao mesmo tempo, foi criado um *Mutex* para delimitar a seção crítica, que é a região onde esta variáveis é acessada para escrita e/ou leitura.

• Tarefa *TaskImageRequest*: Essa tarefa é realizada a cada 200 *ms* com uma prioridade alta (RT_PRIORITY_MAX-2), e sua finalidade é fazer requisição de um par de imagens do conjunto de câmeras em configuração estéreo. Isso é feito definindo a variável compartil-

Estrutura da variável com o resultado da Odometria	
typedef struct{ double X; double Y; double Theta; double Cov[3][3]; double Time; } POSITION_2D, "PPOSITION_2D;	

Figura 6.3: Estrutura da variáveis que armazena os Dados estimados pela Odometria.

hada *pOmniSharedData->Requests.GrabVideosSingleImage* na memória compartilhada com o valor 1.

6.4 Simulação da Girodometria com rede neural

Uma fonte de incerteza para a odometria pode estar no próprio modelo cinemático. Para minimizar o erro de modelamento, duas alternativas podem ser discutidas. A primeira consiste em aumentar a complexidade do modelo procurando incluir outros efeitos, como por exemplo as áreas de contato das rodas com o solo, ou mesmo viés sobre os ângulos de orientação β . Esta alternativa não é seguida neste trabalho, uma vez que existe um aumento do número de parâmetros a serem estimados, podendo haver forte acoplamento entre eles, o que pode tornar o resultado da estimação sem sentido físico (*e.g.*, raio negativo)([Aguiar, Oliveira e Borges 2005]).

A segunda alternativa consiste em empregar um aproximador geral de funções, tal como uma rede neural multilayer perceptron, visando realizar o mapeamento $\Delta \mathbf{q}_k \implies \hat{\mathbf{z}}_k - \hat{\mathbf{z}}_{k-1}$, como mostrado na Figura 6.4. Como este mapeamento é de \Re^6 em \Re^3 , faz-se necessário ter um grande número de dados experimentais para treinar apropriadamente a rede [Haykin 1998]. Uma formulação diferente pode fazer uso da equação da girodometria (4.28), em que ω_s seria provida não por um giroscópio, mas sim por uma rede neural que realizaria o mapeamento $\Delta \mathbf{q}_k \Longrightarrow \omega_k$, ou seja \Re^6 em \Re . Desta forma, o problema fica mais simplificado se for usada uma rede neural para imitar o giroscópio. Esta idéia se justifica pelo fato de grande parte do erro de estimação por odometria ser associado à uma má estimação de θ , conforme discutido na Seção 4.5.



Figura 6.4: Mapeamento utilizando rede neural.

O parâmetro de projeto para a girodometria com rede neural consiste no número de neurônios

na camada oculta. Como treinamento a rede recebe entradas $\Delta \mathbf{q}_k$, medidas pelos sensores de variáveis do espaço articular, e saídas desejadas a medições ω_k providas pelo giroscópio laser, como detalhado na subseção seguinte.

6.4.1 Estrutura da Rede Neural

A Figura 6.5 mostra o grafo arquitetural de um rede neural não-linear perceptron de múltiplas camadas (MLP) com uma camada oculta e uma camada de saída. Esse modelo MLP apresenta aproximações globais, ou generalizações, em regiões do espaço de entrada pouco exploradas na fase de treinamento.

Na entrada desta rede tem-se as medidas providas dos sensores proprioceptivos, que são três codificadores ópticos incrementais, que fornecem as velocidades de cada roda e o deslocamento relativo da mesma, e três codificadores ópticos absolutos, que fornecem a angulação de cada roda. Como saída da rede, na fase de treinamento, ou adaptação dos pesos, tem-se o valor medido pelo Girômetro à laser do ângulo de posição do robô móvel. O treinamento desta rede caracteriza um procedimento de mapeamento não-linear.

A dimensão da camada oculta foi tomada como um parâmetro do projeto. Em testes, a quantidade de neurônios em tal camada foi vaiada de 9 à 50 neurônios. A função de ativação em cada camada escolhida foi a função sigmóide logística dada pela seguinte relação:

$$\varphi\left(\upsilon\right) = \frac{1}{1 + \exp\left(-a\upsilon\right)} \tag{6.4}$$

onde a é o parâmetro de inclinação da função sigmóide.

O procedimento de treinamento consiste em utilizar o método de otimização Levenberg-Marquardt aplicados à rede neural para atualizar os pesos de modo a minimizar a função custo.

6.4.2 Treinamento da rede

O aprendizado da rede resulta das muitas apresentações de um determinado conjunto de exemplos de treinamento para o perceptron de múltiplas camadas. Uma apresentação completa do conjunto de treinamento inteiro é denominado de *uma época*. Na etapa de projeto da rede neural, a quantidade de épocas limite foi variada de modo a fazer com que a rede apresente uma boa característica de generalização. Ao final, fixou-se em 10 a quantidade de épocas limite. Outro fator limitante no treinamento é a convergência do erro, e essa situação é alcançada quando a taxa absoluta de variação do erro médio quadrado por época for suficientemente pequena. No projeto, foi limitado o erro a $1e^{-10}$.

O vetor de pesos \mathbf{w} foi inicializado com valores fixos, de modo a verificar a influência de cada parâmetro da rede, principalmente a influência do número de neurônios na camada oculta, sobre


Figura 6.5: Grafo estrutural da rede perceptron multicamadas utilizada para estimar o ângulo de posição do robô móvel.

a resposta obtida pela mesma quando novos dados era apresentados.

A escolha do conjunto de dados de treinamento foi feita levando em conta as seguintes características da trajetória: a trajetória tinha que ser complexa o suficiente à permitir a realizar rotações completas tanto no ângulo de direção quanto no ângulo de posição. Além disso, a velocidade teria que variar bastante durante a realização da trajetória. Baseados nestas característica, foram escolhidas três trajetórias, totalizando em 66841 dados de entrada para a rede e estas trajetórias foram adquiridas na tese de Borges [Borges 2002]. Esses dados foram apresentados à rede em seqüência aleatória, de forma a reduzir a probabilidade da rede estabilizar em um mínimo local da função de custo.

6.4.3 Resultados Experimentais

Para realizar o treino da rede neural, dados experimentais foram adquiridos com a plataforma móvel Omni em diferentes ambientes executando trajetórias simples e complexas em um ambiente real. Na Figura 6.6 estão apresentadas três trajetórias, 6.6(a) mais suave, e 6.6(b-c) mais ricas em movimentos rápidos e de rotação. Ainda na Figura 6.6 está ilustrado o posicionamento do robô, indicado por um quadrado, estimado por girodometria. Em cinza claro estão mostrados pontos de imagens do radar a laser do Omni, superpostas também usando girodometria. Pode-se perceber por meio da aparentemente correta superposição das imagens que a girodometria se apresenta como um método confiável de localização. De fato, este método é confiável desde que a duração do experimento seja curta. Vale também ressaltar que os parâmetros geométricos usados pela



Figura 6.6: Trajetórias usadas no treinamento da rede neural. O posicionamento do robô, indicado por um quadrado, foi estimado por girodometria. Em cinza claro estão mostrados pontos de imagens do radar a laser do Omni, superpostas também usando girodometria.

girodometria são os mesmos medidos com régua milimétrica, mostrados na equação 4.9.



Figura 6.7: Trajetória usada na validação da Girodometria com rede neural.

Na girodometria com rede neural, foram realizados treinos para duas redes distintas: uma com 9 e outra com 50 neurônios na camada oculta. Esta diferença de neurônios visa verificar a influência do número de neurônios da camada oculta no resultado da estimação.

Para validar os métodos, um outro experimento foi utilizado. Neste, ilustrado na Figura 6.7, o robô Omni executou uma trajetória mais extensa, fechada, quase retornando à posição inicial. Os resultados obtidos pelos diferentes métodos são apresentados a seguir. A análise considera a girodometria como método de referência, devido à sua precisão superior.

Na Figura 6.8 é mostrada a trajetória gerada pela girodometria, em vermelho, e a trajetória de validação gerada pela odometria usando os parâmetros iniciais dados por (eq. 4.9). Pode-se perceber o aumento do erro de posicionamento com relação à girodometria. As Figuras 6.8 e 6.9

apresentam os resultados para o método apresentado. Os resultados do método de girodometria com rede neural não foram satisfatórios, conforme mostram as Figuras 6.9(a) e 6.9(b). Por estas figuras, percebe-se um erro inicial na estimação de ω_k que certamente re-orientou o restante da trajetória. Se não fosse esse o caso, a rede neural com 50 neurônios apresentou o mesmo tipo de curva fechada da girodometria, bem melhor do que a rede com 9 neurônios. Este resultado era esperado, dada a complexidade de mapeamento proporcionada pela rede maior. Mas, mais uma vez, o erro ocorrido na estimação de ω_k pode ser devido à pouca exploração de dados de treino nas condições do erro. Esse exemplo ilustra a vantagem do modelo conhecido *a priori* (modelamento cinemático) sobre o modelo caixa preta.



Figura 6.8: Comparação entre girodometria (vermelho) e odometria usando os parâmetros geométricos iniciais (azul).



Figura 6.9: Comparação entre girodometria (vermelho) e girodometria com rede neural com: (a) 9 neurônios na camada oculta e (b) 50 neurônios na camada oculta (azul).

capítulo 7

Conclusões Parte I

A primeira parte do manuscrito foi reservada para a apresentação de um estudo sobre a plataforma móvel Omni, presente no Laboratório de Controle e Visão por Computador, *LCVC*, para fins de pesquisa. Sua apresentação foi dividida em quatro partes: a classificação geral de plataformas móveis, a descrição da plataforma Omni em estudo e dos seus componentes integrantes, o estudo do modelamento cinemático do robô, sua Odometria e Girodometria, e, por fim, a apresentação das simulações feitas e dos experimentos realizados nesta plataforma.

Para que se possa projetar uma plataforma móvel, há a necessidade de saber a priori em qual tipo de ambiente tal plataforma vai se locomover e qual o tipo de tarefa esta vai desempenhar. A partir destas informações, são escolhidos os sensores mais apropriados, a configuração mecânica mais apropriada para o terreno específico, as especificações mínimas de *hardware* e *software* para que os algoritmos de controle sejam desenvolvidos e quais técnicas são mais indicadas para que o robô móvel seja capaz de cumprir a missão proposta. Foram apresentados dois tipos principais de ambiente, o interno e o externo, sendo estes subdivididos em estruturados e não estruturados. Para cada ambiente específico, foram apresentados os requerimentos básicos de uma plataforma móvel para que esta desempenhe tarefas em cada ambiente, e exemplos foram mostrados de plataformas com configurações diversas que desempenhavam suas tarefas em ambientes similares.

Em seguida, foi apresentada a plataforma móvel em estudo, o robô omnidirecional Omni. Tal robô foi projetado para se locomover em ambientes internos estruturados e executar de modo autônomo tarefas complexas. As particularidades da plataforma que a permite ter um ótimo desempenho em suas missões foram mostradas no Capitulo 3. O modelamento cinemático é apresentado no Capítulo 4, juntamente com a Odometria e Girodometria, métodos utilizados para estimar o deslocamento do robô a partir da leitura dos sensores tais como codificadores ópticos e girômetro.

Por fim foram apresentados experimentos realizados no robô Omni a fim de testar as bibliotecas presentes. Foi comprovada a eficiência de tais bibliotecas e a robustez e versatilidade da plataforma Omni, uma vez que uma infinidade de trabalhos pode ser realizada sem que haja necessidade de um conhecimento profundo sobre a plataforma em si. Por meio de simulações, fez-se um estudo sobre o processo de aquisição de imagens com o robô em movimento e a possibilidade de usar uma rede neural para emular o girômetro a laser. Todas as implementações foram consideras satisfatórias e acrescentaram não só para o desenvolvimento deste trabalho como pode vir a auxiliar em projetos futuros.

Espera-se que o trabalho desenvolvido aqui venha a ser uma ferramenta de auxílio para aqueles que desejam desenvolver projetos na plataforma Omni, servindo de bibliografia inicial para o conhecimento do robô e que os experimentos sirvam de exemplo e ponto de partida para melhoramentos.

Parte II

Calibração de câmeras para a localização 3-D: Aplicação na estimação do movimento cardíaco

CAPÍTULO 8

Atualidade em Robótica Médica

8.1 Robótica médica

Na medicina, o século 20 foi o século da cirurgia. O tratamento de doenças que antes podiam levar o paciente à morte, como apendicite, hoje se tornaram uma rotina clínica. A expectativa é que o século 21 seja o século do desenvolvimento de técnicas de micro acesso em procedimentos cirúrgicos, reduzindo dramaticamente o trauma sofrido pelo paciente. Nesta nova fase, cirurgias auxiliadas por computadores e robôs levarão a uma revolução nos procedimentos cirúrgicos e na própria formação dos profissionais da área [Ortmaier 2003].

Nos últimos anos, diversos robôs cirúrgicos que foram desenvolvidos em institutos de pesquisa foram usados em hospitais para experimentos ou até mesmo em procedimentos cirúrgicos rotineiros. RobodocTM do Integrated Surgical Systems Inc.¹ e o CasparTM do URS Universal Robot Systems² são usados em cirurgias ósseas, enquanto o sistema daVinciTM do Intuitive Surgery Inc. [Guthart e Salisbury 2000] e o sistema ZeusTM (figura 8.1) do Computer Motion Inc³ foram desenvolvidos para realizarem cirurgias minimamente invasivas. O número de aplicações clínicas com auxílio de tais sistemas tem crescido devido a vantagens tais como alta precisão, o planejamento pré-cirúrgico, e a possibilidade de novas técnicas cirúrgicas, como a cirurgia cardíaca minimamente

 $^{^{1}}$ www.robodoc.com

 $^{^2}$ www.urs-groups.com/orth/

³www.computermotion.com

invasiva. As desvantagens de cirurgias utilizando sistemas robóticos são o aumento do tempo em treinamento de especialistas e no ato cirúrgico, alto custo, e as restrições nas aplicações das técnicas. Em neuro cirurgias, por exemplo, o espaço de trabalho é bastante limitado e a precisão dos movimentos requerida é muito alta. Por outro lado, cirurgias ósseas requerem robôs que forneçam grande força, logo, robôs industriais são modificados para tal aplicação. Para lidar com tecidos macios tais como em cirurgias abdominais e cardíacas, há necessidade de menos força na ponta do robô, mas o espaço de trabalho é maior que nas aplicações anteriores, além de que tecidos se deformam quando em contado com a ponta do robô e variam de cor e tamanho com o movimento gerado a partir da respiração e da vazão sangüínea.



Figura 8.1: Sistema Zeus: a console de comando é composta do robô mestre e da tela de vídeo.



Figura 8.2: Exemplos de instrumentos cirúrgicos da $EndoWrist^{TM}$.

8.2 Cirurgias robóticas minimamente invasivas

Cirurgia minimamente invasiva é uma técnica de operação criada na década de 1980. A maior diferença com a técnica de cirurgia aberta é que o cirurgião trabalha com instrumentos longos, que são inseridos no paciente por meio de pequenas incisões (normalmente menor que 10 mm), e são movidos por meio de um sistema mestre/escravo controlado pelo médico. Usualmente, o cirurgião não tem acesso à parte do corpo operado, ao contrário do que acontece na técnica de cirurgia aberta. Geralmente, quatro pequenas incisões são necessárias: duas para os instrumentos cirúrgicos, um para o endoscópio e um para insuflação CO_2 , como mostrado na Figura 8.3⁴.



Figura 8.3: Configuração dos aparelhos em cirurgias minimamente invasivas.

As principais vantagens da técnica de cirurgia minimamente invasiva sobre a cirurgia aberta são: pequenas incisões que fazem com que a dor e o trauma do paciente sejam reduzidos, a estadia do paciente no hospital é reduzida, assim como o período de reabilitação. Dentre as desvantagens temse que o uso de endoscópios pode fazer com que o cirurgião perca a orientação durante o ato cirúrgico ou não veja os instrumentos no campo de visão observado pelo endoscópio. Outro problema é a falta de sensibilidade sobre a textura e consistência da superfície do tecido manipulado pelos instrumentos controlados pelo médico, devido ao fato desse último não ter acesso à área no qual há a intervenção cirúrgica. Mas o problema principal, e considerado o mais difícil de ser eliminado, é do deslocamento e deformação de órgãos e tecidos macios envolvidos no procedimento cirúrgico. Esses movimentos são ditos ser fisiológicos, uma vez que são induzidos pela respiração e pelo batimento cardíaco. A respiração causa um deslocamento complexo e repetitivo em praticamente todos os órgãos na região torácica e esse deslocamento não é compensado pelos sistemas robóticos cirúrgicos tradicionais, sendo então encargo do cirurgião fazer tal compensação.

