

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**ANÁLISE COMPARATIVA DAS TÉCNICAS DE
CONTROLE SERVO-VISUAL DE MANIPULADORES
ROBÓTICOS BASEADAS EM POSIÇÃO E EM IMAGEM**

GLORIA LILIANA LÓPEZ MUÑOZ

**ORIENTADOR: JOSÉ MAURICIO TORRES SANTOS DA MOTTA
DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS**

**PUBLICAÇÃO:
BRASÍLIA/DF: AGOSTO – 2011**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**ANÁLISE COMPARATIVA DAS TÉCNICAS DE CONTROLE
SERVO-VISUAL DE MANIPULADORES ROBÓTICOS BASEADAS
EM POSIÇÃO E EM IMAGEM**

GLORIA LILIANA LÓPEZ MUÑOZ

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA
DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
MESTRE EM SISTEMAS MECATRÔNICOS.**

APROVADA POR:

**Prof. José Mauricio Santos Torres da Motta, Dr. (FT/UnB)
(Orientador)**

**Prof. Carla Maria Chagas e Cavalcante Koike, Dr. (CIC/UnB)
(Examinador Interno)**

**Prof. Lourdes Mattos Brasil, Dr. (FGA/UnB)
(Examinador Externo)**

BRASÍLIA/DF, 11 DE AGOSTO DE 2011

FICHA CATALOGRÁFICA

MUÑOZ, GLORIA LILIANA LÓPEZ

Análise comparativa das técnicas de controle servo visual de manipuladores robóticos baseadas em posição e em imagem [Distrito Federal] 2011.

xvii, 109p., 210 x 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2011).
Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

1. Processamento de imagens digitais 2. Manipuladores robóticos
3. Controle servo-visual 4. Simulação

I. ENM/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

MUÑOZ, G. L. L. (2011). Análise Comparativa das Técnicas de Controle Servo Visual de Manipuladores Robóticos Baseadas em Posição e em Imagem. Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-40A/11, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 109p.

CESSÃO DE DIREITOS

AUTOR: Gloria Liliana López Muñoz

TÍTULO: Análise Comparativa das Técnicas de Controle Servo Visual de Manipuladores Robóticos Baseadas em Posição e em Imagem

GRAU: Mestre

ANO: 2011

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Gloria Liliana López Muñoz
SCLN 407 Bloco A, Sala 111
34.316.203 – Brasília – DF – Brasil.

Aos meus pais, Gloria e Jairo.

Ao Jesús

AGRADECIMENTOS

Agradeço a Deus,

Aos meus Pais, Gloria Muñoz e Jairo Lopez, pelo amor e apoio em cada passo de minha vida, aos meus irmãos que sempre tem estado comigo, animando-me e demonstrando-me seu carinho, ao Jesús, meu amor, pelo seu apoio incondicional, e sua mão sempre estendida nos momentos mais difíceis,

Ao meu orientador, o Professor Jose Mauricio Motta, pela amizade, a orientação, e a confiança depositada em mim.

À Universidade de Brasília, especialmente ao Professor Carlos Llanos, pela oportunidade de fazer o curso de mestrado neste país maravilhoso.

Aos meus amigos e colegas, que durante o curso de mestrado fizeram parte de minha vida, Diego, Diana, Ronald, Rodrigo, Camilo, Daniel, Claudia, Ana Maria, Ana Rosa, Jones, Edgar, Jhon, Luis, Dieguinho, Gerardo, Yesid, Marrocos, a todos eles agradeço pela amizade.

Finalmente, À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, pelo apoio financeiro durante o curso.

RESUMO

ANÁLISE COMPARATIVA DAS TÉCNICAS DE CONTROLE SERVO-VISUAL DE MANIPULADORES ROBÓTICOS BASEADAS EM POSIÇÃO E EM IMAGEM

Autor: Gloria Liliana López Muñoz

Orientador: José Mauricio Santos Torres da Motta

Programa de Pós-graduação em Sistemas Mecatrônicos.

Brasília, Agosto de 2011.

Este trabalho apresenta o desenvolvimento de duas metodologias de controle visual para manipuladores robóticos; a primeira, denominada Controle Servo Visual Baseado em Imagem, determina a variável de controle diretamente em função das características visuais, e a segunda, chamada Controle Visual Baseado em Posição, descreve o comando a partir da informação visual transformada do espaço da imagem ao espaço cartesiano. Na aplicação destas duas abordagens forem usados vários algoritmos de processamento de imagens e de controle cinemático de manipuladores, cada um dos quais cumpriu sua tarefa satisfatoriamente, permitindo ao final integrar os diferentes módulos envolvidos para avaliar a tarefa de controle visual como um todo.

Posterior à validação dos algoritmos de análise visual e de modelagem cinemática do manipulador, foi realizada uma etapa de simulação que permitiu testar o comportamento dos sistemas de controle visual projetados, aplicados em um manipulador robótico de seis graus de liberdade. Foi a partir desta simulação que se comprovou o cumprimento da tarefa de posicionamento das duas abordagens e se obtiveram características de desempenho como tempo de execução da tarefa e erro de posicionamento final.

O principal aporte desta pesquisa se apresenta no estudo comparativo de desempenho das duas metodologias propostas, verificando não só o cumprimento da tarefa de posicionamento, senão também, em alguns casos a robustez frente a variações do entorno, sendo estes últimos eventos que poderiam apresentar-se num entorno real, como por exemplo, ruído nas cenas, fraca calibração da câmera e erros na descrição do modelo do objeto. Contudo este trabalho se apresenta como uma etapa preliminar no desenvolvimento real de um sistema robusto de controle servo visual, que poderia ajudar na escolha da

metodologia apropriada para uma aplicação específica, mas outras considerações devem-se ter em conta, principalmente quanto à modelagem cinemática do manipulador robótico.

ABSTRACT

COMPARATIVE STUDY OF VISUAL SERVOING TECHNIQUES BASED ON IMAGE AND POSITION FOR ROBOTIC MANIPULATORS

Author: Gloria Liliana López Muñoz

Supervisor: José Mauricio Santos Torres da Motta

Programa de Pós-graduação em Sistemas Mecatrônicos

Brasília, Agosto de 2011

This work presents the development of two approaches of Visual Servoing Systems applied to robotic manipulators; the first one, named Image Based Visual Servoing (IBVS), determines the control variable directly as a function of visual features, the second one, called Position Based Visual Servoing (PBVS), describes the control from the visual information expressed in Cartesian coordinates. In order to apply these two approaches, several image processing and kinematic control algorithms were developed, each of them fulfilled its task satisfactorily, allowing, at the end, to integrate the modules involved to evaluate the visual control system as a whole.

Following the validation of visual analysis algorithms and the manipulator's kinematics model, a simulation step was performed allowing to test the behavior of visual control systems on a robotic manipulator of six degrees of freedom. It was proved, from this simulation, the fulfillment of the positioning task of the two approaches and a number of performance characteristics were obtained, as time for task execution and final positioning error.

The main contribution of this research is the comparative study of the performances of the two proposed methodologies, verifying not only the fulfillment of the task for positioning, but, in some cases, their robustness against environment variations. These former events could be present in real environments, such as noise, poor camera calibration, and errors in the object model representation. However, this work is presented as a preliminary step in the actual development of a robust visual servo control, which could help in choosing the appropriate methodology for a specific application. Other considerations must be taken into account, especially in regard to the kinematic modeling of the robotic manipulator.

SUMÁRIO

1 - INTRODUÇÃO	1
1.1- JUSTIFICATIVA DO TRABALHO.....	5
1.2- OBJETIVOS	5
1.2.1- Objetivo Geral.....	5
1.2.2- Objetivos Específicos.....	6
1.2.3- Metodologia	7
1.3- ESTRUTURA DO DOCUMENTO	8
2 - REVISÃO BIBLIOGRÁFICA	9
2.1- MODELAGEM GEOMÉTRICA	9
2.1.1- Espaço Projetivo e Coordenadas Homogêneas	9
2.1.2- Transformações Projetivas	10
2.1.3- Modelo de Câmera	11
2.1.4- Homografia entre planos	13
2.2- CALIBRAÇÃO DE CÂMERAS	15
2.2.1- Calibração com padrão plano.....	15
2.3- CONTROLE SERVO VISUAL.....	17
2.3.1- Controle Servo Visual Baseado em Imagem	18
2.3.2- Controle Servo Visual Baseado em Posição	20
2.3.3- Definição da Lei de Controle	21
2.3.4- Lei de Controle no Controle Servo Visual Baseado em Imagem (IBVS) ..	22
2.3.4.1- Matriz de Interação LS de Pontos.....	23
2.3.4.2- Estimativa de LS	26
2.3.5- Lei de Controle no Controle Servo Visual Baseado em Posição (PBVS) ..	26
3 - CONTROLE SERVO VISUAL	30
3.1- EQUIPAMENTO.....	30
3.2- PROCESSAMENTO DE IMAGENS.....	34
3.2.1- Extração de características	34
3.2.2- Casamento ou <i>Matching</i>	36
3.2.3- Casamento Robusto.....	38

3.2.3.1- Cálculo da Matriz de Homografia H	39
3.2.3.2- RANSAC.....	40
3.2.4- Cálculo da Postura do Objeto Alvo.....	43
3.3- LEI DE CONTROLE.....	46
3.4- CONTROLE CINEMÁTICO DO MANIPULADOR	48
3.4.1- Cinemática Direta do Robô.....	48
3.4.2- Jacobiano do robô J	50
3.4.3- Jacobiano Inverso J^{-1}	52
4 - RESULTADOS E ANÁLISE	54
4.1- ANÁLISE E PROCESSAMENTO DE IMAGENS	54
4.1.1- Detecção de Pontos	54
4.1.2- Casamento por correlação.....	59
4.1.3- Casamento Robusto.....	62
4.1.4- Estimativa da Postura da Câmera.....	68
4.1.5- Calibração da Câmera	73
4.2- LEI DE CONTROLE.....	74
5 - SIMULAÇÃO	77
5.1- INICIALIZAÇÃO E MODELAGEM DA CÂMERA	77
5.2- ROBÔ.....	80
5.3- CONTROLE	81
5.3.1- Controle Servo Visual baseado em Imagem.....	81
5.3.2- Controle Servo Visual baseado em Posição.....	82
5.4- RESULTADOS DE SIMULAÇÃO DO CONTROLE SERVO VISUAL.....	83
5.4.1- Controle Visual Baseado em Imagem.....	83
5.4.2- Controle Visual Baseado em Posição	86
5.5- ADIÇÃO DE ERROS NO SISTEMA	88
5.5.1- Erros de Calibração	88
5.5.2- Erros em Medidas 3D	89
5.5.2.1- Erro na estimativa da profundidade de cena Z	90
5.5.2.2- Erro no Modelo do Objeto.....	91

5.6- COMPARATIVA DO CONTROLE VISUAL BASEADO EM IMAGEM E O CONTROLE VISUAL BASEADO EM POSIÇÃO	93
6 - CONCLUSÕES E PROPOSTAS DE TRABALHOS FUTUROS.....	98
6.1- TRABALHOS FUTUROS.....	100
REFERÊNCIAS BIBLIOGRAFICAS	101
APÊNDICES	105
APÊNDICE A: DETERMINAÇÃO DE H (MATRIZ DE HOMOGRAFIA) A PARTIR DE CORRESPONDÊNCIAS	106
APÊNDICE B: DECOMPOSIÇÃO EM VALORES SINGULARES SVD	108
APÊNDICE C: PARÂMETROS DE DENAVIT-HATEMBERG	109

LISTA DE TABELAS

Tabela 3.1. Parâmetros D-H do Manipulador IRB2000.....	49
Tabela 4.1. Imagens de Teste	55
Tabela 4.2. Resultados da Detecção de Cantos	59
Tabela 4.3 Resultados do Algoritmo de Casamento por Correlação.....	62
Tabela 4.4. Comparação do cálculo da Homografia	63
Tabela 4.5. Resultados do Casamento Robusto.....	68
Tabela 4.6. Coordenadas dos oito vértices do cubo virtual no referencial do objeto.....	70
Tabela 4.7. Resultados do algoritmo de cálculo da postura das imagens das Figuras 4.11 e 4.12	73
Tabela 4.8. Cálculo das Velocidades de Câmera.....	75
Tabela 5.1. Modificação das Coordenadas do Modelo do Objeto.....	91
Tabela 5.1. Comparação entre o IBVS e PBVS, posição inicial próxima ao alvo	93
Tabela 5.2. Comparação entre o IBVS e PBVS, posição inicial afastada ao alvo	95

LISTA DE FIGURAS

Figura 1.1. Controle Visual (a) de Malha Aberta ou “Look- then-Move” (“Ver depois Mover”) (b) de Malha Fechada ou “Visual Servoing” (“Controle Servo Visual”).....	2
Figura 1.2. Configurações Câmera- Robô	3
Figura 2.1. Modelo de Câmera Pinhole, adaptado de (Hartley, et al., 2003)	11
Figura 2.2. Sensor tipo CCD adaptado de (Comport, 2005)	12
Figura 2.3. Projeção de um ponto em dois planos a traves de um centro de projeção	14
Figura 2.4. Controle Servo Visual	18
Figura 2.5. Diagrama de Blocos do IBVS	19
Figura 2.6. Diagrama de Blocos do PBVS	20
Figura 2.7. Modelo do movimento da câmera em PBVS, adaptado de (Malis, et al., 1998).....	27
Figura 3.1. Padrão de calibração	31
Figura 3.2 Objetos de Superfície Plana	31
Figura 3.3 Esquema do Controle Servo Visual Baseado em Imagem.....	32
Figura 3.4 Esquema de Controle Servo Visual Baseado em Posição.....	33
Figura 3.5 Aplicação do detector de cantos de Harris.....	35
Figura 3.6. Resultado da aplicação do algoritmo de casamento (a)Imagem do Alvo na posição inicial (b) Imagem do alvo na posição final (c) Imagens superpostas, casamento de pontos, as linhas vermelhas indicam o movimento de cada ponto entre as duas imagens.....	37
Figura 3.7. Aplicação do casamento (a) método VNC (b) método casamento Robusto.....	42
Figura 3.8. Referencial Associado ao Objeto	43
Figura 3.9. Relação entre cinemática direta e cinemática inversa, adaptado de (Barrientos, et al., 2007)	48