A fim de minimizar ou até mesmo eliminar essas desvantagens da técnica de cirurgia minamente

⁴www.herrmann-gruenstadt.de/Mis.htm

invasiva são desenvolvidos sistemas robóticos com intuito de auxiliar o cirurgião na intervenção cirúrgico. A Figura 8.4⁵ mostra um sistema robótico comercial usado para este fim. Os sistemas robóticos utilizados para fins médicos são classificados de acordo com o tipo de interação no qual o cirurgião dispõe para guiar estes sistemas [Krupa 2003]:

- Sistemas de guiagem passiva: são ditos passivos porque o cirurgião guia pessoalmente os instrumentos com toda a liberdade e são principalmente o sistema de navegação que permite ao cirurgião visualizar os instrumentos em tempo real e a trajetória que este realiza;
- Sistemas de guiagem ativa: são capazes de acionar diversos instrumentos cirúrgicos simultâneos de maneira autônoma a fim de realizar uma intervenção cirúrgica, a partir do planejamento pré-cirúrgico, com a supervisão do médico;
- Sistemas de guiagem semi-ativas: os instrumentos são capazes de seguir uma trajetória planejada na fase pré-cirúrgica até a área de trabalho e então o cirurgião controla os instrumentos nesta região, com um certo grau de liberdade definido;
- Sistemas sinergéticos: são sistemas de guiagem semi-ativas com o grau de liberdade programado dependente do ponto de operação. Permite que os instrumentos se locomovam com todos os graus de liberdade em uma dada região e com restrições em outras regiões;
- Sistemas telecirúrgicos: são os sistemas robóticos do tipo *mestre/escravo*. O cirurgião comanda a distância por intermédio de uma *console*, um manipulador escravo que aciona os motores a fim de realizar o deslocamento dos instrumentos cirúrgicos.
- Sistemas robóticos endocorporais: são compostos de micro sistemas robóticos autônomos que são introduzidos no interior do corpo do paciente pelas vias naturais para ter acesso a locais de difícil acessibilidade. Atualmente, esses sistemas são usados para realizar exames como, por exemplo, coloscopia (visualização interna do cólon).

8.2.1 Cirurgias robóticas minimamente invasivas - compensação do movimento causado pela respiração e batimento cardíaco

Em cirurgias robóticas minimamente invasivas, um dos maiores desafios encontrados no projeto de sistemas robóticos para este fim é fazer a compensação dos movimentos dos órgãos causados pela respiração e do batimento cardíaco. A maior vantagem da compensação de movimento pelo sistema robótico é requerer menos especialização por parte do médico, e com isso popularizar a técnica cirúrgica, além de proporcionar atos cirúrgicos mais precisos.

Diversos outros trabalhos foram desenvolvidos a fim de fazer a compensação de movimento em tecidos macios. Os primeiros resultados encontrados são sobre a compensação do movimento gerado

 $^{^{5}} www.stronghealth.com/.../surgical/davinci.cfm$



Figura 8.4: Sistema robótico usado em cirurgias minimamente invasivas.

devido a respiração, uma vez que este movimento é periódico, e apesar de ser complexo, pode ser compensado com um controle preditivo. Em [Schweikard et al. 2000] foi exposto um trabalho em terapia de câncer que apresentou um sistema que realizava a compensação de movimentos devido à respiração em tecidos macios. Em [Ginhoux et al. 2003] foi apresentado um sistema de servo visão para a sincronização do robô com o movimento de um fígado devido à respiração.

Os movimentos devido ao batimento cardíaco ainda encontram limitações nas técnicas de compensação de movimento tradicionais uma vez que eles são complexos e rápidos e são criados sobre a influência tanto do movimento cardíaco quanto das atividades eletro-mecânica do coração. É necessário então um sistema de medida para estimar precisamente o deslocamento da superfície do coração com toda a sua dinâmica. É necessário também rastrear o movimento estimado com o robô por meio de leis de controle adequado que leve em conta a dinâmica do robô. Poucos estudos foram realizados com o intuito de fazer a compensação do movimento cardíaco.

Ortmaier em [Ortmaier 2003, Gröger et al. 2002] desenvolveu um sistema robusto de rastreio de marcos naturais na superfície do coração observado por um endoscópio e compensação do movimento é feito por meio de algoritmo de controle preditivos que envia os pontos de referência para o robô comercial Zeus. Nakamura em [Nakamura, Kishi e Kawakami 2001, Nakamura et al. 2000] usou câmeras de vídeo rápida para amostrar precisamente o movimento cardíaco e este é estimado a partir de marcos artificiais colocados na superfície do coração. O robô é controlado usando a teoria de controle H_{∞} . Krupa em [Krupa 2003, Krupa et al. 2003] propôs um sistema em que os instrumentos cirúrgicos são posicionados automaticamente no espaço de trabalho e é feita a compensação do movimento cardíaco a partir de imagens e feixes a laser direcionados a área de interesse. Ginhoux em [Ginhoux et al. 2003] apresenta o controle preditivo para compensar os movimentos cardíacos. Todos estes autores validam as técnicas propostas por meio de experimentos *in vivo* realizados em porcos vivos submetidos a condições cirúrgicas.

8.3 O projeto MAX

Como apresentado na seção 8.2, uma das grandes dificuldades em se realizar a compensação do movimento cardíaco está na estimativa precisa do movimento cardíaco. Por isso, grande parte do esforço focaliza no problema de se encontrar a melhor técnica para se fazer a estimativa do deslocamento e qual é o erro associado a tal estimativa.

Ortmaier em [Ortmaier 2003, Gröger et al. 2002], por exemplo, estima o movimento cardíaco a partir de marcos naturais observados por um endoscópio e sua medida é feita em *pixel* e por isso no seu trabalho não há informação de profundidade, nem amplitude do deslocamento em metros. Neste trabalho, são colados estabilizadores mecânicos no coração para limitar o seu movimento. Nakamura em [Nakamura, Kishi e Kawakami 2001, Nakamura et al. 2000] faz a estimação do movimento cardíaco a partir de um scanner a laser 3-D e informações obtidos pelas imagens adquiridas pelo endoscópio da área de interesse dos feixes laser observados na superfície do coração, obtendo assim informações tridimensionais sobre o movimento. A maior limitação desta técnica é a dificuldade encontrada de se implementar o sistema em tempo real. Uma técnica similar é proposta por Krupa em [Krupa 2003, Krupa et al. 2003]. Ele observa na imagem os feixes laser projetados na superfície do órgão e o emissor do feixe com leds fixados em sua estrutura e utiliza as informações geométricas conhecidas para fazer uma estimativa de profundidade e do deslocamento realizado em unidades métricas. Esse sistema é implementado para que se executar em tempo real e com erro de estimação limitado.

Com o objetivo de se projetar um sistema robótico cirúrgico capaz de auxiliar em cirurgias cardíacas, o LIRMM, *Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier*, lançou o projeto MAX, e foi neste contexto que o trabalho aqui apresentado foi desenvolvido. Esse projeto prevê a elaboração, construção e implementação de um sistema robótico cirúrgico para a realização de cirurgias cardíacas. Ele é divido em duas partes: a parte de robótica e a parte de visão computacional.

Na primeira parte foi projetada e construída uma plataforma robótica, o D2M2, um braço manipulador de 5 graus de liberdade capaz de controlar os instrumentos cirúrgicos e algoritmos de controle estão sendo desenvolvidos para compensar os movimentos cardíacos. O controle dos instrumentos cirúrgicos com vários graus de liberdades estão sendo projetados.

Na segunda parte, é feito o modelamento do coração a partir de imagens de vídeo e de informações sobre textura, e são estudadas técnicas de controle a partir de imagens de vídeo, além de técnicas de estimação do movimento cardíaco a partir das imagens.

O trabalho aqui apresentado foi desenvolvido durante o estágio no LIRMM e é referente à segunda etapa, mais especificamente sobre a calibração de câmeras para a estimação de movimento

a partir de imagens de vídeo. O objetivo é determinar as coordenadas 3-D de uma certa região de interesse na superfície do coração a partir de imagens 2-D obtida a partir de uma única câmera. Devido ao mapeamento projetivo, é geralmente impossível determinar as coordenadas 3-D de um certo objeto a partir de uma única imagem 2-D. Porém, ao adicionar um conjunto condições ao arranjo da imagem, reconstrução parcial ou completa se torna possível.

A técnica proposta para se estimar o movimento cardíaco foi utilizar uma câmera calibrada. Uma vez calibrada a câmera, na qual são estimados os parâmetros intrínsecos que descrevem as características internas à câmera, pode usar estes parâmetros para fazer se obter informações 3-D a partir de uma única imagem. Um software foi desenvolvido em linguagem C a fim de possibilitar a estimação do posicionamento 3-D a partir de imagens em tempo real. Para tal estimação, teve que se levar em conta o ambiente no qual são adquiridas as imagens, que é a superfície do coração, uma região viscosa, macia e deformável.

capítulo 9

Calibração de câmeras

9.1 Introdução

A calibração de câmeras é uma tarefa de extrema importância em visão computacional, uma vez que este é um passo essencial para que se possa extrair informações do mundo tridimensional a partir de dados bidimensionais extraídos de uma imagem.

A calibração de uma câmera consiste em estimar os valores dos parâmetros que descrevem o modelo da câmera [Faugeras 1993]. Estes parâmetros são divididos em dois grupos: os intrínsecos, que descrevem as características ópticas da câmera, e os extrínsecos, que relacionam a posição e a orientação do sistema de coordenadas da câmera em relação ao sistema de coordenadas global considerado.

O modelo de câmera considerado será o modelo *pinhole*, um modelo simplificado no qual a formação da imagem se dá a partir da passagem de raios refletidos pela imagem observada por um ponto, como mostrado na Figura 9.1. Este modelo é util pois permite uma formulação matemática simples para a relação entre objeto observado e a imagem formada. Porém, ele não é válido quando há necessidade de alta precisão nas estimativas. Por isso este modelo é estendido para que se considere as deformações causadas na imagem por causa de distorções.



Figura 9.1: Formação da imagem a partir do modelo pinhole.

9.2 Parâmetros intrínsecos

Desconsiderando as componentes não lineares das distorções radiais e tangenciais, e considerando o modelo *pinhole* (figura 9.1), a relação entre o plano da retina ou coordenadas da câmera (u, v), em *pixel*, e as coordenadas 3-D (X, Y, Z), em metros,6 como mostrado na Figura 9.2, pode ser escrita como:

$$-\frac{f}{Z} = \frac{u}{X} = \frac{v}{Y} \tag{9.1}$$

A equação 9.1 pode ser reescrita na forma matricial,

$$\begin{pmatrix} U \\ V \\ S \end{pmatrix} = \begin{pmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$
(9.2)

na qual $u=\frac{U}{S}$ e $v=\frac{V}{S}$ para $S\neq 0,$ sendo S um fator de escala.

A equação 9.2 mostra o fato que a relação entre as coordenadas da imagem e do espaço 3-D é linear em coordenadas projetivas [Faugeras 1993] e pode ser expressa na forma matricial como

$$\mathbf{m} = \mathbf{P}\mathbf{M}$$

no qual \mathbf{P} é a matriz de projeção, que é definido a menos de uma escalar, \mathbf{m} é o vetor homogêneo



Figura 9.2: O plano focal (x, y) é paralelo ao plano da retina(u, v) e a distância entre esses planos é f, no qual f é a distância focal.

das coordenadas em *pixel* da imagem e M é o vetor homogêneo das coordenadas no sistema global.

Uma câmera pode ser considerada como um sistema que realiza a transformação linear projetiva do espaço projetivo \mathcal{P}^3 para o plano projetivo \mathcal{P}^2 . Relações lineares e algebra linear são usadas ao invés de equações não lineares.

As coordenadas correspondentes na imagem \mathbf{m} (u', v'), em *pixels*, de um ponto \mathbf{M} (X, Y, Z)são obtidas pela projeção de (u, v), como mostrado na equação 9.3:

$$\begin{pmatrix} u'\\v' \end{pmatrix} = \begin{pmatrix} D_u s_u u\\D_v v \end{pmatrix} + \begin{pmatrix} u_0\\v_0 \end{pmatrix}$$
(9.3)

Na equação 9.3 tem-se que D_u e D_v são coeficientes usados para a mudança da unidade de medida de metros para *pixels*. O coeficiente s_u é um fator de escala, uma vez que os dois eixos (u, v) na imagem não são iguais. Os coeficientes u_0 e v_0 correspondem à coordenada do centro da imagem, também chamado de ponto principal.

A partir das relações expressas nas equações 9.2 e 9.3, pode-se obter a equação que relaciona as coordenadas na imagem (u', v') com as coordenadas 3-D correspondentes (X, Y, Z), como mostrado na equação 9.4.

$$\begin{pmatrix} U'\\V'\\S' \end{pmatrix} = \begin{pmatrix} -fD_u s_u & 0 & u_0 & 0\\ 0 & -fD_v & v_0 & 0\\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X\\Y\\Z\\1 \end{pmatrix}$$
(9.4)

Na equação 9.4 tem-se que $u' = \frac{U'}{S'}$ e $v' = \frac{V'}{S'}$ para $S \neq 0$.

Os parâmetros f, D_u , D_v , $u_0 \in v_0$ não dependem da posição e da orientação da câmera no espaço. Eles são associados às características físicas da câmera, e são então chamados de parâmetros intrínsecos.

9.3 Parâmetros extrínsecos



Figura 9.3: Mudança de sistemas de coordenadas no espaço 3-D.

A fim de expressar um ponto arbitrário \mathbf{M} , com as coordenadas (X_0, Y_0, Z_0) no sistema de coordenadas global centrado em O, em coordenadas da imagem centrada em c, é necessário primeiramente fazer a transformação para o sistema de coordenadas da câmera ou padrão (X, Y, Z) centrado em C(figura 9.3). Esta transformação consiste em uma rotação \mathbf{R} (rolagem, cabeceio e guinada) ao redor do centro de projeção C da câmera, representado por uma matriz 3×3 , seguido pela matriz de translação \mathbf{T} , representado por uma matriz 3×1 , como mostrado na equação 9.5.

$$\begin{pmatrix} X \\ Y \\ Z \\ S \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{pmatrix}$$
(9.5)

Em coordenadas projetivas, a equação 9.5 é escrita como

$$\mathbf{M}_C = \mathbf{K}\mathbf{M}_O$$

O índice nos vetores das coordenadas dos pontos \mathbf{M} indica qual o sistema de coordenadas estes pontos referem.

As matrizes $\mathbf{R} \in \mathbf{T}$ descrevem a posição e orientação do sistema da câmera com respeito ao sistema de coordenadas globais. Como estas matrizes dependem da configuração da câmera e do sistema de referência global escolhido, os parâmetros que descrevem a rotação e translação são chamados de parâmetros extrínsecos.