Figura 3.10. Referenciais das juntas na obtenção dos parâmetros D-H, (unidades em metros).....	49
Figura 3.11. Relação entre o Jacobiano e o Jacobiano Inverso, adaptado de(Barrientos, et al., 2007)	51
Figura 4.1. Detecção de Cantos com Algoritmo de Harris.....	58
Figura 4.2. Casamento por Correlação	60
Figura 4.3. . Casamento por correlação do Par 6 (Ver Tabela 4.1.), no círculo vermelho se destacam os casamentos incorretos	61
Figura 4.4. Casamento por correlação do Par 9 (Ver Tabela 4.1), no círculo vermelho se destacam os casamentos incorretos	61
Figura 4.5. Casamento de Pontos Par 1 (Ver Tabela 4.1) (a) com só Correlação (b) com o cálculo da Homografia por RANSAC	64
Figura 4.6. Casamento de Pontos Par 3 (Ver Tabela 4.1.) (a) com só Correlação (b) com o cálculo da Homografia por RANSAC	65
Figura 4.7. Casamento de Pontos Par 5 (Ver Tabela 4.1.) (a) com só Correlação (b) com o cálculo da Homografia por RANSAC	66
Figura 4.8. Casamento de Pontos Par 10 (Ver Tabela 4.1) (a) com só Correlação (b) com o cálculo da Homografia por RANSAC	67
Figura 4.9. Referencial do Objeto	69
Figura 4.10. Comportamento desejado do movimento do Cubo Virtual.....	70
Figura 4.11. Testes do Cálculo da Postura do Webcam com o objeto em duas posturas diferentes.....	71
Figura 4.12. Testes do Cálculo da Postura da Câmera CCD com o objeto em duas posturas diferentes	72
Figura 4.13. Padrão de Calibração.....	74
Figura 5.1. Diagrama do Sistema Simulado. O sub-índice c refere-se à câmera	77
Figura 5.2. Referenciais associados ao sistema (Robô, Câmera e Objeto Alvo) com suas respectivas transformações	78
Figura 5.3. Inicialização do sistema de controle visual.....	79

Figura 5.4. Obtenção da imagem atual dos pontos do objeto.....	80
Figura 5.5. Determinação da postura do elemento terminal a partir de sua velocidade.....	81
Figura 5.6 Controle Visual Baseado em Imagem.....	81
Figura 5.7 Controle Visual Baseado em Posição	82
Figura 5.8. Modelo em Simulink do IBVS	83
Figura 5.9. Casos de Estudo (a) Caso 1: Posição Inicial próxima à Posição Alvo (b) Caso 2: Posição Inicial afastada da Posição Alvo	84
Figura 5.10. Evolução da Trajetória dos pontos durante o controle IBVS (caso 1).....	84
Figura 5.11. Evolução da Velocidade da câmera durante o controle IBVS (caso 1)	84
Figura 5.12. Evolução da Trajetória dos pontos durante o controle IBVS (caso 2).....	85
Figura 5.13. Evolução da Velocidade da câmera durante o controle IBVS (caso 2)	85
Figura 5.14. Modelo em Simulink do PBVS.....	86
Figura 5.15. Evolução da Trajetória dos pontos durante o controle PBVS (caso 1).....	86
Figura 5.16. Evolução da Velocidade da câmera durante o controle PBVS (caso 1)	87
Figura 5.17. Evolução da Trajetória dos pontos durante o controle PBVS (caso 2).....	87
Figura 5.18. Evolução da Velocidade da câmera durante o controle PBVS (caso 2)	87
Figura 5.19. Evolução da Trajetória dos pontos e da Velocidade da câmera durante o controle IBVS em presença de erros de calibração (caso 1)	88
Figura 5.20. Evolução da Trajetória dos pontos e da Velocidade da câmera durante o controle PBVS em presença de erros de calibração (caso 1)	89
Figura 5.21. Evolução da Trajetória dos pontos e da Velocidade da câmera durante o controle IBVS com valor de profundidade Z real ($Z = 1\text{m}$)	90
Figura 5.22. Evolução da Trajetória dos pontos e da Velocidade da câmera durante o controle IBVS com valor de profundidade Z modificado ($Z = 0.1\text{m}$).....	90
Figura 5.23. Evolução da Trajetória dos pontos e da Velocidade da câmera durante o controle IBVS com valor de profundidade Z modificado ($Z = 10\text{m}$).....	91
Figura 5.24. Velocidade da câmera durante o controle PBVS com coordenadas do modelo modificado (caso 1)	92

Figura 5.25. Velocidade da câmera durante o controle PBVS com coordenadas do modelo modificado (caso 2) 92

LISTA DE ALGORITMOS

Algoritmo 1. Detecção de Cantos de Harris	36
Algoritmo 2. Casamento por Correlação	38
Algoritmo 3. Cálculo da Matriz de Homografia.....	40
Algoritmo 4. RANSAC	41
Algoritmo 5. Casamento Robusto	43
Algoritmo 6. Estimativa da Postura da Câmera	45
Algoritmo 7. Lei de Controle Visual	47
Algoritmo 8. Cinemática Direta do Manipulador IRB2000.....	50
Algoritmo 9. Jacobiano do IRB2000.....	53

LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

CCD	- <i>Charged Coupled Device</i>
IBVS	- <i>Image Based Visual Servoing</i> : Controle servo visual baseado em imagem
PBVS	- <i>Position Based Visual Servoing</i> : Controle servo visual baseado em posição
RANSAC	- <i>Random Sampled Consensus</i>
VNC	- <i>Variance Normalized Correlation</i> : Correlação por Variância Normalizada
D-H	-Denavit Hatemberg
SVD	- <i>Singular Value Decomposition</i> : Decomposição em Valores Singulares
DLT	- <i>Direct Linear Transformation</i> : Transformação Linear Direta
CAD	-Computer-Aided Design: Desenho Assistido por Computador

O	Origem do sistema coordenado
X	Matriz ou vetor
x	Escalar
x^*	Vetor de valores esperados da variável x
X^T	Transposta matriz X
X^{-1}	Inversa da matriz X
X^+	Pseudo Inversa da matriz X
\hat{X}	Valor aproximado da Matriz ou vetor X
\tilde{X}	Matriz ou vetor em representação homogênea
aT_b	Matriz ou vetor de Transformação do referencial b para o referencial a
at_b	Vetor de Translação do referencial b para o referencial a
aP	Vetor em coordenadas do referencial a
R	Matriz de Rotação
H	Matriz de Homografia
P	Matriz de Projeção de uma câmera
K	Matriz de parâmetros intrínsecos de uma câmera ou matriz de calibração
f	Comprimento Focal
(u, v)	Coordenadas de um ponto numa Imagem
(u_0, v_0)	Coordenadas do ponto central numa Imagem
(s_x, s_y)	Tamanho do Píxel nas direções x e y respectivamente, dado em $[\frac{pixel}{metro}]$

π_i	Plano i
e	Erro
\mathbf{v}_c	Velocidade da Câmera
q_i	Coordenadas de Juntas num manipulador robótico
\dot{q}_i	Velocidade de Juntas num manipulador robótico
\mathcal{J}	Jacobiano de um manipulador robótico
\mathcal{J}^{-1}	Jacobiano Inverso de um manipulador robótico

1 - INTRODUÇÃO

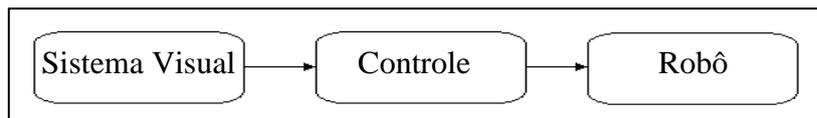
Os robôs são ferramentas que atualmente se desempenham em distintos níveis da nossa sociedade, abrangendo desde a indústria de grande porte até o entretenimento, passando por aplicações educativas ou de salvamento. Um aspecto inerente na concepção de um sistema robótico é o seu controle, o qual pretende que o sistema não só realize a tarefa para a qual foi desenvolvido, mas também que saiba reagir quando se apresentem situações não desejadas e inesperadas. Por essa razão se faz uso de diferentes tipos de sensores, entre os quais estão aqueles que dão uma noção ao sistema do espaço no qual se está realizando a tarefa (ambiente de trabalho) e, a partir dessa informação, tomar as decisões pertinentes.

Um dos sensores com maior capacidade para cumprir a tarefa de extração da informação do ambiente é o sensor visual já que, como se conhece por experiência própria, a visão junto com a nossa capacidade de interpretação, proporcionam a maior quantidade de dados do entorno permitindo interagir com ele. Deste modo, o uso de sensores visuais e seus sistemas de interpretação em aplicações robóticas permitem uma melhor percepção do seu entorno ampliando assim suas aplicações em ambientes dinâmicos, como exteriores, ou no desempenho de tarefas não repetitivas.

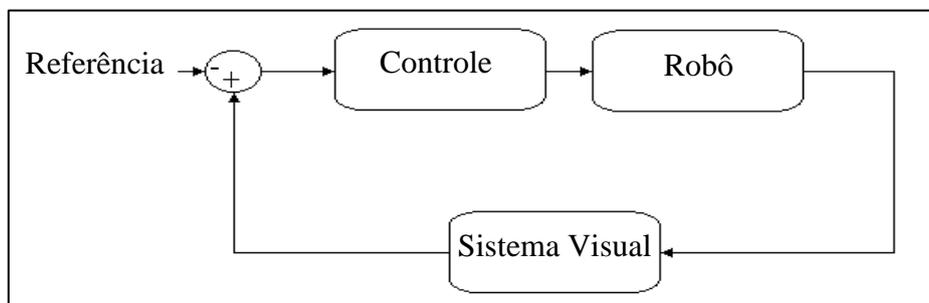
O uso de sensores visuais no controle de manipuladores robóticos tem sido de amplo interesse dentro da comunidade científica em virtude da sua grande riqueza de informação, sua natureza não invasiva e a sua capacidade de integração nos sistemas robóticos. Os primeiros relatórios deste tipo de aplicações datam dos anos 1960 (McCarthy, et al., 1968), mas não foi antes da década dos anos 1990 que a técnica apresentou seus maiores avanços, principalmente pelo desenvolvimento paralelo de tecnologias nas áreas de processamento de imagens assim como da microeletrônica, o que permitiu o surgimento de sistemas de aquisição e análise de imagens mais robustos e eficientes.

O problema de deslocar um manipulador robótico, para atingir certa postura partir da informação de uma cena fornecida por uma ou mais câmeras, tem sido abordado de formas diferentes (Figura 1). Inicialmente, a maior parte dos trabalhos relata o uso de uma abordagem em malha aberta, onde a extração da informação visual e o controle do movimento são duas tarefas totalmente separadas, neste tipo de sistemas denominados “Ver depois Mover” *“Look-then-Move”* (Corke, 1993) o sistema visual determina a

posição alvo do manipulador e o controle ordena ao robô realizar um movimento cego até esse ponto, assumindo que o entorno permanece estático desde o início do movimento. Ainda que este tipo de sistemas seja bastante simples, sua aplicação está limitada a tarefas altamente específicas e seu desempenho depende da correta calibração tanto do sistema mecânico como do sistema visual. Posteriormente, e com o fim de melhorar a precisão, (Shirai, et al., 1973) foi introduzido dentro do sistema de controle o sinal de realimentação visual em forma de malha fechada, este tipo de sistema é conhecido como Controle Servo Visual (“Visual Servoing”) (Chaumette e Hutchinson, 2006). No controle servo visual, a informação visual é introduzida a cada ciclo do controle com o fim de corrigir a posição do manipulador até atingir a posição desejada. Esta abordagem tem demonstrado sua capacidade de efetuar eficientemente a tarefa de controle de manipuladores (Marchand e Chaumette, 2002) e, atualmente, é a técnica mais utilizada no controle visual de sistemas robóticos.



(a)



(b)

Figura 1.1. Controle Visual (a) de Malha Aberta ou “Look- then-Move” (“Ver depois Mover”) (b) de Malha Fechada ou “Visual Servoing” (“Controle Servo Visual”)

Os manipuladores robóticos são, na atualidade, predominantemente encontrados desempenhando tarefas programadas ou em ambientes altamente estruturados. O controle servo visual, abordado neste trabalho, fornece aos manipuladores robóticos uma maior flexibilidade e, por tanto, sua possível atuação em cenários mais comuns, tanto na indústria assim como nas aplicações cotidianas. Alguns exemplos destas aplicações poderiam ser processos de manipulação de objetos moveis, tarefas de controle em ambientes exteriores dinâmicos, como ambientes subaquáticos, desenvolvimento de lares inteligentes, enfim,

tarefas nas quais o sistema tenha que se adaptar às incertezas do ambiente e não o contrário. Neste trabalho, realiza-se o controle de posicionamento de um manipulador robótico mediante técnicas de controle servo visual, de forma a verificar sua aplicabilidade a manipuladores industriais, de forma simulada, com objetivo de estender sua aplicação em tarefas reais como as mencionadas acima.

Existem diferentes formas de classificar o controle servo visual em função de critérios como a configuração câmera-robô e a definição do erro. As topologias apresentadas de acordo com a configuração física do sistema visual podem-se classificar em função do número de câmeras utilizadas, em *Monocular*, com só uma câmera; *Estéreo* ou *Binocular*, que utiliza duas câmeras e *Redundante*, que utiliza mais de duas câmeras; ou de acordo com a posição da câmera com respeito ao robô, em *Câmera Fixa* (“eye to hand”), onde a câmera encontra-se imóvel durante o processo; e *Câmera Embarcada* (“eye in hand”), onde a câmera é fixada sobre o robô e movimenta-se solidariamente com ele. A Figura 1.2 apresenta um resumo destas configurações (Kragic e Christensen, 2002).