9.4 Modelo completo da câmera

Considerando o modelo *pinhole* da câmera, os parâmetros intrínsecos e extrínsecos descritos na seções 9.2 e 9.3, respectivamente, pode-se fazer a correspondência entre um ponto **M** com coordenada (X_0, Y_0, Z_0) no espaço tridimensional (sistema de coordenadas global) e seu equivalente na imagem com a coordenada (u', y'), como mostrado na equação 9.6.

$$\begin{pmatrix} U' \\ V' \\ S' \end{pmatrix} = \begin{pmatrix} -fD_u s_u & 0 & u_0 & 0 \\ 0 & -fD_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}_3^T & 1 \end{pmatrix} \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{pmatrix}$$
(9.6)

A equação 9.6 pode ser reescrita como:

$$\begin{pmatrix} U'\\V'\\S' \end{pmatrix} = \begin{pmatrix} -fD_us_u & 0 & u_0\\0 & -fD_v & v_0\\0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{T} \end{pmatrix} \begin{pmatrix} X_0\\Y_0\\Z_0\\1 \end{pmatrix}$$
(9.7)

Na equação 9.7, \mathbf{r}_i representa a *i-ésima* coluna da matriz de rotação \mathbf{R} . Essa mesma relação pode ser representado pela matriz de homografia \mathbf{H} , expressa na equação 9.8 e estimada no procedimento de calibração.

$$s \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \mathbf{H} \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{pmatrix}$$
(9.8)

 $\mathbf{m}_{c}^{\prime} = \mathbf{H}\mathbf{M}_{O}$

 $\operatorname{com} \mathbf{H} = \lambda \mathbf{A} \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{T} \end{pmatrix} = \begin{pmatrix} \mathbf{h}_1^T & h_{14} \\ \mathbf{h}_2^T & h_{24} \\ \mathbf{h}_3^T & h_{34} \end{pmatrix}, \text{ no qual } \lambda \text{ é uma escalar e } \mathbf{A} \text{ é uma matriz}$

 3×3 com os parâmetros intrínsecos. Essa mesma relação pode ser expressa como:

$$u_i' = h_{11}X_{Oi} + h_{12}Y_{Oi} + h_{13}Z_{Oi} + h_{14}$$
(9.9)

$$v_i' = h_{21}X_{Oi} + h_{22}Y_{Oi} + h_{23}Z_{Oi} + h_{24}$$
(9.10)

$$1 = h_{31}X_{Oi} + h_{32}Y_{Oi} + h_{33}Z_{Oi} + h_{34}$$

$$u'_{i} = u'_{i}(h_{31}X_{Oi} + h_{32}Y_{Oi} + h_{33}Z_{Oi} + h_{34})$$

$$v'_{i} = v'_{i}(h_{31}X_{Oi} + h_{32}Y_{Oi} + h_{33}Z_{Oi} + h_{34})$$

(9.11)

No qual i denota a coordenada do i-ésimo objeto. Reorganizando estas relações em forma matricial, tem-se:

$$\mathbf{L} = \begin{pmatrix} X_{O1} & Y_{O1} & Z_{O1} & 1 & 0 & 0 & 0 & 0 & u'_{1}X_{O1} & u'_{1}Y_{O1} & u'_{1}Z_{O1} & u'_{1} \\ 0 & 0 & 0 & 0 & X_{O1} & Y_{O1} & Z_{O1} & 1 & v'_{1}X_{O1} & v'_{1}Y_{O1} & v'_{1}Z_{O1} & v'_{1} \\ \vdots & & \vdots & & & \vdots & & & \vdots \\ X_{Oi} & Y_{Oi} & Z_{Oi} & 1 & 0 & 0 & 0 & 0 & u'_{i}X_{Oi} & u'_{i}Y_{Oi} & u'_{i}Z_{Oi} & u'_{i} \\ 0 & 0 & 0 & 0 & X_{Oi} & Y_{Oi} & Z_{Oi} & 1 & v'_{i}X_{Oi} & v'_{i}Y_{Oi} & v'_{i}Z_{Oi} & v'_{i} \\ \vdots & & & \vdots & & & \vdots \\ X_{ON} & X_{ON} & X_{ON} & 1 & 0 & 0 & 0 & 0 & u'_{N}X_{ON} & u'_{N}Y_{ON} & u'_{N}Z_{ON} & u'_{N} \\ 0 & 0 & 0 & 0 & X_{ON} & Y_{ON} & Z_{ON} & 1 & v'_{N}X_{ON} & v'_{N}Y_{ON} & v'_{N}Z_{O1} & v'_{N} \end{pmatrix}$$

$$(9.12)$$

com $\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{14}, h_{21}, h_{22}, h_{23}, h_{24}, h_{31}, h_{32}, h_{33}, h_{34})^T$ e $\mathbf{L}\mathbf{h} = 0$.

A matriz **L** tem a dimensão $2N \times 12$ e depende do número de pontos de referência(N) no sistema de coordenadas 3-D e 2-D, e **h** é um vetor 12×1 , definido a menos de uma escalar. Para evitar a situação **h** = 0, o qual não tem significado físico, deve ser imposto condições sobre **h**. As duas condições propostas por Faugeras em [Faugeras 1993] e utilizado no toolbox de calibração do Matlab ¹ são:

$$\|\mathbf{h}_3\|^2 = 1 \tag{9.13}$$

е

$$(\mathbf{h}_1 \wedge \mathbf{h}_2) \cdot (\mathbf{h}_1 \wedge \mathbf{h}_3) = 0 \tag{9.14}$$

¹http://www.vision.caltech.edu/bouguetj/calib doc/

Se a condição 9.13 for satisfeita, nota-se que pode haver diversas soluções para os parâmetros intrínsecos e extrínsecos. A condição para que haja uma solução é que o determinante das matrizes \mathbf{r}_1 , \mathbf{r}_2 e \mathbf{r}_3 seja igual a um, de modo que a matriz \mathbf{R} seja ortogonal.

Primeiramente os valores dos pontos observados na imagem (u'_i, v'_i) , relacionados ao plano da retina centrado em c, representado pela matriz $\hat{\mathbf{m}}'_c$, e seu correspondente em coordenadas globais (X_{Oi}, Y_{Oi}, Z_{Oi}) , representado pela matriz $\hat{\mathbf{M}}'_O$ e relacionado sistema de coordenadas do objeto centrado em O, são substituídos na matriz \mathbf{L} . Então a matriz \mathbf{h} pode ser calculado]a como auto vetor da matriz $\mathbf{L}^T \mathbf{L}$ associado ao seu menor auto valor. Em outras palavras, a solução para a estimação de \mathbf{h} é última coluna da matriz \mathbf{V} em SVD, $\mathbf{L} = \mathbf{U} \mathbf{D} \mathbf{V}^T$ [Hartley e Zisserman].

Uma vez que a matriz \mathbf{h} é calculada, os parâmetros intrínsecos e extrínsecos são estimados como [Heikkila. e Silvén 1997]:

$$u_0 = \mathbf{h}_1^T \cdot \mathbf{h}_3 \tag{9.15}$$

$$v_0 = \mathbf{h}_2^T \cdot \mathbf{h}_3 \tag{9.16}$$

$$-fD_u s_u = \sqrt{\mathbf{h}_1^T \mathbf{h}_1 - u_0} = \|\mathbf{h}_1 \wedge \mathbf{h}_3\|$$
(9.17)

$$-fD_v = \sqrt{\mathbf{h}_2^T \mathbf{h}_2 - v_0} = \|\mathbf{h}_2 \wedge \mathbf{h}_3\|$$
(9.18)

$$\mathbf{A} = \begin{pmatrix} -fD_u s_u & 0 & u_0 \\ 0 & -fD_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$
(9.19)

Tem-se que $\lambda = \frac{1}{\|\mathbf{A}^{-1}(h_{11},h_{21},h_{31})^T\|}$. Assim, pode-se estimar os valores dos parâmetros extrínsecos \mathbf{r}_1 , \mathbf{r}_2 , $\mathbf{r}_3 \in \mathbf{T}$ como:

$$T_z = h_{34}$$
 (9.20)

$$T_x = \frac{(h_{14} - u_0 T_z)}{-f D_u s_u} \tag{9.21}$$

$$T_y = \frac{(h_{24} - v_0 T_z)}{-f D_v} \tag{9.22}$$

$$\mathbf{r}_1 = \lambda \mathbf{A}^{-1} \left(h_{11}, h_{21}, h_{31} \right)^T \tag{9.23}$$

$$\mathbf{r}_3 = \mathbf{r}_1 \wedge \mathbf{r}_2 = \lambda \mathbf{A}^{-1} \left(h_{14}, h_{24}, h_{34} \right)^T \tag{9.24}$$

$$\mathbf{r}_2 = \lambda \mathbf{A}^{-1} \left(h_{21}, h_{22}, h_{32} \right)^T \tag{9.25}$$

9.5 Modelando a distorção



Figura 9.4: Efeito da distorção radial e tangencial, respectivamente. Linhas sólidas: sem distorção, linhas tracejadas: distorção positiva, linhas pontilhadas: distorção negativa (radial).

No modelo dos parâmetros intrínseco da câmera apresentado por Heikkila e Silvén em [Heikkila. e Silvén 1997], é utilizado o modelo *pinhole* como base e é entendido com correções nas coordenadas de pontos na imagem que sofrem distorção. A correção mais usada é para a distorção radial que causa o deslocamento radial dos pontos no plano imagem $(\delta u^{(r)}, \delta v^{(r)})^{\mathbf{T}}$, e pode ser aproximado pela equação 9.26.

$$\begin{pmatrix} \delta u^{(r)} \\ \delta v^{(r)} \end{pmatrix} = \begin{pmatrix} u \left(k_1 r^2 + k_2 r^4 + \cdots \right) \\ v \left(k_1 r^2 + k_2 r^4 + \cdots \right) \end{pmatrix}$$
(9.26)

Na equação 9.26, $k_1 \in k_2$ são os coeficientes da distorção radial e $r = \sqrt{u^2 + v^2}$. Sabe-se que a função de distorção é dominada por sua componente radial e modelos mais elaborados e de ordem superior, além de ineficazes, podem ser numericamente instável. Logo, o deslocamento radial detectado na calibração da câmera é dado pela equação 9.27.

$$\begin{pmatrix} \delta u^{(r)} \\ \delta v^{(r)} \end{pmatrix} = \begin{pmatrix} u \left(k_1 r^2 + k_2 r^4 + k_3 r^6 \right) \\ v \left(k_1 r^2 + k_2 r^4 + k_3 r^6 \right) \end{pmatrix}$$
(9.27)

A expressão que modela a componente tangencial da distorção em um ponto, distorção causada principalmente por imperfeições no centro da lente, é geralmente escrita como mostrado na equação 9.28

$$\begin{pmatrix} \delta u^{(t)} \\ \delta v^{(t)} \end{pmatrix} = \begin{pmatrix} 2p_1 uv + p_2 \left(r^2 + 2u^2\right) \\ p_1 \left(r^2 + 2v^2\right) + 2p_2 uv \end{pmatrix}$$
(9.28)

Na equação 9.28, p_1 e p_2 são os coeficientes da distorção tangencial e $\delta u^{(t)}$ e $\delta u^{(t)}$ são as componentes tangenciais da distorção da coordenada (u, v) no plano da retina.

O modelo completo da câmera, considerando o modelo *pinhole* com correção para as componentes da distorção radial e tangencial é mostrado na equação 9.29.

$$\begin{pmatrix} u'\\v' \end{pmatrix} = \begin{pmatrix} -fD_u s_u \left(u + \delta u^{(r)} + \delta u^{(t)}\right)\\ -fD_v \left(v + \delta v^{(r)} + \delta v^{(t)}\right) \end{pmatrix} + \begin{pmatrix} u_0\\v_0 \end{pmatrix}$$
(9.29)

Neste modelo, o conjunto de parâmetros intrínsecos $(f, s_u, u_0, v_0, D_u, D_v)$ é aumentado com os coeficientes de distorção $(k_1, k_2, k_3, p_1, p_2)$.

9.6 Estimação do erro associado aos parâmetros calculados

Problemas de visão computacional começam normalmente com um ou um conjunto de imagens que serão processadas em diversas etapas. O desenvolvimento do melhor algoritmo exige o conhecimento de como incertezas vindas de perturbação randômicas que afetam a qualidade da imagem adquirida são propagadas nas diversas etapas do algoritmo, resultando em uma perturbação na variável calculada [Bebis, Harris e Erol 2002].

No processo de calibração de câmeras, a matriz de homografia \mathbf{H} é estimada baseado na coordenada na imagem (u', v') correspondente ao ponto $\mathbf{M}(X_0, Y_0, Z_0)$ em *pixel*, como mostrado na Figura 9.3 e modelado na equação 9.8.

As matrizes $\overline{\mathbf{m}'_c}$ e \mathbf{M}'_0 são definidas como sendo quantidades medidas não observáveis, ou ou seja, sem erro associado a tais medidas. Associadas a essas matrizes não observáveis ideais estão os vetores randômicos de ruído $\overline{\Delta \mathbf{m}'_c}$ e $\overline{\Delta \mathbf{M}'_0}$, devido à distorção, por exemplo. As medidas observáveis, ou estimadas, são representadas por $\widehat{\mathbf{m}'_c}$ e $\widehat{\mathbf{M}'_0}$. As equações 9.30 e 9.31 mostram a relação entre essas quantidades.

$$\widehat{\mathbf{m}}_c' = \overline{\mathbf{m}}_c' + \Delta \mathbf{m}_c' \tag{9.30}$$

$$\widehat{\mathbf{M}'_O} = \overline{\mathbf{M}'_O} + \Delta \mathbf{M}'_O \tag{9.31}$$

A matriz de homografia **H** representa a matriz ideal não observável dos parâmetros da câmera. Adicionada a essa matriz ideal, tem-se Δ **H**, uma matriz 3 × 3 que representa a pertubação randômica em **H** introduzida pelas pertubações randômicas $\overline{\Delta \mathbf{m}'_c} \in \overline{\Delta \mathbf{M}'_0}$ em $\widehat{\mathbf{m}'_c} \in \widehat{\mathbf{M}'_O}$, respectivamente.

$$\widehat{\mathbf{H}} = \mathbf{H} + \Delta \mathbf{H} \tag{9.32}$$

Em procedimentos de calibração, o erro é considerado com sendo uma distribuição Gaussiana tanto nos parâmetros estimados como nos pontos medidos, com o valor médio igual a zero e com as matrizes de covariância $\sum_{H}, \sum_{m_c, M_O}$, respectivamente. Com isso a equação 9.32 pode ser reescrita como mostrado na equação 9.33.

$$\widehat{\mathbf{H}} = \mathbf{H} + \Delta \mathbf{H} = \mathbf{H} + 3\sigma_{\mathbf{H}} \tag{9.33}$$

capítulo 10

Localização 3-D a partir de 4 pontos usando marcador passivo em superfícies deformáveis

10.1 Introdução

Ao conhecer as relações geométricas entre diversas características do espaço 3-D, pode-se estimar a posição e a orientação da câmera em relação ao sistema de coordenadas do objeto a partir de uma única imagem. Dependendo do número de pontos característicos, as técnicas de estimação são subdivididas em problema com 3 pontos, com 4 pontos e com n pontos. Encontram-se soluções analíticas somente para os dois primeiros casos. Para o caso de n pontos, o conjunto de equações das projeções são sobre determinados, e, conseqüentemente, métodos estatísticos devem ser aplicados. Soluções analíticas são normalmente rápidas de serem calculadas, mas devido a erros nas medidas das coordenadas dos pontos na imagem elas não proporcionam resultados precisos.

Métodos de estimação de posição baseados em uma única vista possuem técnicas distintas dos métodos de estimação de posição baseada em múltiplas vistas, uma vez que as técnicas utilizadas nesta última manipulam as correspondências entre vistas diferentes. Informações extras sobre textura, e contorno, aumentam a precisão da estimação 3-D em ambas abordagens.

Localização 3-2D a partir de 4 pontos usando marcador passivo em superfícies deformáveis

No sistema proposto de estimação de posição 3-D, padrões planares colocados em uma superfície deformável são identificados em vídeos tempo-real, são rastreados e as coordenadas dos cantos em *pixel* são usadas para estimar a posição 3-D do padrão planar relativo ao sistema de referência da câmera. O número de pontos característicos usados para a estimação da posição e da orientação da câmera em relação ao sistema de coordenadas do objeto, além das características do objeto a ser observado, como sua forma, cor, textura, serão também estudados.

10.2 Marcadores passivos

Em aplicações tempo-real, no qual o tempo de processamento de imagens é limitado, a qualidade das imagens adquiridas e o tipo de marcador usado como referência para o rastreio são de extrema importância, uma vez que é melhor rastrear poucos marcadores, mas confiáveis, do que diversos marcadores, mas com o resultado do rastreio não confiável.

Marcadores são objetos proeminentes, que agem como pontos de referência. Há três tipos principais de marcadores: os marcadores passivos, os marcadores ativos, e os marcadores naturais.

Para a aplicação proposta, que é estimar o movimento cardíaco a partir de imagens de vídeo, optou-se por marcadores passivos na superfície viscosa do coração, ao invés de utilizar marcadores ativos, tais como feixes de laser. Os principais motivos dessa decisão foram:

- A viscosidade da região deformável a ser considerada (superfície do coração) e as condições de iluminação provida pelo endoscópio causam a reflexão da luz do endoscópio, observada pelas imagens adquiridas. Além disso, feixes de lasers são difundidos, gerando pontos de luz largos na superfície com contornos nebulosos. Estes fatores dificultam o rastreio de marcadores ativos e naturais, mas não afetam o rastreio de marcadores passivos, devido ao grande contraste entre este último e a superfície considerada;
- Se forem considerados pequenos marcadores passivos, pode-se supor que a região padrão no qual se encontra o marcador permanece plano, mesmo se este objeto padrão for colocado em superfícies deformáveis, como é o caso. Além disso, a posição relativa 3-D entre os pontos rastreados não muda, para pequenos marcadores, e não há oclusão desses marcadores pois o deslocamento destes não é grande quando colocados na superfície do coração.

O marcador passivo escolhido para a aplicação foi uma folha quadriculada, com retângulos brancos e pretos alternados, como um tabuleiro de xadrez, como mostrado na Figura 10.1. Optouse por um padrão preto e branco ao invés de padrões coloridos, pois estes primeiros oferecem maior contraste com a região de interesse, fator importante para detecção de região e extração de informações. A maior vantagem de utilizar cantos para o rastreio é que cantos são invariantes à escala, ou seja, os cantos de interesse podem ser rastreados a distâncias distintas da câmera.

78



Figura 10.1: Exemplos dos marcadores passivos planares usados.

10.3 Localização 3-D

O problema de estimação do posicionamento e orientação 3-D do objeto padrão em relação ao sistema de coordenadas da câmera é resolvido usando a matriz de homografia **H**, expressa pela equação 9.7. Na fase de estimação, é suposto que a câmera já foi calibrada e os seus parâmetros intrínsecos são usados para calcular os vetores de translação e rotação que descrevem a configuração do objeto padrão em relação à câmera. A primeira estimativa é feita pelas relações lineares expressas nas equações 9.20 a 9.25.