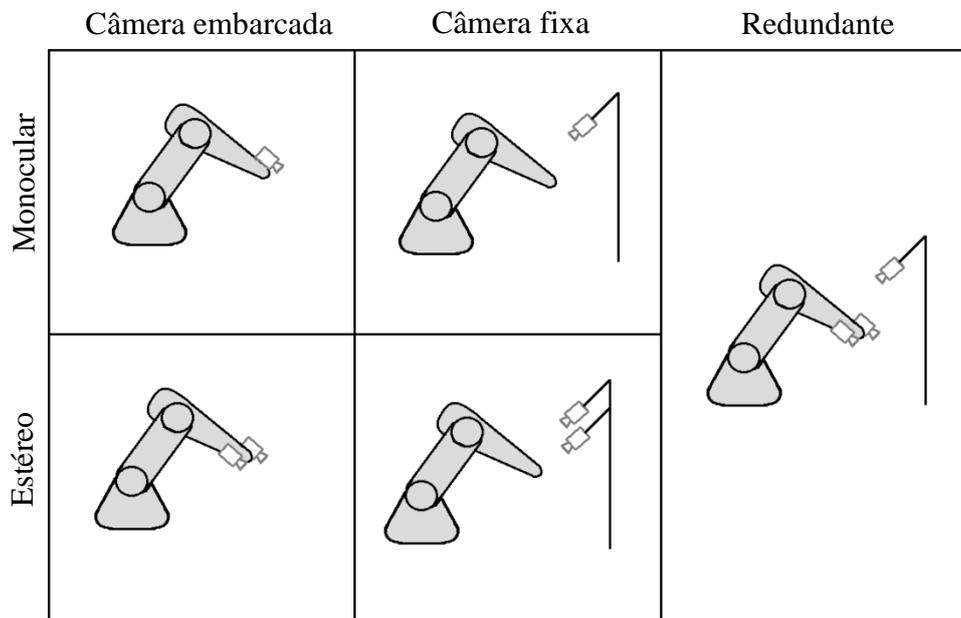


Figura 1.2. Configurações Câmera- Robô

Quanto à definição do erro o controle servo visual pode ser classificado em controle servo visual baseado em posição ou 3D (PBVS: *Position Based Visual Servoing*), o qual define o erro no espaço tridimensional; o controle baseado em imagem ou 2D (IBVS: *Image Based Visual Servoing*), que define o erro no espaço de características da imagem; e o controle

híbrido ou 2 ½ D, uma combinação dos dois anteriores, que define o erro tanto no espaço da imagem como no espaço tridimensional (Malis,1998).

No presente trabalho, faz-se um estudo em forma simulada de dois dos principais tipos de controle visual: o controle baseado em imagem IBVS e o controle baseado em posição PBVS, com o fim de comparar seus desempenhos em condições similares e, desta forma, ter uma idéia preliminar de qual destas metodologias se adapta melhor a parâmetros de contorno específicos que, em condições reais, poderiam ser difíceis de controlar, como por exemplo:

- A iluminação da cena;
- A posição inicial da câmera;
- Os parâmetros de calibração da câmera e
- O modelo do objeto.

Desta forma, foram estudados diferentes casos em que se incluem parâmetros de contorno citados anteriormente nas duas metodologias de controle em estudo.

Para tal fim, foram desenvolvidos os algoritmos de cada um dos tipos de controle para uma configuração de uma câmera (monocular) embarcada num manipulador robótico de 6 graus de liberdade. Esses algoritmos baseiam-se em diferentes conceitos teóricos encontrados em referências bibliográficas estudadas no desenvolvimento da pesquisa, mas a grande maioria deles foi elaborada especificamente para esta pesquisa, como descrito no capítulo 3.

O sistema desenvolvido baseia-se no cálculo da relação de homografia existente entre as imagens de objetos planos e na reconstrução da cena para o caso do controle baseado em posição. Além disso, foi feita a modelagem da lei de controle para cada um dos casos. Os resultados da aplicação dos algoritmos numa plataforma virtual permitiram a comparação direta dos dois tipos de controle procurando assim encontrar as condições de desempenho ótimas para cada uma das metodologias. Desta forma o trabalho apresentado pode ser considerado como uma etapa preliminar na construção de um sistema de controle servo visual em uma plataforma real.

1.1- JUSTIFICATIVA DO TRABALHO

É comum encontrar robôs em ambientes industriais altamente estruturados ou desenvolvendo tarefas repetitivas. No entanto, é difícil vê-los em aplicações em exteriores ou em ambientes dinâmicos onde os elementos podem variar de posição ou de forma, mas existe um interesse crescente em encontrar métodos para poder ampliar a faixa de atuação dos sistemas robóticos, para o qual se faz necessária a integração dos sistemas robóticos com sensores que forneçam uma informação completa e confiável do entorno. Dentre os sensores desenvolvidos para este propósito, o mais destacado é sensor visual, neste caso, a câmera, que junto com os sistemas de interpretação proporcionam não só a localização como também a geometria dos elementos dentro uma cena.

Dois dos métodos mais populares nos quais se faz uso da informação visual fornecida por uma câmera, para realizar o controle de movimento de um robô são os denominados Controle Servo Visual Baseado em Imagem e Controle Servo Visual Baseado em Posição. Porém são poucas as publicações nas quais se faz uma comparação destas duas metodologias sob uma mesma base teórica.

Neste trabalho, faz-se a implementação e o estudo destas duas técnicas de controle a fim de estabelecer tanto suas virtudes como limitações sob características de entorno similares e pretendendo, assim, fornecer uma base comparativa de desempenho que poderia ajudar na escolha de uma metodologia adequada de controle visual em diferentes tipos de aplicações.

1.2- OBJETIVOS

1.2.1- Objetivo Geral

Desenvolver os algoritmos de Controle Servo Visual Baseado em Posição (PBVS) e de Controle Servo Visual Baseado em Imagem (IBVS) com o fim de realizar uma análise comparativa de seus desempenhos em função de variáveis de entorno similares.

1.2.2- Objetivos Específicos

Com o fim de atingir o objetivo geral desta pesquisa forem propostos os seguintes objetivos específicos:

- Garantir o correto funcionamento de cada um dos algoritmos, desenvolvidos durante as diferentes etapas, de forma individual, mediante a apreciação visual e análise estatística de erros.
- Desenvolver uma etapa de integração dos algoritmos de análise e processamento de imagens, de aplicação da lei de controle e da cinemática do robô, na forma de uma plataforma simulada.
- Testar o desempenho de cada uma das técnicas de controle sob estudo em quatro circunstâncias diferentes:
 1. Condições de contorno ideais: livres de erros nos parâmetros de calibração da câmera e nos parâmetros de medição 3D;
 2. Na presença de erros nos parâmetros de calibração da câmera;
 3. Na presença de erros nos parâmetros de medição do modelo, para o controle visual baseado em posição, ou nos parâmetros de medida da profundidade da cena, no controle visual baseado em imagem e
 4. Na presença tanto de erros de calibração como de erros nas medições 3D.

Comparar os resultados obtidos de cada um dos tipos de controle a partir da análise do comportamento da trajetória dos pontos nas cenas durante a realização do controle, assim como da evolução da velocidade da câmera .

1.2.3- Metodologia

Com o fim de desenvolver os sistemas simulados de controle servo visual baseado em imagem e o controle servo visual baseado em posição para sua posterior comparação de desempenho, propõem-se os seguintes passos:

- Determinar as características na imagem que identificarão os objetos alvo utilizados nesta pesquisa.
- Obter os parâmetros de calibração da câmera utilizada na aquisição de imagens.
- Desenvolver um sistema de *Casamento de Características* ou *Matching*, entre duas imagens do mesmo objeto sob diferentes vistas, com o fim de rastreá-las e determinar a variação de suas posições.
- Determinar a postura da câmera em função do referencial do objeto alvo, a partir de uma imagem, dos parâmetros intrínsecos da câmera e do modelo do objeto.
- Definir o espaço de parâmetros no qual se define a lei de controle de cada uma das metodologias propostas.
- Aplicar uma lei de controle proporcional, cuja variável de controle seja a velocidade da câmera, aos dois modelos de controle servo-visual.
- Determinar a velocidade da câmera em cada um dos casos de estudo.
- Transformar as variáveis do espaço cartesiano ao espaço de juntas do robô, mediante o Jacobiano Inverso do Manipulador.
- Desenvolver a cinemática direta do manipulador para a tarefa de posicionamento do elemento terminal, que neste caso particular é a mesma posição e orientação da câmera.
- Modelar a projeção da câmera a partir de seus parâmetros de calibração e das matrizes de transformação de coordenadas nos sistemas de referência do objeto e do elemento terminal do robô.