O passo final da estimação dos parâmetros extrínsecos é a otimização dos parâmetros já estimados. Isso é feito pela minimização erro da projeção, em *pixel*, ou seja, para cada conjunto de estimativas é verificado os pontos na imagem a partir dos parâmetros intrínsecos e extrínsecos, e então estes pontos são comparados com os seus valores reais na imagem e o algoritmo de otimização tenta minimizar este erro. Este é um problema de minimização não linear, sendo este solucionado pelo método do Gradiente descendente, no qual a cada interação os parâmetros extrínsecos são atualizados.

Ao final da estimativa, é feita novamente a projeção dos pontos na imagem, a partir dos parâmetros extrínsecos, e o erro entre estes pontos e os reais são calculados e armazenados, para validar a estimativa.

10.4 Rastreando marcadores ao longo do tempo

O problema de rastreamento é: de dado um certo marcador (ou objeto padrão) de interesse na i-ésima imagem, deseja-se saber a localização deste mesmo marcador na (i+1)-ésima imagem. Como há várias possibilidades na escolha do marcador correspondente na (i+1)-ésima, o problema de correspondência é dito ambíguo, e por isso levanta uma série de questões tais como quais marcadores, objetos padrão ou condições são usadas para reduzir tal ambigüidade.

Dois tipos de rastreamento podem ser distintamente abordados: rastreamento baseado em características na imagem ou rastreamento de regiões completas e objetos padrão.

Rastreamento baseado em características requer a extração das características na imagem, permitindo a rápida correspondência entre duas imagens devido ao pequeno número de informações. Rastreamento baseado em região economiza a extração de características na imagem, mas a correspondência entre duas imagens requer grande custo computacional.

Na abordagem feita neste manuscrito, a região a ser rastreada é um objeto padrão quadrado de pequena dimensão (fig.10.1), e as informações necessárias para estimar o deslocamento deste padrão são as coordenadas dos quatro cantos. Logo, o primeiro modelo (rastreamento baseado em características) de rastreamento é usado para encontrar os cantos do quadrado correspondente em duas imagens consecutivas e o segundo modelo (rastreamento baseado em região) pode ser usado para identificar o padrão quadrado dentro da imagem.

10.4.1 Método de rastreamento: Correlação

As técnicas de correlação baseadas na intensidade dos *pixels* na imagem são o meio básico por quais pontos em imagens diferentes são relacionados. Funções de correlação medem a similaridade entre pontos vizinhos. Há diversas funções de correlação, e será apresentado o princípio básico da técnica da correlação e a função usada na aplicação proposta.



Figura 10.2: Princípio da técnica de correlação.

O princípio das técnicas de correlação é mostrado na figura 10.2. A fim de se achar as coordenadas do *pixel* na imagem 2 que corresponda ao *pixel* com coordenada (u_0, v_0) na imagem 1, considera-se primeiramente uma janela retangular (vizinhança ou janela de pesquisa) de tamanho $2P + 1 \times 2N + 1$ centrada em (u_0, v_0) , no qual $P \in N$ definem o tamanho da janela e são parâmetros de projeto. Então, é calculado a correlação $C_{12}(\tau)$ com a intensidade da segunda imagem percorrendo a linha $v_2 = v_0$, como mostrado na equação 10.1.

$$C_{12}(\tau) = \frac{1}{K} \sum_{u_1 = -N}^{+N} \sum_{v_1 = -P}^{+P} \left(\left(I_1(u_1 + u_0, v_1 + v_0) - \overline{I_1(u_0, v_0)} \right) \cdot \left(I_2(u_1 + u_0 + \tau, v_1 + v_0) - \overline{I_1(u_0 + \tau, v_0)} \right) \right)$$
(10.1)

80

Na equação 10.1 $K = (2N + 1) (2P + 1) \sigma_1 (u_0, v_0) \sigma_2 (u_0 + \tau, v_0)$ e $\overline{I_1 (u_0, v_0)}$ e $\sigma_1 (u_0, v_0)$ são a intensidade média e o desvio padrão da imagem 1 no ponto (u_0, v_0) , dados pelas equações 10.2 e 10.3, respectivamente.

$$\overline{I_1(u_0, v_0)} = \frac{1}{(2N+1)(2P+1)} \sum_{u_1=-N}^{+N} \sum_{v_1=-P}^{+P} \left(I_1(u_1+u_0, v_1+v_0)\right)$$
(10.2)

$$\sigma_1^2(u_0, v_0) = \frac{1}{(2N+1)(2P+1)} \sum_{u_1=-N}^{+N} \sum_{v_1=-P}^{+P} \left(I_1(u_1+u_0, v_1+v_0) - \overline{I_1(u_0, v_0)} \right)^2$$
(10.3)

As relações $\overline{I_2(u_0 + \tau, v_0)} e \sigma_2(u_0 + \tau, v_0)$ possuem fórmulas similares as equações e 10.2 e 10.3, respectivamente. Nota-se que, por causa da normalização por $\sigma_1 e \sigma_2$, C_{12} sempre fica no intervalo entre -1 e 1. A curva de $C_{12}(\tau)$ tem geralmente um máximo que é alcançado no valor τ_0 de τ .

capítulo 11

Experimentos em calibração de câmeras e localização 3-D

11.1 Introdução

A Seção experimental deste projeto foi dividida em três partes. Na primeira é feita a validação do procedimento de calibração e os primeiros resultados são apresentados. Em seguida, é feita a validação do algoritmo de localização 3-D desenvolvido e, finalmente, é feita a estimativa do movimento cardíaco, baseadas em imagens reais do batimento cardíaco de um porco. Todos os experimentos foram feitos com imagens adquiridas pela câmera CMOS DALSA TM CAD6->1M75, que tem uma taxa de aquisição de até 1000 quadros por segundos a uma resolução de 512 × 512 pixels. Tanto a taxa de aquisição quanto a resolução são programáveis e foram utilizadas lentes focais de 4,8 mm e 25 mm nos testes.

Nos experimentos de calibração, estimação de movimento e rastreamento de padrões, foi usado um padrão de calibração, como mostrado na figura 10.1. Esse padrão é composto por uma superfície plana composta por 20×12 quadrados em malha, com cada quadrado variando de 5 mm a 30 mm de lado, dependendo da aplicação.

Após imprimir o padrão de calibração, é feita a medida do tamanho dos lados do quadrado, uma vez que pode haver distorção induzido pela impressora, e esse valor que é considerado tanto nos algoritmos de calibração quanto nos de estimação de deslocamento. A precisão nas estimativas da calibração é limitada pelo erro associado à medida do lado do quadrado. No caso de um padrão de calibração com dimensão de 100 mm e com a dimensão de cada lado do quadrado de 5 mm e erro máximo associado a esta medida de 0.5 mm (utilizando uma régua milimetrada). Supondo a distribuição normal do erro com o desvio padrão de σ_P e erro máximo de $3\sigma_P$, tem-se então uma precisão máxima alcançável proporcional a 1/50000 pela estimativa realizada na calibração.

Tanto a calibração quanto as técnicas de estimação de movimento necessitam que, no procedimento de aquisição de imagens, a câmera observe o padrão de calibração de diferentes pontos de vista. Para isso, pode-se mover tanto o padrão de calibração quanto a própria câmera. Foi escolhido manter a câmera fixa em um tripé observando o alvo se movendo. O movimento relativo do padrão de calibração foi medido para que os resultados da estimação pudessem ser comparados com seus valores reais.

11.2 Avaliação do modelo da câmera e do procedimento de calibração

11.2.1 Primeiros resultados em calibração

A estimação dos parâmetros que descrevem o modelo da câmera é feito primeiramente por meio do cálculo da matriz da homografia \mathbf{H} (equações de 9.15 a 9.19) e em seguida as estimativas são otimizadas por meio do método de regressão não linear gradiente descendente, para minimizar o erro entre as estimativas feitas e os dados observados. Para que se possa realizar a calibração, deve-se obter um conjunto de imagens do padrão de calibração em diferentes configurações dentro do campo de trabalho. Este é o procedimento mais importante do processo de calibração. Pode-se mover tanto a câmera quanto o quadro alvo, mas deve-se assegurar que a luminosidade sobre o quadro se mantenha homogênea durante todo o processo de aquisição de imagem. O número mínimo de imagens necessárias para realizar a calibração são duas imagens em configurações distinta. Porém, é altamente recomendável fazer no mínimo 15 imagens do padrão de calibração, uma vez que a acurácia na estimativa dos parâmetros da câmera são diretamente proporcional ao número de imagens usadas. A Figura 11.1 mostra um conjunto de 15 imagens com um padrão de calibração com quadrados de 5 mm de lado dispostos em diversas configurações.

Se o plano do padrão de calibração realiza somente translação entre duas imagens, então $\mathbf{R}_1 = \mathbf{R}_2$, logo a segunda imagem não adiciona nenhuma informação na calibração. Como exposto por Zhang em [Zhang 1998], quando o ângulo entre duas imagens é θ , e $\theta \leq 5^o$, os planos são quase paralelos, e a estimativa dos parâmetros da câmera deve falhar. Zhang então propõe, para melhor desempenho, mudar significantemente o ângulo de orientação do plano alvo entre duas aquisições de imagens. Melhor desempenho parece ser alcançada quando o ângulo de rotação entre duas imagens quaisquer seja cerca de 30^o , mas ao aumentar o ângulo entre o plado da imagem e o plano



Figura 11.1: Conjunto de imagens usadas para calibração.

Tabela 11.1: Parâmetros intrínsecos estimado a partir do conjunto de imagens da Figura 11.1. **Estimativa** $(pixel) \mid 3\sigma(pixel) \mid Desvio padrão relativo (%)$

	Estimativa (<i>pizei</i>)	$\mathbf{O}(purce)$	Desvio padrao relativo (70)
Distância Focal 1	2480,48454	7,97156	0,1071
Distância Focal 2	2470,53980	7,75492	0,1046
\mathbf{x}_0	$374,\!83951$	5,56831	$0,\!4952$
\mathbf{y}_0	$288,\!90475$	$5,\!60200$	0,6463
Coeficiente de distorção \mathbf{k}_1	-0,58746	0,01648	0,9351

do padrão de calibração faz com que a detecção dos cantos seja menos precisa.

Uma vez que o sistema tenha extraído as extremidades de todos os quadrados do padrão de calibração a partir de um conjunto de imagens, é realizada a calibração, resultando as estimativas dos parâmetros intrínsecos, assim como o erro associado a essas estimativas, como mostrado na Tabela 11.1.

A precisão nas estimativas mostradas na Tabela 11.1 pode ser analisada por meio do erro da projeção na imagem dos pontos, a partir dos parâmetros estimados. Esse erro é uma medida da qualidade da calibração com respeito ao conjunto de imagens dadas, assim como é usado para descartar imagens com muito ruído. O erro da projeção obtida com o conjunto de imagens da Figura 11.1 e os parâmetros estimados (tabela 11.1) é mostrado na figura 11.2. A Figura 11.3 mostra um gráfico com a configuração da câmera em relação às 15 imagens usadas para realizar a calibração.



Figura 11.2: (a) Erro da projeção - cada cor é associada a uma das imagens consideras na calibração; (b) detalhe do ponto real(cruz) e ponto estimado (círculo).

11.2.2 Influência do modelo de distorção radial e tangencial

O principal objetivo deste experimento é verificar a influência do modelo de distorção radial e tangencial na imagem. Essa influência pode ser observada pelo deslocamento do pixel na imagem induzido pela distorção. Para que a distorção possa ser facilmente observada, um conjunto de imagens altamente distorcidas foi usado para realizar a calibração, uma vez que o modelo de distorção tem um papel importante no modelo da câmera. No procedimento de aquisição de imagens, foi utilizado uma lente focal de $4.8 \ mm$ observando o quadro de tabuleiro de xadrez com quadrados de $5 \ mm$ de lado. A escolha de lentes focais curtas foi pelo fato que nessa configuração o ângulo de visão é maior, e as imagens observadas com esse tipo de lente sofrem grande distorção. Foram tiradas 14 imagens em diferentes configurações do padrão de calibração, como mostrado na figura 11.4.

A figura 11.5a mostra o impacto do modelo completo de distorção (radia e tangencial) sobre cada *pixel* da imagem e o modelo da câmera estimado no processo de calibração. Cada flecha representa do deslocamento efetivo de um *pixel* induzido pela distorção das lentes. Nota-se que os pontos próximos aos cantos da imagem são deslocados por mais de 60 *pixels*, o que é visível nas imagens da Figura 11.4. A Figura 11.5b mostra o impacto da componente da distorção tangencial. Neste gráfico, o deslocamento máximo provocado pela distorção tangencial foi de 0.4 *pixels* (na parte superior a esquerda da imagem). Finalmente, a Figura 11.5c mostra o impacto da componente radial da distorção. Este gráfico é bastante similar ao gráfico do impacto de toda a distorção. Em todas as três figuras, a cruz indica o centro da imagem, e o círculo incida o ponto principal estimado na imagem.


Figura 11.3: Configuração do padrão de calibração em relação ao plano da imagem, expressa em milímetro: (a) câmera fixa, (b) quadro fixo.

11.2.3 Análise do desvio padrão e da estabilidade nos parâmetros estimados

O objetivo deste experimento foi o de verificar a influência do número de imagens usadas no procedimento de calibração sobre as estimativas dos parâmetros do modelo da câmera, assim como influência do número de imagens sobre o desvio padrão associado a cada estimativa e a estabilidade numérica das estimativas.

Neste experimento, foi utilizada a lente focal de 25 mm e o padrão de calibração foi composto por quadrados de 28 mm de lado. A fim de investigar a estabilidade das estimativas do modelo da câmera, a calibração foi realizada com um número variável de imagens, sendo que o conjunto de imagens foram tomadas em combinação do conjunto de 20 imagens que foram adquiridas. Os resultados são mostrados em forma gráficas na Figura 11.6 e na Figura 11.7.

É claro que as incertezas associadas às estimativas diminuíram com o aumento no número de imagens no processo de calibração. Vale também ressaltar a inconsistência do parâmetro estimado k_2 quando varia-se o número de imagens na calibração, uma vez que que o valor estimado oscila bastante com a adição de novas imagens. Isso demonstra claramente instabilidade numérica, e por este motivo, nos experimentos que se seguem é considerado apenas o primeiro parâmetro da distorção radial.

Foi também observado que a componente tangencial do modelo de distorção não tem uma influência significativa no modelo da câmera, uma vez que a estimação dos parâmetros da distorção tangencial foram próximos de zero, mesmo para imagens altamente distorcidas (figs. 11.6d, 11.7d e 11.5b). O modelo de distorção poderia ser então resumido somente pelo primeiro termo da distorção radial, o parâmetro k_1 .

Quanto ao desvio padrão relativo referente a cada estimativa, estes são mostrados na Figura



Figura 11.4: Conjunto de imagens bastante distorcidas usadas para calibração.

11.7. À medida que se aumenta o número de imagens, a estimativa se torna mais precisa e logo o desvio padrão relativo tende a decrescer. Mas a queda no valor do desvio padrão tende a um certo limite, e a partir daí, mesmo se aumentar o número de imagens, não há alteração significativa nos valores do desvio padrão nem na estimativa feita. Notou-se neste teste, que ao ser considerado mais que 15 imagens no procedimento de calibração, tantos os parâmetros estimados quanto seus erros associados não mudavam significativamente. Por isso, optou-se em sempre utilizar 15 imagens no procedimento da calibração.

11.2.4 Espaço de trabalho

Neste ensaio foi usado o robô manipulador PA10, com 7 graus de liberdade, mostrado na Figura 11.8 para fixar o padrão de calibração em sua garra, a fim que se pudesse controlar precisamente o deslocamento do padrão de calibração. Uma câmera estática foi montada em um tripé observando o quadro alvo se movendo e foi montado de tal forma que o plano da imagem ficasse inicialmente paralelo ao plano do padrão de calibração. A câmera CMOS DALSA foi usada configurada para capturar 500 quadros por segundo à uma resolução de 512×512 pixels. Foi escolhida a lente focal de 25 mm e o padrão de calibração com quadrados de 5 mm de lado. Foram usadas no procedimento de calibração entre 15 e 27 imagens. Com este sistema montado foram feitas as aquisições das imagens. Em seguida foi realizado a calibração para diferentes configurações do padrão de calibração ao plano da imagem, e os resultados são mostrados nas Tabelas 11.2 e 11.3. A distância média entre o plano da imagem e o plano do padrão de calibração foi de cerca de 900 mm para a primeira configuração e 600 mm para a segunda configuração, medidas



Figura 11.5: Efeito da distorção sobre cada *pixel* da imagem: (a)efeito do modelo da distorção na imagem, (b)efeito da distorção tangencial e (c) efeito da distorção radial.