- Calcular a velocidade do elemento terminal do manipulador necessária para atingir o objetivo, fazendo uso dos modelos da câmera e do robô.
- Modelar os sistemas de controle visual mediante a aplicação da lei de controle para cada caso proposto, assumindo que a câmera encontra-se inicialmente em uma posição diferente à posição que se quer atingir.

1.3- ESTRUTURA DO DOCUMENTO

O documento apresentado encontra-se dividido em seis capítulos; no capítulo dois é feita uma revisão bibliográfica das áreas relevantes nesta pesquisa. Inicialmente são abordados conceitos geométricos envolvidos na formação de imagens e introduzido o modelo de câmera estudado. Posteriormente são tratados os sistemas de controle servo visual identificando as principais metodologias e estabelecidas as diferenças entre as duas abordagens implementadas. Já no capítulo três é analisado o controle servo visual em função da suas etapas constitutivas; cada uma destas etapas é descrita detalhadamente e são apresentados os algoritmos aplicados a elas. Neste capítulo também é introduzido o modelo cinemático do manipulador robótico utilizado. No capítulo quatro são apresentados os resultados da aplicação dos algoritmos de processamento de imagens e controle cinemático para sua posterior integração nos modelos de controle. No capítulo cinco é feito um estudo comparativo das duas técnicas de controle servo visual, o controle baseado em imagem e o controle baseado em posição. Inicialmente são descritas as características do sistema de simulação e posteriormente são apresentados os resultados de sua aplicação. Finalmente, no capítulo seis são apresentadas as conclusões do trabalho e mostradas algumas propostas de trabalhos futuros.

2 - REVISÃO BIBLIOGRÁFICA

2.1- MODELAGEM GEOMÉTRICA

As imagens são representações visuais em planos bidimensionais de cenas do mundo real, ou seja, trata-se de uma projeção ou mapeamento do espaço 3D num plano 2D. Com o fim de compreender uma imagem e as medições feitas sobre ela, precisa-se conhecer a cena que está sendo projetada, ou mais exatamente, a forma na qual foi obtida a imagem, considerando-se que estando disponível um modelo de projeção é possível inferir as relações entre o mundo real e a imagem analisada. Esse processo de transformação espacial, entre espaço e imagem, é feito por câmeras e pode-se descrevê-lo através da *Geometria Projetiva*. Apresentam-se a seguir algumas noções básicas de geometria projetiva e, em seguida, descrevem-se o modelo de câmera e outras entidades que serão empregadas ao longo deste trabalho.

2.1.1- Espaço Projetivo e Coordenadas Homogêneas

Os objetos em geral, tanto aqueles que são manipulados diariamente como aqueles que se encontram num ambiente industrial, podem ser descritos pela geometria Euclidiana. De fato, o próprio espaço entende-se de forma intuitiva em Coordenadas Cartesianas, que são uma representação no espaço Euclidiano de três dimensões. Porém a geometria euclidiana tem uma dificuldade na definição de alguns conceitos básicos como o de linhas paralelas, em que duas linhas se interceptam no “infinito”, o que cria certa ambiguidade. Uma forma de ultrapassar esse aspecto é adicionar pontos ao infinito, onde as linhas paralelas se cruzam, denominados *pontos ideais* (Hartley e Zisserman, 2003). Esta modificação do espaço euclidiano o transforma em uma nova classe de objeto geométrico, o *espaço projetivo* \mathbb{P}^n , tal que o espaço projetivo seja simplesmente uma extensão do espaço euclidiano, o que permite a manipulação de entidades geométricas “ao infinito” (Malis, 1998).

Um ponto no espaço Euclidiano n -dimensional \mathbb{R}^n , pode ser descrito mediante *coordenadas homogêneas*, através de um vetor $(n+1)$ -dimensional. Assim, o vetor $\mathbf{X} = (x_1, x_2, \dots, x_n)$ em coordenadas cartesianas, torna-se em $(\mathbf{X}, 1) = (kx_1, kx_2, \dots, kx_n, k)$, em coordenadas homogêneas, onde k é um fator de escala não nulo (Barrientos, et al., 2007).

Um vetor no espaço euclidiano tridimensional \mathbb{R}^3 então, em coordenadas homogêneas, se representa por:

$$\begin{bmatrix} kx_1 \\ kx_2 \\ kx_3 \\ k \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \quad (2.1)$$

, sendo esta sua forma no espaço projetivo \mathbb{P}^3 , onde os vetores nulos se representam por $[0, 0, 0, k]$, com k não nulo e os vetores da forma $[x_1, x_2, x_3, 0]$ representam os pontos “ao infinito” (a divisão por zero representa vetores de comprimento infinito).

Observa-se, assim, que os pontos no espaço euclidiano \mathbb{R}^n podem-se estender ao espaço projetivo \mathbb{P}^n mediante sua representação em coordenadas homogêneas, sendo possível retornar do espaço projetivo ao espaço euclidiano com uma simples divisão pelo fator de escala k .

2.1.2- Transformações Projetivas

Uma transformação projetiva ou projetividade é um mapeamento linear entre pontos em coordenadas homogêneas no espaço projetivo. Por esta razão a mudança das coordenadas cartesianas para coordenadas homogêneas permite descrever de uma maneira mais simples algumas transformações do espaço euclidiano. Um exemplo é a localização de corpos rígidos que, no espaço euclidiano tridimensional, representa-se como:

$${}^b\mathbf{X} = {}^b\mathbf{R}_a {}^a\mathbf{X} + {}^b\mathbf{t}_a \quad (2.2)$$

, onde ${}^a\mathbf{X}$ e ${}^b\mathbf{X}$ representam as posições de um ponto em relação aos referenciais a e b , respectivamente; ${}^b\mathbf{R}_a$ e ${}^b\mathbf{t}_a$ são as matrizes de rotação e translação que permitem a mudança entre o referencial a e o referencial b . Esta mesma transformação, tratada como uma projetividade, ou seja, em coordenadas homogêneas, se representa como uma relação linear, tal que:

$${}^b\tilde{\mathbf{X}} = \mathbf{T} {}^a\tilde{\mathbf{X}} \quad (2.3)$$

, onde $\tilde{\mathbf{X}}$ é a representação em coordenadas homogêneas de \mathbf{X} (Comport, 2005).

Como visto anteriormente, uma imagem é uma projeção do mundo (3D) no plano da imagem; este tipo de mapeamento, assumindo que o sistema de coordenadas do mundo esteja representado em coordenadas homogêneas, é uma transformação projetiva e se representa mediante uma matriz homogênea, que descreve o processo de formação da imagem. Tal aspecto será visto com mais detalhe nas próximas seções quando se formule o modelo de câmera.