	Estimativa (pixel)	$3\sigma(pixel)$	Desvio padrão relativo (%)
Distância Focal 1	2818,78320	210,91217	2,4941
Distância Focal 2	$2785,\!07762$	207,33710	2,4815
\mathbf{x}_0	$307,\!28682$	4,79308	0,5199
\mathbf{y}_0	314,32430	4,84972	0,5143
Coeficiente de distorção \mathbf{k}_1	8,53363	2,50187	9,7726

Tabela 11.2: Parâmetros intrínsecos estimado com o espaço de trabalho de 3,5cm.

a partir de configurações articulares do manipulador. O espaço de trabalho em que o padrão se deslocou na primeira configuração foi cerca de 35 mm (realizando somente rotação) e na segunda foi de 40 mm. Pelos resultados das Tabelas 11.2 e 11.3, é evidente que o parâmetro de distorção estimado foi maior que o esperado, pois normalmente este parâmetro assume valores menores que 1. Isso se deve principalmente a grande distância entre os dois planos (900 mm), fazendo com que a malha do padrão de calibração não seja visível à esta distância.

Após configurar o sistema, foi gerada uma trajetória no qual, a cada ponto de passagem, o robô manipulador parava para que pudesse fazer a aquisição da imagem e a configuração articular

Estimativa (pixel)	$3\sigma(pixel)$	Desvio padrão relativo (%)
2534,38525	105,44028	1,3868
2481,63384	$100,\!67320$	1,3522
138,07819	$125,\!17637$	30,2187
315,01687	37,75029	$3,\!9945$
-0,60329	$0,\!17245$	9,5283
	Estimativa (<i>pixel</i>) 2534,38525 2481,63384 138,07819 315,01687 -0,60329	Estimativa (pixel) $3\sigma(pixel)$ 2534,38525105,440282481,63384100,67320138,07819125,17637315,0168737,75029-0,603290,17245

Tabela 11.3: Parâmetros intrínsecos estimado com o espaço de trabalho de 4 cm.



Figura 11.6: Influência do número de imagens usados na calibração sobre os valores estimados dos parâmetros e a estabilidade de tais valores: Parâmetros estimados.

do robô, para posterior comparação entre os dados.

Utilizando parâmetros intrínsecos estimados na calibração, pode-se estimar os parâmetros extrínsecos, logo a posição e orientação do padrão de calibração em relação ao plano da imagem. Podendo em seguida comparar o deslocamento relativo estimado com base nos parâmetros intrínsecos e em diferentes imagens e o deslocamento real realizado. Os resultados são mostrados na Figura 11.9.

Na Figura 11.9a, o padrão de calibração realizou somente uma translação senoidal de amplitude 2,5 cm e período de 5 cm nos eixos $x \in z$, definindo um espaço de trabalho de 5 cm quadrados. Como no processo de calibração o espaço de trabalho foi limitado a 3,5 cm, a estimação do deslocamento relativo do padrão mostrado na Figura 11.9a foi compatível com o deslocamento real (obtida pela configuração das juntas do manipulador) somente próximo da origem, onde as imagens da calibração foram tiradas, e o erro cresce a medida que o padrão se distancia do espaço



Figura 11.7: Influência do número de imagens usados na calibração sobre os valores estimados dos parâmetros e a estabilidade de tais valores: desvio padrão relativo associado as estimativas.

de trabalho definido na calibração, como mostrado na Figura 11.9c. Outra fonte de erro que gerou uma má estimação do deslocamento foi o parâmetro da distorção e o pequeno tamanho dos quadrados no padrão para a dada configuração.

Na Figura 11.9d, por outro lado, tanto o procedimento de calibração quanto o deslocamento posterior do padrão se deram dentro do mesmo espaço de trabalho, que foi de 4 cm em todos os eixos, e as condições externas, tais como iluminação e configuração da câmera, se mantiveram constante durante todo o experimento. Conseqüentemente, o erro entre o deslocamento relativo real e o estimado a partir dos parâmetros intrínsecos e das imagens foi praticamente constante para todos os pontos e a estimativa, além das estimativas terem sido mais precisas que aquelas feitas na primeira configuração.



Figura 11.8: Robô manipulador PA10.

Tabela 11.4: Parâmetros intrínsecos estimado a partir do conjunto de imagens da Figura 11.11. **Estimativa** $(pixel) | 3\sigma(pixel) |$ Desvio padrão relativo (%)

		$O \circ (P \cup O \cup V)$	\mathbf{D} os no paarao rolativo (\mathbf{r})
Distância Focal 1	2361,87542	14,06551	0,1985
Distância Focal 2	2366,27331	$13,\!98857$	0,1971
\mathbf{x}_0	$313,\!51746$	$11,\!29570$	1,2010
\mathbf{y}_0	267,07987	9,18785	$1,\!1467$
Coeficiente de distorção \mathbf{k}_1	-0,54086	0,02650	1,6332

11.2.5 Avaliação da calibração extrínseca

Nos experimentos sobre a calibração extrínseca, foi desenvolvido um experimento para validar a estimativa do modelo da câmera e a robustez da técnica de estimação do movimento. Foi utilizada neste experimento a câmera DALSA com a lente focal de 25 mm montada em um tripé e o robô manipulador Mitsubishi PA10 com o padrão de calibração fixado em seu "pulso". Iluminação externa foi utilizada para garantir luminosidade uniforme e controlada durante todo o procedimento de aquisição das imagens. As configurações da câmera, como abertura e ajustes focais, foram coletados no início de cada experimento e a câmera foi montada de modo a garantir o paralelismo entre o plano da imagem e o plano do padrão de calibração na inicialização.

A primeira parte do experimento é fazer a aquisição das imagens para obter o modelo da câmera por meio da calibração. Foram obtidas 15 imagens do padrão (malha de quadrados com 5 mm de lado) em diferentes configurações, sendo que as posições eram controladas e salvas para posterior análise. As imagens obtidas são mostradas na Figura 11.11 e o resultado da calibração é mostrado na Tabela 11.4. O espaço de trabalho foi definido de modo que ocupasse toda a área usada para o deslocamento do padrão na etapa de localização do padrão.

A etapa seguinte foi programar o robô para que este se deslocasse dentro do espaço de trabalho definido na fase de calibração. A magnitude entre dois pontos da trajetória de 1,65 mm na direção do eixo z, que é a mesma amplitude esperada do batimento cardíaco entre duas imagens.



Figura 11.9: Efeito do espaço de trabalho de 3,5 cm na estimação do deslocamento. (a) Deslocamento relativo estimado e real realizado em duas configurações. (b) o espaço de trabalho definido na calibração e o definido no movimento de translação. (c) erro entre o deslocamento relativo real e o estimado.

Não houve deslocamento do padrão nos eixos x e y. A cada ponto de controle, era feita a aquisição da imagem e armazenada as configurações das juntas do robô. Depois de feita a aquisição de todas as imagens, os parâmetros extrínsecos para cada imagem foram estimados e o deslocamento relativo estimado foi comparado com o deslocamento real realizado pelo manipulador, uma vez que não foi considerado erro associado às medidas das configurações das juntas, por serem estas bastante precisas. Os resultados obtidos são mostrados na Figura 11.12.

Como mostrado na Figura 11.12b, o erro entre o deslocamento relativo real e o estimado é bem menor que o deslocamento realizado entre duas imagens, que foi de 1,65 mm. A precisão na estimação da posição é conseqüência de um modelo de câmera preciso. Com base nos resultados mostrados na Figura 11.12, pode-se validar o modelo de câmera aqui considerado, assim como o procedimento adotado para realizar a aquisição das imagens. Os resultados foram úteis também



Figura 11.10: Efeito do espaço de trabalho de 4 *cm* na estimação do deslocamento. (a) Deslocamento relativo estimado e real realizado em duas configurações. (b) o espaço de trabalho definido na calibração e o definido no movimento de translação. (c) erro entre o deslocamento relativo real e o estimado.

para ter uma estimativa do erro de posicionamento em milímetros, uma vez que os algoritmos usados apresentam os erros de estimativa em unidades de *pixel*.

11.3 Fontes do erro nas medidas

Em processamento de imagens, a aquisição de imagens e a qualidade destas imagens são ponto chave na avaliação dos resultados. Erros sistemáticos interferem na estimativa dos parâmetros do modelo da câmera e por isso devem ser levados em conta. Aqui serão mostrados alguns resultados de experimentos que mostram como cada fator interfere na estimativa dos parâmetros intrínsecos da câmera.



Figura 11.11: Conjunto de imagens usadas para calibração.



Figura 11.12: Resultados da calibração extrínseca: (a) deslocamento real e estimado realizado pelo padrão de calibração; (b) erro entre o deslocamento relativo real e o estimado.

11.3.1 Padrão de calibração

Existem diversas tipos de padrões de calibração. Eles podem ser coplanar, tridimensional, sólido, transparente, entre outros. O tamanho do padrão depende das dimensões do espaço no qual serão feitas as medidas e das grandezas a serem observadas. No caso ideal, o padrão de calibração deveria cobrir todo o espaço de interesse. Isso pode ser feito colocando um conjunto de pontos de referência no ambiente de estudo e as coordenadas desses pontos são medidas com régua e usadas na calibração como pontos de controle. Porém, em diversos casos, é impossível usar pontos de referência externa com a precisão suficiente. Por questões práticas, o padrão não pode ser muito grande. Logo, a distância entre o padrão e a câmera é geralmente menor que a distância de interesse.

Para garantir uma melhor qualidade nas estimativas feitas durante a calibração, deve-se fazer uma medida precisa do padrão de calibração antes de iniciar o processo de aquisição de imagem. Os erros nas coordenadas nos pontos de controle podem causar resultados insatisfatórios, uma vez que a precisão nas estimativas realizadas na calibração não pode ser maior que a precisão das medidas dos pontos de controle. Por exemplo, se for necessário uma precisão na estimativa próxima a 1/1000, o desvio padrão nos pontos de controle não pode ser maior que 0,05 mm em todo o volume do padrão de calibração de dimensão de 500 mm. Esse requisito é dificilmente atingido sem aparelhos de alta precisão. Outro fator que deve ser observado em padrões planos é a homogeneidade em sua superfície. Exemplos de diversos tipos de padrão são mostrados na Figura 11.13.



Figura 11.13: Exemplos de padrões de calibração.

11.3.2 Iluminação

Em sistemas que empregam lentes, como é o caso de aquisição de imagens, a iluminação no plano da imagem decresce a partir do eixo óptico com a quarta potência do co-seno do ângulo de obliqüidade com o eixo óptico [Heikkila 1997]. Essa queda de iluminação, ou "fall-off", pode causar distorção no perfil dos padrões observados e conseqüentemente é observado um erro na localização de padrões quando são usados algoritmos de extração de padrões.

A conseqüência da mudança da iluminação nas imagens usadas na calibração é uma pequena variação na estimação do parâmetro da distância focal. A distorção causada pela queda de iluminação resulta em um erro nos cantos observados na imagem pelo algoritmo de extração de padrões. Esse erro é menor perto do centro da imagem e cresce radialmente até chegar nas bordas da imagem, e é linear em ambos eixos vertical e horizontal, como mostrado na Figura 11.14. Essa variação é então compensado por uma pequena variação na estimação da distância focal, e esse tipo de distorção não pode ser detectado pela análise do erro associado a estimativa.

A magnitude da variação no valor estimado da distância focal devido a queda de iluminação na imagem depende do tamanho do da lente focal. Normalmente, lentes focais pequenas, como a usada em alguns experimentos de 4,8 mm, proporciona maior disparidade que lentes com focal maior (f = 25 mm).



Figura 11.14: Efeito da iluminação: Vista do padrão de calibração e do erro correspondente observado nos eixos $x \in y$: (a) superfície do padrão e da câmera paralelas, (b) o erro é centrado em zero e não tem efeito "fall-off"; (c) Superfície do padrão e da câmera não paralelas, (d) o erro é menos próximo ao centro da imagem e cresce radialmente quando se aproxima das bordas.

A intensidade luminosa no ambiente é a causa também do deslocamento horizontal na localização dos pontos de interesse, observados nas imagens. Este deslocamento é causado por uma mudança na intensidade luminosa geral no ambiente e é compensado por uma pequena mudança no valor estimado do ponto principal. Porém, esse mesmo fenômeno ocorre em superfícies de objetos com iluminação não uniforme, causando uma variação na magnitude do deslocamento de pontos de controle diferente para cada imagem. Essa variação não é completamente compensada na estimação do modelo da câmera, e, logo, pode ser detectada a partir do aumento do erro residual estimado no processo de calibração. Para evitar esses tipos de erros, iluminação uniforme deve ser providenciada a fim de garantir uma calibração correta.

Na prática, iluminação uniforme é muito difícil de ser obtida quando um objeto plano 2-D é observado por uma câmera que se move durante a aquisição das imagens, uma vez que esse

	Estimativa $(pixel)$	Estimativa $(pixel)$
	$\mathbf{Z} = \mathbf{50cm}$	$\mathbf{Z} = \mathbf{54cm}$
Distância Focal 1	2366,67634	2361,87542
Distância Focal 2	$2374,\!98791$	$2366,\!27331$
\mathbf{x}_0	190,76639	$313,\!51746$
\mathbf{y}_0	$268,\!69403$	267,07987

Tabela 11.5: Resultados da calibração: efeito do ajuste do foco na estimativa de parâmetros.

movimento causa uma variação na quantidade de luz "percebida"em diferentes ângulos. Neste caso, o erro não é necessariamente observável no resíduo, uma vez que ele é compensado na estimação dos parâmetros estimados do modelo da câmera.

11.3.3 Foco e Íris

O modelo da câmera usado para realizar a calibração é baseado na projeção *pinhole* e não é considerado neste modelo o efeito dos ajustes no foco e na íris da câmera. Logo, deve-se determinar como esses ajustes afetam os resultados da calibração.

No caso do foco da câmera, a distância entre o plano da imagem e o plano principal é ajustado de modo a obter imagens nítidas. É evidente que a operação de ajuste focal afeta a estimação da distância focal, mas pode também mudar outros parâmetros. Na Tabela 11.5 é mostrado um exemplo de como o ajuste do foco afeta a estimação dos parâmetros. Isso foi feito colocando o plano do padrão e o plano da imagem à distâncias de 50 cm e 54 cm (fig. 11.15) em dois processos de calibração e regulando o foco no início de cada etapa. Em ambos processos de calibração, as imagens do padrão foram obtidas na mesma configuração exata, usando para tal o robô manipulador *PA10*. Os resultados mostraram que a distância focal e o ponto principal estimados sofreram grande influência do ajuste focal, mas as componentes de distorção radial e tangencial não sofreram mudanças em suas estimativas.

No caso da abertura e fechamento da íris, Li em [Li 1994] notificou que mudanças na configuração da íris não alteram significantemente a estimativa da distância focal. Também notou que o deslocamento na localização do ponto principal não varia tanto quanto o verificado com o ajuste do foco. Todavia, os resultados não podem ser generalizados porque eles dependem muito na configuração óptica das lentes. Logo, a calibração deve ser realizada com as configurações da câmera nas mesmas condições usadas na fase de medidas.

11.3.4 Discussão sobre o procedimento de calibração

A calibração de câmeras é necessária quando se quer estimar o deslocamento 3-D com uma alta precisão. Todavia, na literatura sobre calibração de câmeras não é dada muita atenção aos erros das



Figura 11.15: Configuração do plano do padrão e da imagem no processo de calibração.

estimativas originados do procedimento de calibração. Uma grande quantidade de experimentos foi realizada a fim de identificar as diversas fontes de erro que afeta a acurácia das estimativas.

O modelo de câmera aqui usado possibilita a compensação somente dos erros originados da distorção radial e tangencial. Não foram levados em consideração a influência da intensidade de luz nas imagens, nem o ajuste focal e da íris, nem o espaço de trabalho, entre outros. As fontes de erro devem ser identificadas e ações devem ser tomadas a fim de evitar que influenciem no resultado da calibração e que este erro seja propagado para a estimativa. Como medida de precaução pode-se equipar o ambiente experimental com iluminação externa homogênea, padrões de calibração com medidas confiáveis, além de elaborar o procedimento de aquisição de imagens que minimize esses erros e que englobe todo o espaço de trabalho desejado na fase de medição.

O efeito de cada fonte de erro na estimação dos parâmetros foi verificado a partir de diversos experimentos de aquisição de imagens e calibração da câmera. Foi observado que quando a câmera era calibrada nas mesmas condições e configurações utilizadas na fase de medidas, parte das fontes de erros era compensada na estimativa do modelo da câmera, e caso o erro residual permanecesse pequeno, podia-se usar os parâmetros estimados para fazer a localização 3-D bastante precisa.