2.1.3- Modelo de Câmera

O modelo de câmera utilizado é o modelo de *projeção central* ou *pinhole* que, apesar de sua simplicidade, representa de forma precisa a formação de uma imagem a partir de câmeras CCD que são as de maior interesse no escopo deste trabalho.

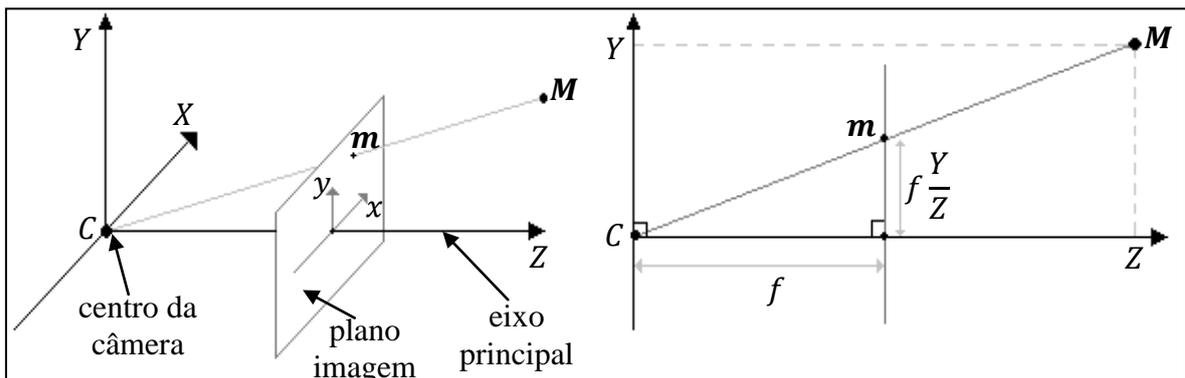


Figura 2.1. Modelo de Câmera Pinhole, adaptado de (Hartley, et al., 2003)

O modelo assume como centro de projeção C a origem de um referencial euclidiano, no qual convergem os feixes de luz emitidos por pontos no espaço. Na posição $Z = f$ localiza-se um plano chamado de plano da imagem, sendo na interseção deste plano com o feixe de luz, proveniente do ponto M no espaço, onde se forma a imagem do ponto como se apresenta na Figura 2.1.

A partir das relações geométricas (Figura 2.1) se determinam as coordenadas da imagem, em função das coordenadas do ponto.

$$x = f \frac{X}{Z} \quad y = f \frac{Y}{Z} \quad (2.4)$$

Obtido assim o mapeamento de um ponto no espaço 3D em um ponto no plano da imagem, da forma:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} f \frac{X}{Z} \\ f \frac{Y}{Z} \\ 1 \end{bmatrix} \quad (2.5)$$

O qual, escrito em coordenadas homogêneas, pode-se representar de forma matricial.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.6)$$

$$\tilde{\mathbf{x}} = \mathbf{P}\tilde{\mathbf{X}} \quad (2.7)$$

Onde \mathbf{P} é a matriz de projeção da câmera e $\tilde{\mathbf{x}}$ e $\tilde{\mathbf{X}}$ são as representações de \mathbf{x} e \mathbf{X} em coordenadas homogêneas.

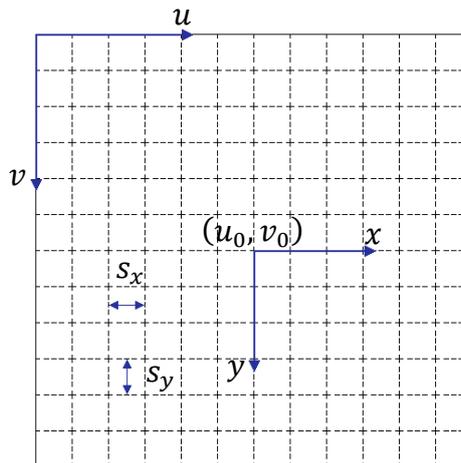


Figura 2.2. Sensor tipo CCD adaptado de (Comport, 2005)

As coordenadas da imagem (x, y) na equação (2.6) estão dadas com respeito ao *ponto principal* onde o eixo Z ou *eixo principal* atravessa o plano imagem. Para expressar a imagem do ponto em coordenadas de píxel (u, v) se faz o mapeamento:

$$\begin{aligned}
u &= s_x x + u_0 \\
v &= s_y y + v_0
\end{aligned}
\tag{2.8}$$

, onde s_x e s_y são os tamanhos dos píxeis nas direções x e y respectivamente (dados em píxeis/m), e u_0, v_0 são as coordenadas em píxeis do ponto principal (Figura 2.2).

A matriz de projeção de uma câmera, em coordenadas homogêneas, expressa-se como:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f s_x & 0 & u_0 & 0 \\ 0 & f s_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
\tag{2.9}$$

, onde a matriz

$$\mathbf{K} = \begin{bmatrix} f s_x & 0 & u_0 \\ 0 & f s_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}
\tag{2.10}$$

, é conhecida como a matriz de calibração da câmera; se os píxeis não forem quadrados incorpora-se à matriz o fator s_θ que representa a relação angular entre as duas direções do píxel, obtendo a matriz de calibração ou matriz de parâmetros intrínsecos \mathbf{K} da forma:

$$\mathbf{K} = \begin{bmatrix} f s_x & s_\theta & u_0 \\ 0 & f s_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}
\tag{2.11}$$

2.1.4- Homografia entre planos

Uma transformação projetiva entre planos, chamada Homografia, é uma transformação linear representada por uma matriz \mathbf{H} não singular de dimensões 3×3 :

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}
\tag{2.12}$$

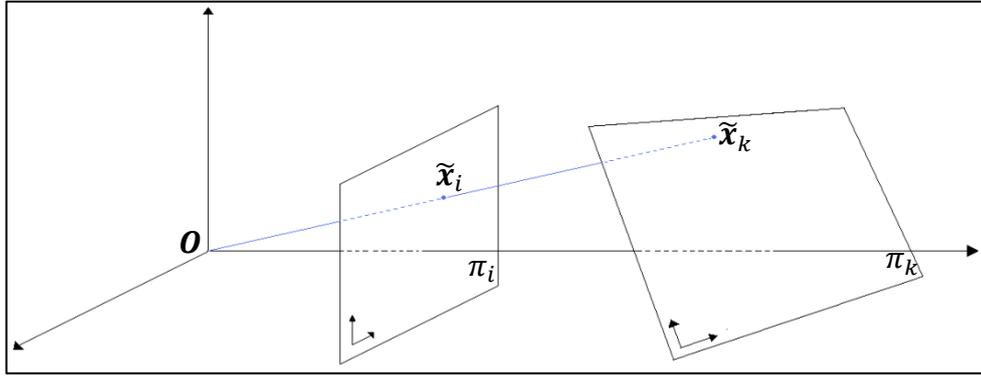


Figura 2.3. *Projeção de um ponto em dois planos a traves de um centro de projeção*

Um ponto $\tilde{\mathbf{x}}_k$, em coordenadas homogêneas, que pertence ao plano π_k relaciona-se linearmente com sua projeção (Figura 2.3) $\tilde{\mathbf{x}}_i$ pertencente ao plano π_i mediante uma homografia da forma:

$$\tilde{\mathbf{x}}_i = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tilde{\mathbf{x}}_k$$

, ou

$$\tilde{\mathbf{x}}_i = \mathbf{H} \tilde{\mathbf{x}}_k \quad (2.13)$$

, sendo possível relacionar linearmente pontos pertencentes a planos e suas imagens mediante sua homografia. Devido ao uso de coordenadas homogêneas, a igualdade da equação (2.13) só é conhecida sobre um fator de escala, pelo qual a matriz \mathbf{H} tem oito graus de liberdade (oito de suas nove entradas são independentes).