Foi também verificado que um modelo complexo da distorção não proporciona uma estimação 3-D mais precisa. Então, na fase de estimação 3-D, foi simplificado o modelo de distorção, considerando apenas o primeiro coeficiente da distorção radial e sem considerar o modelo de distorção tangencial.

11.4 Localização 3-D

O objetivo dos experimentos seguintes é avaliar e validar o *software* desenvolvido em linguagem de programação C para rastreamento e estimativa da localização 3-D a partir de quatro pontos usando um marco passivo em uma superfície deformável, em tempo real.

	Toolbox de calibração do Matlab Sistema em OpenCV		
Vetor de Translação	$\left(\begin{array}{ccc} -56,299 & 4,754 & 476,244 \end{array}\right)$	$\left(\begin{array}{cccc} -56,299 & 4,754 & 476,244 \end{array}\right)$	
Vetor de Rotação de Rodrigues	$\left(\begin{array}{ccc} 3,141 & -0,001 & -0,048 \end{array}\right)$	(3, 129 -0, 001 -0, 049)	
Erro em pixel	$(0,0013 \ 0,0360)$	$(0,0021 \ 0,0417)$	

Tabela 11.6: Estimação dos parâmetros extrínsecos.

11.4.1 Validação do método

O objetivo deste experimento foi validar o método usado para estimar a localização 3-D, baseado na matriz de homografia, como mostrado na Seção 10. Isso foi feito comparando os resultados do sistema com os resultados estimados com *toolbox* de calibração do Matlab 1 .

O toolbox de calibração de câmeras do Matlab foi usado como o sistema de referência, uma vez que este utiliza o mesmo método que foi desenvolvido por nós para estimar os parâmetros extrínsecos. Foram usados os mesmos conjuntos de imagens e parâmetros do modelo da câmera no experimento.

A comparação é feita usando as mesmas coordenadas dos pontos de controle em ambos sistemas para estimar os parâmetros extrínsecos da câmera com respeito ao sistema de coordena fixado na superfície do padrão de calibração, correspondendo os vetores de translação e rotação. Os resultados obtidos são mostrados na Tabela 11.6, obtidos pela estimação dos parâmetros extrínsecos da imagem mostrada na Figura 11.16.



Figura 11.16: Imagem usada para estimação dos parâmetros extrínsecos.

Como esperado, os resultados numéricos obtidos para o vetor de translação dos dois sistemas

¹http://www.vision.caltech.edu/bouguetj/calib doc/

são idênticos, como mostrado na Tabela 11.6, uma vez que foi usado o mesmo método. A única diferença observada foi na estimação do erro em pixel. Essa diferença é devida aos diferentes algoritmos usados para calcular a reprojeção da malha de pontos de controle na imagem a partir dos parâmetros extrínsecos.

A segunda parte do experimento avalia o sistema de rastreamento. Os mesmos pontos iniciais de controle foram usados em ambos sistemas. No *toolbox* de calibração do Matlab, para cada imagem, foi escolhido manualmente as quanto cantos dos pontos de controle e no sistema implementado o sistema de rastreamento encontrava os pontos de controle correspondentes em cada imagem. Os resultados são mostrados na Figura 11.17.



Figura 11.17: (a) Estimação de movimento usado dois sistemas distintos e o deslocamento relativo real; (b) erro entre o deslocamento relativo real e o estimado.

Observando o gráfico do erro mostrado na Figura 11.17, nota-se que o melhor desempenho foi obtida pelo sistema implementado em linguagem C usando a biblioteca de imagem OpenCV. Essa performance só foi possível devido ao sistema de rastreamento eficiente e pelo fato de se escolher os pontos de controle manualmente é uma fonte de erro que tem que ser considerado.

11.4.2 Análise do erro usando o robô manipulador *PA10* e o sistema de localização *Polaris*

Este experimento investistiga o erro da estimativa de posicionamento obtido pelo sistema proposto, implementado em linguagem C. Para isso, foi usado o robô manipulador *PA10* para realizar os deslocamentos do padrão de calibração de forma precisa. A estimação de posicionamento é feita pelo sistema proposto e pelo sistema de localização *NDI Polaris*, apresentado na Seção 11.4.2.1. A configuração do experimento é mostrado na Figura 11.18.



Figura 11.18: Configuração experimental: câmera de vídeo e o sistema de localização NDI Polaris.

11.4.2.1 Sistema de localização NDI Polaris

O sistema *Polaris* é um instrumento de medição capaz de rastrear movimentos tridimensionais em tempo real com três graus de rotação. Ele rastreia ferramentas ativas que emitem luz infravermelho por meio de *leds* e ferramentas passivas com esferas refletoras. Esse rastreio é realizado pela medição óptica da posição dos marcadores nas ferramentas. Quando a configuração geométrica dos marcadores na ferramenta é rastreada e reconhecida através de um banco de dados, ambas posição e orientação do ponto de origem definida na ferramenta são obtidas. O erro associado à uma medida é a raiz quadrada do erro médio quadrado. O erro mínimo obtido pelo sistema *Polaris* para o erro volumétrico é $0,350 \ mm$ e o erro máximo é de $2 \ mm$. Por este sistema apresentar erros pequenos em suas medidas, optou-se por utilizar essas medidas como referência absoluta nos experimentos que se seguem.



Figura 11.19: Sistema de localização NDI Polaris.

O sistema de coordenas de referência global é também chamado de sistema de coordenadas padrão. Ele é obtido na fabricação do aparelho e a origem é centrada na face da frente do sensor

de Posição (*"Position Sensor"* na figura 11.19²). O sistema *Polaris* pode medir posições em uma grande área no espaço, conhecido como campo de visão, que é mostrado na Figura 11.20. A



Figura 11.20: Campo de visão do sistema de localização NDI Polaris.

estimação de posição do sistema *Polaris* é realizada da seguinte maneira: o sensor de posição (*Position Sensor*) mede a posição da luz infravermelho que pode ser tanto refletido ou emitido dos marcadores da ferramenta (figura 11.21). Ele coleta quadros na cadência de 60 Hz para marcadores refletores ou emissores de luz infravermelho. A recepção do sinal dos marcadores refletores de luz infravermelho pode ser feito na freqüência de 20, 30 ou 60 Hz. Na configuração do sistema deve-se escolher a freqüência da aquisição de dados, além de escolher o modelo da ferramenta no qual será feito a medida. A translação e rotação de cada ferramenta são então calculadas quando requisitado, e o resultado é mostrado na tela e salvo em disco.

11.4.2.2 Configuração e resultados experimentais usando o sistema de localização Polaris

Padrões de calibração quadrados de diferentes tamanhos foram fixados na superfície da ferramenta com marcadores ativos do sistema *Polaris* e a ferramenta foi então fixada na ponta de prova do robô manipulador *PA10*. A configuração experimental é mostrada na figura 11.22a.

Ambos câmera DALSA e o sensor de posição Polaris foram colocados a uma distância de 150 cm do plano da ferramenta. Com este sistema foi realizado a aquisição de imagens com o padrão de calibração de 5 mm de lado para se poder estimar os parâmetros intrínsecos da câmera. Foram obtidas 15 imagens e os resultados da calibração são mostrados na tabela 11.7.

O movimento da ferramenta foi controlado precisamente, por meio do controle das juntas do

 $^{^{2}} http://www.ndigital.com/polaris.php$



Figura 11.21: Medição da posição:(a) Marcadores ativos emitem luz infravermelho que é recebido pelo sensor de posição para estimar a posição da ferramenta . (b) Marcadores passivos refletem luz infravermelho para o sensor de posição, e este estima assim a posição da ferramenta.

Tabela 11.7: Parâmetros intrínsecos estimado	para o experimento	o usando o sistema <i>Polaris</i> .	
--	--------------------	-------------------------------------	--

	Estimativa (pixel)	$3\sigma(pixel)$	Desvio padrão relativo (%)
Distância Focal 1	2461,94714	13,39370	0,1813
Distância Focal 2	$2366,\!27331$	$13,\!20296$	$0,\!1788$
\mathbf{x}_0	242,06410	8,03882	$0,\!1088$
\mathbf{y}_0	$308,\!81025$	$11,\!32742$	$0,\!1534$
Coeficiente de distorção \mathbf{k}_1	-0,61521	0,03705	0,0005

robô PA10, para que realizasse deslocamentos relativos de 1,65 mm na direção do eixo z entre duas aquisição de imagens. Para cada configuração do robô manipulador, foi feita a aquisição da imagem e estimada a posição da ferramenta usando o algoritmo proposto e o sistema de localização *Polaris.* O movimento relativo da ferramenta é mostrado na figura 11.22b e o erro associado à esta estimativa são mostrados nas figuras 11.22c e 11.22d.

A figura 11.22c mostra o erro absoluto na estimação do deslocamento do padrão de calibração que consiste na soma do erro estimado pelo sistema de localização *Polaris* com a diferença entre o deslocamento relativo estimado pelo sistema proposto e aquele encontrado pelo sistema *Polaris* (figura 11.22b). A reconstrução do deslocamento do alvo planar é mostrado na figura 11.22e. Notase, a partir do gráfico mostrado na figura 11.22c que o erro de estimação no eixo z é maior que o erro encontrado nos eixos x e y, e alcançou o valor máximo de aproximadamente 4 mm. Isso pode parecer pequeno, quando comparado com a distância entre o plano da imagem e o plano do padrão de calibração (figura 11.22e), que foi cerca de 150 cm. Porém, se for considerar o erro relativo, o erro máximo encontrado foi de 244% para o erro de 4 mm, o que não é aceitável para o nosso objetivo, que é estimar deslocamentos milimétricos utilizando imagens de uma câmera calibrada.

Avaliando a configuração experimental e o procedimento de aquisição de imagens, não foi

encontrado nenhuma fonte de erro que pudesse justificar os erros de estimativa encontrados, uma vez que o modelo da câmera foi obtida com sucesso e a configuração experimental foi mantida intocável durante o experimento. Porém, foi observado que a luz infravermelho emitida pela ferramenta do sistema *Polaris* mudava a luminosidade da superfície do padrão de calibração. Com isso, as coordenadas dos pontos de controle utilizada para a estimação de posicionamento mudava com a variação da luminosidade. Isso explica a razão porque o maior erro encontrado foi na direção do eixo z, uma vez que a estimação de profundidade é extremamente sensível à pequena variação nas coordenadas dos pontos de controle.

11.4.3 Estimação do movimento cardíaco

Há dois tipo de movimentos principais distinto dentro do corpo humano: o movimento devido a respiração e o devido ao batimento cardíaco. Porém, são os movimentos devido à respiração que causam o maior distúrbio, porque ele provoca longos ciclos de deslocamento em diversos órgãos do abdômen e do peito. Distúrbios devido ao batimento cardíaco são restritos a uma área pequena em volta do coração.

O objetivo do próximo experimento é caracterizar o movimento cardíaco a partir de imagens da superfície do coração. Neste trabalho foi reconstruída a distância, em milímetros entre o plano da imagem e os padrões de calibração fixados na superfície do coração, em condições cirúrgicas reais.

A configuração experimental do experimento *in vivo* é descrito na figura 11.23. Foi feito um acesso ao coração de um porco vivo anestesiado com técnicas cirúrgicas no Laboratório Cirúrgico experimental de Montpellier. Foi usado a câmera CMOS DALSA com a lente focal nominal de 25 mm, configuração da íris em 4 e a aquisição de imagens a uma taxa de 273 quadros por segundos à uma resolução de 512 × 512 *pixels*. Foi usada iluminação externa para garantir iluminação homogênea durante a aquisição de imagens. Nem a configuração do foco nem a distância entre o coração e o plano da imagem foram obtidos. Na superfície do coração do porco foram colocados padrões de calibração com aproximadamente 3 mm de lado, como mostrado na Figura 11.23b.

A fim de fazer a estimação do movimento cardíaco usando uma câmera, é necessário conhecer o modelo da câmera (parâmetros intrínsecos) nas condições experimentais reais. Todavia, não foi realizada a calibração da câmera durante o experimento, e informações como a distância entre o plano da imagem e o coração, configuração do foco não foram medidos.

Uma alternativa para se estimar os parâmetros intrínsecos da câmera nas mesmas condições experimentais é tentar reproduzir as mesmas configurações físicas observadas no teste (iluminação, ajustes da câmera) e então fazer aquisição de imagens para a calibração. Para que isto pudesse ser feito, a distância entre a câmera e a superfície do coração teve que ser estimado. A partir das imagens adquiridas nos experimentos e as relações geométricas conhecidas, obteve-se a distância média entre a câmera e a superfície do coração que foi de 40 cm. Com esta configuração, e com

Tabela 11.8: Parâmetros intrínsecos estimados e usados para estimação do movimento cardíaco (Distância de 48 *cm* entre a câmera e o padrão de calibração).

	Estimativa $(pixel)$	$3\sigma(pixel)$	Desvio padrão relativo (%)
Distância Focal 1	2361,87542	14,06551	0,1985
Distância Focal 2	2366,27331	$13,\!98857$	0,1971
\mathbf{x}_0	$313,\!51746$	$11,\!29570$	1,2010
\mathbf{y}_0	267,07987	9,18785	$1,\!1467$
Coeficiente de distorção \mathbf{k}_1	-0,54086	0,02650	1,6332

o padrão de calibração de 5 mm de lado, foi realizada a calibração, variando a distância entre a câmera e o plano do padrão de calibração em torno de 40 cm. Os resultados são mostrados na Figura 11.24.

Para cada modelo de câmera estimado, e usando as imagens experimentais do coração, foi estimada a posição do padrão de calibração fixo na superfície do coração relativo ao sistema de coordenadas da câmera. Com isso, obteve-se uma segunda estimativa da distância entre o plano da imagem e o padrão de calibração que foi cerca de 48 cm. A amplitude do batimento cardíaco foi de cerca de 3 cm no eixo $z \in 1 cm$ nos eixos $x \in y$. Usando esse resultado, foi feita uma nova calibração da câmera e os valores dos parâmetros estimados são mostrados na Tabela 11.8, e estes foram usados para estimar o movimento cardíaco.

A etapa final do experimento consistiu em estimar o movimento cardíaco, usando a técnica de estimação de posicionamento mostrado no capítulo 10. Foram usadas 1249 imagens e o algoritmo de rastreio foi utilizado para encontrar os quatro pontos de controle em cada nova imagem. O deslocamento da superfície do coração em relação ao sistema de coordenadas da câmera é mostrado na figura 11.25.

Como pode ser visto na figura 11.25b, o movimento cardíaco é bem mais complexo que os movimentos causados pela respiração, uma vez que o primeiro é uma combinação das duas componentes e a influência de cada uma delas pode ser facilmente observada no gráfico com o deslocamento na direção do eixo x (primeiro gráfico da figura 11.25b). O terceiro gráfico da figura 11.25b mostra a distância estimada entre a superfície do coração e o sistema de coordenadas da câmera. Neste experimento a câmera está estática, logo, as informações de profundidade obtida se refere somente ao movimento do coração. A amplitude total do movimento é de cerca de 8 mm na direção x, 6 mm na direção y e 20 mm na direção z.

Se considerarmos o erro da reprojeção da imagem, mostrado na Figura 11.25c, como um parâmetro sobre a qualidade na estimativa, pode-se concluir que os valores estimados são bastante próximos dos valores reais, uma vez que o erro estimado do movimento cardíaco foi muito similar ao obtido no experimento 11.4.2. No teste 11.4.2 os erros da reprojeção, em *pixel*, correspondia a um erro máximo de 4 mm na estimativa, limite que pode ser aplicado a este experimento.

Na Figura 11.26b tem-se a análise de freqüência do movimento cardíaco que auxilia na identificação da freqüência dominante e na discussão sobre os requisitos da velocidade de processamento das imagens. Pode-se decompor o sinal em duas componentes principais: a de menor freqüência, que é causada pela respiração, de 0, 43 Hz, e a de maior freqüência, devido ao batimento cardíaco, com valor de 1,75 Hz. A amplitude do sinal é causada principalmente pelo movimento cardíaco, uma vez que a respiração foi forçada por equipamentos mecânicos e foi configurada de maneira a minimizar a sua influência no movimento cardíaco. Nota-se que a onda de alta freqüência não é cíclica e envolve movimentos aleatórios e caóticos. Para que se possa detectar todas as transições rápidas no movimento do coração e para evitar perda de dados por *aliasing*, o período de amostragem das imagens devem ser maiores que 100 Hz.

Os resultados mostrados foram bastante satisfatório, uma vez que foi possível limitar o erro na estimativa do movimento cardíaco e assim, pode-se caracterizar o seu comportamento. Resultados similares são apresentados em [Ginhoux et al. 2003], no qual a estimação do movimento cardíaco é apresentado em *pixel* e a amplitude da respiração foi induzida de modo a facilitar a distinção do movimento devido a repiração e ao batimento cardíaco.



Figura 11.22: (a) Configuração experimental: padrão de calibração com 15 mm de lado.(b) Deslocamento relativo estimado. Estimação do erro em (c) milímetros e (d) *pixel*. Estimação 3-D do posicionamento e intervalo de erro.



Figura 11.23: Experimento *in vivo*: (a) Configuração do sistema, (b) padrão de calibração colocado na superfície do coração.



Figura 11.24: Resultados da calibração: (a) Distância focal, (b) ponto principal e (c) coeficiente de distorção radial.



Figura 11.25: Estimação do movimento da superfície do coração de um porco: (a) padrão de calibração usado; (b) Movimento cardíaco estimado; (c) erro sobre a estimativa, em *pixel*.



Figura 11.26: (a) Estimação do movimento cardíaco no domínio do tempo. (b) Espectro e amplitude para baixas freqüências.

capítulo 12

Conclusões da Parte II

Na segunda parte do manuscrito foi abordado o problema de calibração de câmeras de vídeo e estimação 3-D a partir marcos passivos observados por uma câmera calibrada. Neste estudo, o objetivo foi determinar quais técnicas possibilitam extrair informações 3-D em imagens de forma precisa, utilizando para isso uma única câmera.

O primeiro problema abordado foi o de calibração de câmeras. A calibração de câmeras, baseadas no modelo *pinhole*, foi apresentada, assim como os parâmetros que descrevem o modelo da câmera, ou parâmetros intrínsecos, e os que descrevem a localização da câmera no espaço, ou parâmetros extrínsecos. Sem uma boa estimação do modelo da câmera, a localização 3-D pode se tornar inadequada. Grande atenção foi dada no procedimento de calibração e na fase de aquisição de imagem. Diversos experimentos foram realizados com diversos procedimentos sugeridos na literatura especializada em visão computacional a fim de definir o melhor procedimento. Por fim, foi definido um procedimento de aquisição de imagens para a realização da calibração e as configurações necessárias para minimizar a influência externa na estimação dos parâmetros intrínsecos e, conseqüêntemente extrínsecos.

A partir dos resultados obtidos dos experimentos de calibração, foram identificados diversas fontes de erros que afetam a acurácia da estimativa do modelo da câmera. Foi constatado que alguns erros são causados por um modelo de câmera insuficiente e outros são devido a fatores externos.

Outro problema abordado foi a de estimação da posição 3-D a partir de imagens de uma só câmera calibrada com erro de estimativa limitada. Foi apresentado o método baseado na matriz de homografia, no qual a posição e orientação da câmera são obtidas a partir dos parâmetros intrínsecos e pontos de controle conhecidos. Foi abordado também técnica de rastreio para localizar os pontos de controle de interesse em diferentes imagens.

Por fim, foram apresentados resultados de experimentos que permitiram uma ótima estimação de posição 3-D a partir de imagens de uma câmera calibrada. E a partir das técnicas e procedimentos propostos, foi implementado um sistema de localização e rastreio *on line*, possibilitando assim fazer a estimativa do movimento cardíaco de um porco a partir de imagens de uma câmera calibrada.

Bibliografia

- [Adôrno, Aguiar e Borges 2005]ADÔRNO, B. V.; AGUIAR, C. S. R.; BORGES, G. A. Planejamento de trajetória para o robô omni utilizando o algoritmo mapa de rotas probabilístico. *VII SBAI*, 2005.
- [Aguiar, Oliveira e Borges 2005]AGUIAR, C. S. R.; OLIVEIRA, F. M. G. S. A.; BORGES, G. A. Estimação de parâmetros geométricos de um robô móvel omnidirecional. *VII SBAI*, setembro 2005.
- [Aragones 2002]ARAGONES, J. Commande et Evaluation des perfomances d'un robot omnidirectionnel a roues. Tese (Doutorado) — Université Montpellier II, LIRMM, 161 rue ADA, 34392, Montpellier, Cedex 5, France. One of the recipients of the 2001/2002 Club EEA prize for the best french thesis in Automatic Control, 2002.
- [Aragones, Borges e Fournier 2002]ARAGONES, J.; BORGES, G. A.; FOURNIER, A. Accuracy improvement for a redundant vehicle. In: *33rd International Symposium on Robotics*. [S.l.: s.n.], 2002.
- [Aragones e Fournier 2002]ARAGONES, J.; FOURNIER, A. Commande dynamique d'un robot omni-directionnel redondant. In: *Conférence Internationale Francophone d'Automatique (CIFA)*. [S.l.: s.n.], 2002.
- [Bebis, Harris e Erol 2002]BEBIS, G.; HARRIS, F.; EROL, A. Development of a nationally competitive program in computer vision technologies for effective human-computer interaction in virtual. 2002.
- [Borges 2002]BORGES, G. A. Architectures matérielle et logicielle du robot Omni. 161, rue ADA 34392 Montpellier - France, 2002.
- [Borges 2002]BORGES, G. A. Cartographie de l'environnement et localisation robuste pour la navigation de robots mobiles. Tese (Doutorado) — Université Montpellier II, LIRMM, 161 rue ADA, 34392, Montpellier, Cedex 5, France. One of the recipients of the 2001/2002 Club EEA prize for the best french thesis in Automatic Control, May 2002.
- [Campion, Bastin e D'Andréa-Novel 1996]CAMPION, G.; BASTIN, G.; D'ANDRÉA-NOVEL, B. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics* and Automation, v. 12, n. 1, p. 47–62, February 1996.
- [Dantas 2002]DANTAS, M. Técnologias de Redes de Comunicação e Computadores. [S.l.]: Axcel Books do Brasil, 2002.

- [Faugeras 1993]FAUGERAS, O. Three-Dimensional Computer Vision A Geometric Viewpoint. [S.l.]: The MIT Press, 1993.
- [Ginhoux et al. 2003]GINHOUX, R. et al. Active filtering of physiological motion in robotized surgery using predictive control. *IEEE Transactions on robotics*, X, n. Y, November 2003.
- [Ginhoux et al. 2003]GINHOUX, R. et al. Model predictive control for tracking of repetitive organ motions during teleoperated laparoscopic interventions. *European Control Conference (ECC)*, 2003.
- [Gröger et al. 2002]GRÖGER, M. et al. Tracking local motion on the beating heart. SPIE Medical Imaging: Visualization, Image-Guided Procedures, and Display, v. 4681, 2002.
- [Guthart e Salisbury 2000]GUTHART, G.; SALISBURY, J. The intuitive telesurgery system: Overview and application. *IEEE Internation Conference on Robotics and Automation*, 2000.
- [Hartley e Zisserman]HARTLEY, R.; ZISSERMAN, A. Multiple View Geometry in Computer vision. [S.I.]: Cambridge Press.
- [Haykin 1998]HAYKIN, S. Neural Networks: A Comprehensive Foundation. Second. [S.I.]: Prentice-Hall, 1998.
- [Heikkila 1997]HEIKKILA, J. Accurate camera calibration and feature based 3-D reconstruction from monocular image sequences. Tese (Doutorado) University of Oulu, 1997.
- [Heikkila. e Silvén 1997]HEIKKILA., J.; SILVÉN, O. A four -step calibration procedure with implicit image correction. CVPR97, 1997.
- [Komoriya e Oyama 1992]KOMORIYA, K.; OYAMA, E. Position estimation of a mobile robot using optical fiber gyroscope. *IEEE*, 1992.
- [Krupa 2003]KRUPA, A. Commande par vision d'un robot de chirgie laparoscopique. Tese (Doutorado) Institut National Polytechnique de Lorraine, juillet 2003.
- [Krupa et al. 2003]KRUPA, A. et al. Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using servoing. *IEEE Transations on Robotics and Automation*, v. 19, n. 5, p. 842–853, october 2003.
- [le Corre 1998]le Corre, Y. Conception et Commande d'un Robot Omnidirectionnel. Tese (Doutorado) LIRMM-UM2, Decémbre 1998.
- [Li 1994] Camera calibration on a head-eye system for active vision.
- [Nakamura, Kishi e Kawakami 2001]NAKAMURA, Y.; KISHI, H.; KAWAKAMI, H. Heartbeat synchronization for robotic cardiac surgery. Proc. of the 2001 Int. Conf. on Robotics and Automation (ICRA), 2001.
- [Nakamura et al. 2000]NAKAMURA, Y. et al. Motion-cancelling robot system for minimally invasive cardiac surgery. *Journal of robotics society of Japan*, v. 18, 6 2000.
- [Ortmaier 2003]ORTMAIER, T. J. Motion Compensation in Minimally Invasive Robotic Surgery. Tese (Doutorado) — Technische Universität München, March 2003.
- [Schweikard et al. 2000]SCHWEIKARD, A. et al. Robotic motion compensation for respiratory movement during radiosurgery. *Journal of Computer Aided Surgery*, v. 5, n. 4, p. 263–277, 2000.
- [Zhang 1998]ZHANG, Z. A Flexible New Technique for Camera Calibration. [S.I.], December 1998.

Anexos

APÊNDICE A

Funções de acesso aos componentes da plataforma Omni

Todas as funções de acesso aos sensores, circuitos periféricos, foram implementados em linguagem C e estão presentes na biblioteca St.h. As funções são dividas em módulos, sendo que cada módulo desempenha uma tarefa específica. Há os módulos com funções de mais baixo nível, com a tarefa de enviar e receber dados dos diversos sensores, respeitando os protocolos de comunicação e o meio de comunicação. Há módulos de mais alto nível, que gerenciam os módulos de mais baixo nível, e sua principal tarefa é gerenciar os sensores, inicializando-os, lendo seus dados e padronizando-os. Como exemplo, tem-se os módulos de gerenciamento dos sensores girômetro laser Gyr.h, do joystick Man.h e ladar a laser Ldr.h. Por último, há os módulos de mais alto nível que independe do tipo de sensores e faz a manupulação de dados e gera sinais de controle para os módulos de níveis menores. Tem-se como exemplo o módulos de Assrv.h, que realiza a tarefa periódica de controle do robô, podendo este operar de diferentes maneiras, com a utilização de diferentes sensores em diferentes configurações. Aqui serão apresentados todos os módulos que compõem a biblioteca, assim como as suas funções.

A.1 Assrv.h: Módulo de controle, independente dos dispositivos de E/S

As funções presentes no módulo Assrv.h tem como finalidade gerar e enviar as referências de posição e velocidade dentro do espaço operacional do robô. Neste módulo, são gerados e manipulados as variáveis _OpDeltaPos [NO] e _OpPos [NO], que representam o deslocamento relativo entre o período de amostragem e absoluto de referência do robô, repectivamente. NO é o número de variáveis que representam a posição do robô, que no caso do robô Omni são três (X, Y, θ) . As funções definidas neste módulo são:

- AssrvInit(): Inicialização. Nesta função são incializada todas as variáveis em zero, o modo de operação é definido como modo parado (ModeArret:). São inicializados também os módulos utilizados, tais como RbtInit(), CorInit(), EstInit(), PntInit()e ManInit().
- AssrvGetPeriod(): retorna o período de amostragem ΔT_a no qual são feitas as leitura dos sensores e é realizado o controle. O valor padrão do período de amostragem ΔT_a é de 5 ms.
- AssrvGetMode(): retorna o tipo de modo de controle a plataforma móvel Omni está operando.
- AssrvSetMode(int mode): a variável *mode* define o modo dfe controle desejado. Os tipos que a variável *mode* podem assumir:
 - I. ModeArret: ou modo de parada. Neste modo, a cada período de amostragem ΔT_a , é aplicado nos motores a tensão necessária para que o robô pare (ou permaneça parado). Usado tipicamente na inicialização e na finalização de tarefas.
 - II. ModeOperateurSol: ou modo controle pelo operador com referência no solo. Os pontos de controle são enviados pelo joystick.
 - III. ModeOperateurVeh: ou modo controle pelo operador com referência no veículo. Os pontos de controle são enviados pelo joystick.
 - IV. ModeAutomatique: ou modo automático. Os pontos de controle são obtidos por um algoritmo de planificação de trajetória.
- AssrvUpdateOpPos(): procedimento de atualização da posição. Este é o procedimento central do módulo e é realizado por meio de uma tarefa RTX à um período ΔT_a . Três tarefas são realizadas neste procedimento. Primeiramente é feito a leitura dos sensores e estimado o deslocamento relativo realizado pelo robô entre um período de amostragem, atualizando a variável _OpDeltaPos[NO] . Em seguida é feita a estimativa do deslocamento global do robô e este valor é armazenado na variável _OpPos[NO]. Por fim, são enviados o sinal de controle para as rodas e, dependendo então do modo (mode) de controle, são gerados os pontos de controle a serem enviados no próximo período de amostragem.

A.2 Pnt.h: Módulo de Gestão do ponto

As funções presentes no módulo Pnt.h tem como finalidade atualizar as variáveis do posicionamento do robô, a partir das estimativas realizadas externamentes (outros módulos) e independe do robô e dos dispositivos de entrada/saída utilizados. As variáveis de posição atualizadas neste módulos, _OpDeltaPos[NO] e _OpPos[NO] , representam o deslocamento relativo entre o período de amostragem e absoluto do robô, repectivamente. NO é o número de variáveis que representa a posição do robô, que no caso do robô Omni são três (X, Y, θ) . As funções definidas neste módulo são:

- void PntInit(): Zera a variável _OpPos[NO] .
- void PntSetOpPos(unsigned long _localTime , const Reel _opDeltaPos[NO] , const Reel _opPos[NO]) : Subsitui os valores de _OpDeltaPos[NO]e _OpPos[NO] pelas novas estimativas armazenadas nas variáveis _opDeltaPose _opPos
- void PntUpdateOpPos(): Atualiza a estimativa das variáveis _OpDeltaPos e _OpPos[NO] (Reel *EstGetOpPos(&corbeille , _OpDeltaPos , estOpPos)) e integra a variável _OpDeltaPos para obter uma estimativa do deslocamento absoluto realizado pelo robô.
- Reel *PntGetOpPos(unsigned long *_localTime , Reel _opDeltaPos[NO] , Reel _opPos[NO]
): Obtém as últimas estimativas sobre o posiocionamento do robô. São armazenados nas variáveis _opDeltaPos[NO] e _opPos[NO] os valores obtidos na última estimativa, armazenadas nas variáveis globais _OpDeltaPose _OpPos

A.3 Est.h: Módulo de Estimação da posição

As funções presentes no módulo Est.c estima a posição e a velocidade do robô dentro do espaço operacional do mesmo, gerando as variáveis _OpDeltaPos[NO] e _OpPos[NO]. As funções definidas neste módulo são:

- EstInit: faz a inicialização do Girômetro laser (GyrInit) e inicializa as variáveis que armazenam o deslocamento relativo estimado durante um período, _OpDeltaPos[NO] , e a posição estimada no sistema de coordenadas global,_OpPos[NO], com zero.
- void EstSetOpPos(unsigned long _localTime, const Reel _opDeltaPos[NO], const Reel _opPos[NO]): Atualiza as variáveis _OpDeltaPos[NO] e _OpPos[NO] com os valores passados como dados de entrada nesta função.
- void EstClrOpPos(): inicializa a variável _OpPos[NO] com zero.

- void EstUpdateOpPos() : Faz a estimação do posicionamento operacional do robô. É a função principal de tal módulo. Esta função faz a leitura do girômetro GyrUpdateData(), dos dados articulares ArtGetData(&_LocalTime, artDeltaPosCount, artPosCount), e estima a posição do robô no sistema de coordenadas global (RbtChgDir)
- Reel *EstGetOpPos(unsigned long *_localTime, Reel _opDeltaPos[NO], Reel _opPos[NO]): Atualiza as variáveis globais _opDeltaPose _opPos.

A.4 Gyr.h: Módulo do Girômetro a laser

O módulo Gyr.h possui todos os procedimentos para a configuração e acesso ao girômetro a laser. Ele é independente das conecções físicas de entrada/saída do processo, uma vez que estes endereços são obtidos na inicialização do sistema. As funções que compõem este módulo são:

- GyrInit(): Inicialização do Girômetro. Neste procedimento é feita a configuração do girômetro para que este trabalhe no modo *Angle*, para que sua saída seja proporcional à velocidade angular em radianos por segundo, e no modo *Request*, para que o sensor só envie dados de sua leitura após um pedido pelo computador central. A configuração é feita por meio de comunicação serial RS-232 síncrona. Uma primeira leitura é feita e descartada, uma vez que esta não é válida.
- GyrUpdateData(): Atualiza a leitura do girômetro. Neste procedimento faz-se o necessário para se obter uma medida do girômetro, ou seja, é feito o requerimento de uma leitura ao girômetro, por meio de uma comunicação serial assíncrona, faz a leitura do dado lido e o instante de clock correspondente à tal leitura e armazena tal valor na variável global _LocalTime, e por fim, seta o flag indicando que os dados referentes ao girôemtro foram atualizados.
- GyrGetDeltaAngle(unsigned long *_localTime) : Obtém a variação do ângulo entre duas medidas consecutivas. Este procedimento retorna a variação angular (ω), em radiando, entre duas medidads consecutivas e salva o instante no qual foi realizada a última medida.

A.5 Man.h: Módulo do joystick

Este módulo faz a leitura do estado do joystick. As vaiáveis globais em tal módulo são _bouton [ManNbBoutons] e _position [ManNbAxes], que armazenam os estados dos botões e das coordenadas do controle, respectivamente. Os parâmetros ManNbBoutons (igual a três) e ManNbAxes(igual a três) são referentes ao joystick em particular que são o número de botões e o número de coordenadas de controle $(X, Y e \theta)$, respectivamente. Os procedimentos presentes em tal módulo são:
- ManInit(): Inicialização do Módulo de leitura do joystick. Faz a primeira de leitura do joystick (ManUpdateData());
- ManUpdateData(): Faz a leitura do estado do joystick referentes aos três botões, sendo seu estado habilitado ou não (digital), e à três coordenadas de referência, sendo o seu deslocamento nas direções X, Y e θ;
- ManGetBoutons(unsigned long *_localTime, int_bouton[ManNbBoutons]): Retorna os estados de controle dos botões lidos do joystick;
- ManGetPositions(unsigned long *_localTime, Reel _position[ManNbAxes]): Retorna as posições de controle do joystick.

A.6 Ldr.h: Módulo do Ladar a laser

O módulo Ldr.h possui todas as funções para acesso, configuração e leitura dos dados do Ladar a laser. As funções que compõem tal módulo são:

- LdrInit(): Inicialização do Ladar a laser. Esta função configura o ladar a laser para trabalhar com o plano máximo de dispersão de 270° e com a passo de angular de 0,6°. Em seguida faz uma primeira leitura do sensor;
- int LdrGetRxRdy(): Indica quando há dados disponíveis para serem lidos. Quando o ladar a laser faz a varredura de 270°, obtendo assim um mapa 2-D do ambiente, é enviado um sinal, via comunicação serial, que é lido por esta função;
- LdrStop(): Finalização do Ladar a laser;
- LdrUpdateData(): Faz a leitura dos dados provenientes do Ladar a laser. Espera até que uma varredura completa seja efetuada (int LdrGetRxRdy()) e então faz a leitura dos dados e seta a variável que indica que há dados novos disponíveis;
- int *LdrGetData(unsigned long *_localTime): Retorna os dados referente a última leitura do Ladar a laser.

A.7 McrRbt.h: Acesso à placa de Entrada/Saída

As funções que compõem o módulo McrRbt.h são funções de mais baixo nível que faz a comunicação entre o computador central e a placa de Entrada/saída fixada no barramento PCI (Seção 3.1). Estão presentes os protocolos de acesso a todas as entradas seriais presentes na plataforma Omni, além das funções básica de acesso e leitura dos codificadores ópticos. As funções que compõem o módulo McrRbt.h são:

- McrInit(): Retorna o endereço da FIFO da placa de *Entrada/Saída PCI*, em *RTX*. Pesquisa automaticamente a placa de *Entrada/Saída* nos barramentos PCI, obtém o endereço e autorização de acesso da FIFO;
- McrWatchDogInit(): Inicializa o cão de guarda da placad de Entrada/Saífa;
- McrWatchDogRestart():Reseta o cão de guarda da placa de Entrada/Saída;
- McrCmdInit(): Inicializa a tensão de referência dos motores. Obtém o endereço da placa *Entrada/Saída* e seta em zero a tensão de referência em todos os motores;
- McrCmd(const Reel _artCmd[NA]): Aplica a tensão de referência _artCmd[NA] nos motores. A variável _artCmd[NA] pode assumir os valores entre -10V a 10V, que será a tensão na saída dos conversores digiral/analógico da placa de Entrada/Saída.
- McrIncEncInit(): Inicilializa os codificadores ópticos incrementais. As variáveis que armazenam os dados provenientes dos codificadores ópticos são zerados, e o buffer na placa de *Entrada/Saída* que armazena a contagem antes de enviar ao procesador é também zerado.
- long McrIncEncGetGain() : Retorna o valor do ganho $K_{c_{ij}}$ dos codificadores ópticos incrementais, que no caso do Omni este valor é igual a quatro para todos os codificadores incrementais (Seção 3.2.2);
- void McrIncEncClearCount(const int _clear[NbIncEnc]): Zera os contadores dos codificadores ópticos incrementais. Isso é feito fazendo a leitura dos buffers que estão na placa de *Entrada/Saída* e que armazenam o valor da contagem dos codificadores. Este valor é salvo como valor inicial;
- int *McrIncEncGetCount(int _incEnc[NbIncEnc]): Faz a leitura dos buffers na placa de *Entrada/Saída* que fazem a contagem dos codificadores ópticos incrementais e retorna o valor da contagem dos seis codificadores;
- McrIncEncClearIndex():
- McrIncEncGetIndex(int _incEncIndex[NA]):
- McrAbsEncGetCount(int _absEnc[3]): Faz a leitura dos três codificadores ópticos absolutos e retorna os valores lidos;

A.8 Rbt.h:

O módulo Rbt.h possui funções dependente do robô, dos sensores proprioceptivos e acionadores. Ele é independente da configuração física dos componentes e de protocolos de comunicação e é independente também dos métodos de controle usados. As principais funções que compõem este módulo são:

- RbtInit(): Inicialização do módulo. Aplica 0V em todos motores e inicializa os valores do ganho $K_{c_{ij}}$ de cada codificador óptico (Seção 3.2.2). Inicializa os contadores dos codificadores ópticos em zero e faz a primeira leitura dos codificadores (ArtInit());
- RbtGetGains(Reel _artKP[NA]): Obtenção dos ganhos Kp_{ij} para o controle proporcional de cada um do seis motores. A variável _artKP[NA] possui seis campos e os valores de cada variável é definido nesta função especificamente para o robô Omni.
- Reel *RbtCmp(Reel _artErr[NA]): Compara a leitura das articulações por meio da leitura dos codificadores ópticos com a estimação feita da configuração das articulações na função void RbtChgInv(const Reel _opDeltaPosVeh[NO]).
- RbtCmd(const Reel _artDeltaCmd[NA]): Comando placas de controle de velocidade/corrente. O argumento de entrada desta função pode ser o comando de velocidade ou a variação de velocidade necessária durante um período, dependendo do tipo de controle, e seu valor é armazenado na variável _artDeltaCmd[NA]. Esta variável pode assumir valores entre -1, para velocidade mínima, e +1, para velocidade máxima, sendo 0 correspondente a velocidade nula no motor correspondente. Para o robô Omni, o estado parado é igual a tensão de 0V sobre o motor e a velocidade máxima corresponde a 10V.
- RbtIncEncInit(): Inicialização dos codificadores ópticos. Faz a leitura dos codificadores absolutos e obtém a configuração inicial de cada roda e armazena este valor em _ArtPosCount0. Em seguida faz a inicialização dos codificadores ópticos incrementais (McrIncEncInit());
- RbtIncEncGetCount(int _incEncCount[NA]): Obtém a leitura dos codificadores ópticos incrementais considerando as condições iniciais das rodas. São lidos os codificadores incrementais (McrIncEncGetCount(_incEncCount)), é feita a transformação para considerar o acoplamento do sensor na roda o *offset* devido a configuração inicial dos motores de direção e é retornado o valor modificado da leitura dos codificadores ópticos, em *ticks*;
- Reel *RbtOpDeltaPosSolToVeh(Reel _opDeltaPosVeh[NO], const Reel _opDeltaPosSol[NO], const Reel _opPos[NO]) : Transforma pequenos deslocamentos relativos no espaço operacional (do sistema de referência global)para o sistema de referência do robô. As variáveis _opDeltaPosSol[NO]e _opPos[NO]representam do deslocamento relativo e a posição anterior do robô no sistema de referência global. O deslocamento relativo no sistema de coordenadas do robô é calculado então como:

$$\begin{pmatrix} \Delta x^R \\ \Delta y^R \end{pmatrix} = \begin{pmatrix} \cos\left(\bar{\theta}\right) & \sin\left(\bar{\theta}\right) \\ -\sin\left(\bar{\theta}\right) & \cos\left(\bar{\theta}\right) \end{pmatrix} \begin{pmatrix} \Delta x^G \\ \Delta y^G \end{pmatrix}$$
(1)

com $\bar{\theta} = \theta + \frac{\Delta \theta}{2}$, sendo θ o ângulo de rotação antes do deslocamento e $\Delta \theta$ o ângulo realizado durante o pequeno deslocamento. $\Delta \theta^R = \Delta \theta$ e Δx^R e Δy^R são retornados nesta função por meio da variável _opDeltaPosVeh[NO].

- void RbtChgInv(const Reel _opDeltaPosVeh[NO]): ou Mudança inversa de coordenadas. Transforma pequenos deslocamentos relativos ao sistema de referência do robô para o sistema de referência das articulações. A partir do posicionamento do robô no sistema de coordenadas do robô, representado por _opDeltaPosVeh[3]), obtém-se as configurações articulares respectivas, armazenadas na variável _ArtDeltaPosConsCount[6].
- void RbtChgDir(const int _artDeltaPosCount[NA], const int _artPosCount[NA], Reel _opDeltaPos[NO], const Reel _opPos[NO]): ou mudança de coordenadas direta. Transforma pequenos deslocamentos relativos observados nas articulações para o sistema de referência global. Dados as leituras do codificadores ópticos, _artDeltaPosCount, a configuração articular anterior ao deslocamento β, _artPosCount, e a posição armazenado antes do deslocamento, _opPos, obtém-se o deslocamento relativo do robô em coordenadas globais, armazenadas em _opDeltaPos.

A.9 Art.h: Leitura dos codificadores ópticos incrementais e absolutos

Este módulo é de nível 2 e faz a leitura dos codificadores ópticos incrementais e bsolutos, como apresentado na seção 3.2.2. Os codificadore ópticos absolutos só são lidos na inicialização e o ângulo de direção é obtido infrementando ou decrementando este valor aos deslocamentos incrementais lido pelos codificadores ópticos incrementais. Os valores lidos são apresentados na unidade *ticks*. As funções de tal módulo são:

- void ArtInit(): Inicialização do módulo de leitura dos codificadores ópticos. Primeiramente a leitura dos codificadores ópticos incrementais são setados em zero e é feita a primeira, e única, leitura dos ódificadores ópticos absoluto, e sua leitura é convertida para radianos. Em seguida, é feito uma primeira leitura dos codificadores incrementais.
- void ArtUpdateData(): Leitura dos codificadores ópticos incrementais. Nesta função é salvo os dados da última leitura na variável local _ArtPosCountPree então é feito a leitura nos seis codificadores ópticos incrementais e os dados são salvos na variável local _ArtPosCount. O instante no qual a leitura é feita é salva.
- int *ArtGetData(unsigned long *_localTime, int _artDeltaPosCount[NA], int _artPosCount[NA]) Retorna a variação angular entre as duas últimas leituras dos codificadores ópticos realizada.

A.10 St.h: primitivas relativos ao computador central

O módulo ST.h possui as primitivas que são dependentes do micro computador. Não há nenhuma referência ao robô ou acesso a nenhum dos seus componentes, apenas as portas de entrada/saída do computador central. As funções de tal módulo são:

- Procedimentos relacionados ao relógio:
 - McrTimerInit(): Inicializa o relógio central que conta o tempo de execução do experimento a partir do momento da inicialização. Utiliza função *RTX* para ter maior precisão na leitura do tempo.
 - McrTimerStop(): Função vazia. Não realiza ação alguma.
 - unsigned long McrTimerTickToMicroS(unsigned long Tick): retorna o tempo lido pelo relógio na unidade micro segundos. Transforma a leitura do relógio de *ticks* (dado de entrada) para micro segundos.
 - unsigned long McrTimerMicroSToTick(unsigned long MicroS): Retorna a leitura de tempo em *ticks*. Transforma a leitura do tempo de micro segundos (dado de entrada) para *ticks*.
 - unsigned long McrTimerGetCount(): Faz a leitura do relógio. A saída de tal procedimento é o valor do relógio lido em *ticks*.
- Funções para o envio e leitura de dados via porta serial:
 - void McrComSetChan(int Channel): Seta o canal de comunição. Define como sendo o canal da comunicação o endereço armazenado em *Channel*. Os valores que *Channel* pode asumir são:
 - * COM1: Porta serial padrão do micro computador;
 - * PCI_COMO: Porta serial COMO presente na placa de comunição Entrada/Saída;
 - * PCI_COM1: Porta serial COM1 presente na placa de comunição Entrada/Saída;
 - * PCI_COM2: Porta serial COM2 presente na placa de comunição Entrada/Saída;
 - * PCI_COM3: Porta serial COM3 presente na placa de comunição Entrada/Saída;
 - * C012: Porta serial que comunica com os *transputers* que faz a leitura dos dados do girômetro a laser, ladar a laser e do leitor de balisa;
 - * C0124;
 - * C0125;
 - int McrComGetChan(): Retorna o último canal de comunicação definido na função acima.
 - int McrComInit(int Bps_100): Inicializa a comunicação com a porta serial definida em void McrComSetChan(int Channel);

- int McrComGetRxRdy(): Realiza a leitura do status porta serial definida por *Channel* e retorna os dados dos campos sobre a permissão para leitura em tal canal;
- char McrComGetChar(): Faz a leitura do caractere recebido pela porta serial com o endereço definido por *Channel*.
- int McrComGetErr(): Faz a leitura do erro recebido pela porta serial com o endereço definido por *Channel*. O primeiro bit do valor de erro retornado pela função indica o erro de paridade (não é usado), o segundo bit indica o *framing error* e o terceiro bit indica o *overrun error*.
- int McrComGetTxRdy(): Faz a leitura dos dados enviados pela porta serial e retorna os dados dos campos sobre a permissão de escrita em tal canal;
- void McrComPutChar(char C): Envia o caractere C por meio da porta serial definida.
- void McrComBreak(int _setBreak):
- Funções para o envio e leitura de palavras via porta serial:
 - void *ComGetMem(void *Dest, int Count): Faz a leitura de Count caracteres recebidos via porta serial e armazena tal palavra em Dest, caso este não seja nulo (NULL);
 - void ComPutMem(const void *Src, int Count): envia os Count caracteres contidos na palavra Src via porta serial;
 - char *ComGetS(char *Dest): lê a palavra enviada via porta serial e a armazena em Dest. Faz a leitura de caracteres até que receba o caractere indicando o fim dos dados ('r');
 - void ComPutS(const char *Src): Envia a string Src via porta serial.

A.11 Tra.h

O módulo Tra.h possui as funções que geram os pontos intermediários da trajetória e as leis de controle. A curva gerada é de Bezier, como apresentado em 5. As principais variáveis globais desta função são a posição desejada, no sistema de coordenadas globais,_OpPos, o deslocamento relativo desejado entre dois períodos de amostragem, _OpDeltaPos, e o ganho do controle proporcional, _KOpPos. Cada vetor tem dimensão 3, sendo que os dados corresponde as coordenadas de controle (X, Y, θ) . As principais funções deste módulo são:

 void TraInit(): Inicialização do módulo. Incializa o ganho proporcional do controle com o valor 1, verifica o primeiro ponto de controle da trajetória, que pode se estar na lista com pontos de controle (ListeGetPoint()) ou definido no módulo *Est.h* e obtido pela funçãoEstGetOpPos. Em seguida, é aplicado a lei de controle, por meio da função TraSetOpPos(McrTimerGetCount(), _OpNul , opPos), tendo o ponto de controle (opPos) como argumento de entrada;

- void TraSetOpPos(unsigned long _localTime, const Reel _opDeltaPos[NO], const Reel _opPos[NO]): Aplica o controle no ponto de controle. A partir dos pontos de controle _opDeltaPose _opPos, é aplicado o controle proporcional nestes pontos e armazenados nas respectivas variáveis globais _OpDeltaPose _OpPos;
- void TraClrOpPos(): É incializado os valores da variável global _OpPoscom zero;
- void TraUpdateData(): gerador de trajetória a partir dos pontos de passagem fornecido pelo algoritmo de planificação da trajetória, baseado em curva de Bezier. A partir dos pontos de controle lidos da lista (ListeGetPoint) ou definido no módulo *Est.h* (EstGetOpPos) são gerados os pontos intermediários de controle e é aplicado a lei de controle e esses valores são armazenados nas variáveis globais _OpDeltaPose _OpPos;
- Reel *TraGetData(unsigned long *_localTime, Reel _traDeltaOpPos[NO], Reel _traOpPos[NO]; são lidos os pontos de controle. São armazenados nas variáveis _traDeltaOpPose _traOpPosos valores dos pontos de controle obtidos a partir das variáveis globais _OpDeltaPose _OpPosdividos pelo ganho do controle correspondente, _KOpPos;