A homografia \mathbf{H} pode-se determinar a partir de correspondências de pontos $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}_k$ entre um plano e sua imagem ou entre duas imagens de um plano. Cada par destas correspondências gera duas equações linearmente independentes quando a equação (2.13) é reescrita da forma:

$$\tilde{\mathbf{x}}_i \times \mathbf{H} \tilde{\mathbf{x}}_k = 0 \quad (2.14)$$

Assim, quatro correspondências seriam necessárias para determinar completamente \mathbf{H} (Apêndice A), mas, devido à impossibilidade de ter dados completamente livres de erros, é geralmente necessário ter um maior número de correspondências e \mathbf{H} deve-se calcular

mediante um método de minimização (Hartley e Zisserman, 2003). Neste trabalho foi utilizado o método de minimização por *decomposição em valores singulares* (SVD) descrito no Apêndice B e finalmente utiliza-se uma estimação robusta mediante o método de RANSAC (seção 3.2.3.2-).

2.2- CALIBRAÇÃO DE CÂMERAS

A calibração de câmeras, no contexto da visão computacional, faz referência ao processo de determinar as características geométricas e ópticas da câmera (parâmetros intrínsecos) e (ou) a posição e orientação do sistema de referência da câmera relativo a um sistema de coordenadas externo (parâmetros extrínsecos) (Tsai, 1987). A calibração da câmera pode ser um processo complexo devido à quantidade de parâmetros envolvidos e aos fatores que influenciam os resultados. Mas existem vários métodos de calibração que fazem uso de modelos de câmeras ideais e conseguem, assim, uma redução considerável de parâmetros com resultados próximos à realidade, porém nunca exatos.

O objetivo do processo de calibração é a descrição do processo pelo qual uma câmera projeta um objeto 3D no plano da imagem e assim poder fazer uma estimativa métrica da cena, como por exemplo, a obtenção da posição no espaço de objetos dentro de campo visual ou a sua reconstrução tridimensional, entre outros.

Uma das abordagens clássicas para a calibração de câmeras baseia-se no uso de padrões de calibração (Tsai, 1987), (Zhang, 1998), (Hartley e Zisserman, 2003). Esta abordagem será vista a seguir, mas existem também outros métodos como o de autocalibração (Mendoça e Cipolla, 1999), (Hemayed, 2003) ou métodos iterativos, que não estão dentro do escopo de estudo de este trabalho.

2.2.1- Calibração com padrão plano

O método aqui descrito é o Método de Zhang (1998), o qual utiliza duas ou mais vistas diferentes de um padrão plano de pontos e faz uso do modelo de câmera *Pinhole* visto na seção 2.1.3. Da equação (2.7), e tendo em conta a natureza planar do objeto obtém-se a relação

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \quad (2.15)$$

Onde \mathbf{P} é a matriz de projeção da câmera (tendo em conta o fator de escala). Esta transformação, em função da rotação e translação, pode ser escrita como:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3 \ \mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \quad (2.16)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2.17)$$

, onde \mathbf{K} é matriz de parâmetros intrínsecos da câmera (equação (2.11)) e $[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3 \ \mathbf{t}]$ é a matriz que representa a posição da câmera no espaço, também chamada matriz de parâmetros extrínsecos.

A relação entre pontos em um plano e sua imagem pode ser descrito em função da matriz de Homografia \mathbf{H} (Hartley e Zisserman, 2003), (seção 2.1.4), obtendo assim que, o ponto $\tilde{\mathbf{M}}$ se relaciona com sua imagem $\tilde{\mathbf{m}}$, mediante:

$$\tilde{\mathbf{m}} = \mathbf{H}\tilde{\mathbf{M}} \quad (2.18)$$

, com

$$\mathbf{H} = \mathbf{K}[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}] = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3] \quad (2.19)$$

Da expressão anterior:

$$\begin{aligned} \mathbf{h}_1^T (\mathbf{K}^{-1})^T \mathbf{K}^{-1} \mathbf{h}_2 &= 0 \\ \mathbf{h}_1^T (\mathbf{K}^{-1})^T \mathbf{K}^{-1} \mathbf{h}_1 &= \mathbf{h}_2^T (\mathbf{K}^{-1})^T \mathbf{K}^{-1} \mathbf{h}_2 \end{aligned} \quad (2.20)$$

, fazendo $(\mathbf{K}^{-1})^T \mathbf{K}^{-1} = \mathbf{B}$ e usando (2.11)

$$\mathbf{B} = \begin{bmatrix} \frac{1}{(fs_x)^2} & -\frac{s_\theta}{(fs_x)^2(fs_y)} & \frac{u_0 s_\theta - u_0(fs_y)}{(fs_x)^2(fs_y)} \\ -\frac{s_\theta}{(fs_x)^2(fs_y)} & \frac{s_\theta^2}{(fs_x)^2(fs_y)^2} + \frac{1}{(fs_y)^2} & -\frac{s_\theta(v_0 s_\theta - u_0(fs_y))}{(fs_x)^2(fs_y)^2} - \frac{v_0}{(fs_y)^2} \\ \frac{v_0 s_\theta - u_0(fs_y)}{(fs_x)^2(fs_y)} & -\frac{s_\theta(v_0 s_\theta - u_0(fs_y))}{(fs_x)^2(fs_y)^2} - \frac{v_0}{(fs_y)^2} & \frac{(v_0 s_\theta - u_0(fs_y))^2}{(fs_x)^2(fs_y)^2} + \frac{v_0^2}{(fs_y)^2} + 1 \end{bmatrix} \quad (2.21)$$

$$\mathbf{h}_1^T \mathbf{B} \mathbf{h}_2 = 0 \quad (2.22)$$

$$\mathbf{h}_1^T \mathbf{B} \mathbf{h}_1 - \mathbf{h}_2^T \mathbf{B} \mathbf{h}_2 = 0$$

, ou:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = 0 \quad (2.23)$$

, onde $\mathbf{b} = [B_{11} B_{12} B_{22} B_{13} B_{23} B_{33}]$, e \mathbf{v}_{ij} é um vetor de combinações lineares de h_{12} . Se forem adquiridas n imagens, e empilhando n equações como (2.20) se obtém:

$$\mathbf{V} \mathbf{b} = 0 \quad (2.24)$$

, onde \mathbf{V} é uma matriz de $2n \times 6$. Se $n \geq 3$, se pode obter uma única solução \mathbf{b} dos parâmetros intrínsecos da câmera com uma decomposição de valores singulares; uma estimativa inicial dos parâmetros extrínsecos se calcula utilizando (2.16). Para encontrar uma solução robusta se implementa um método iterativo de minimização do erro e se inclui na solução uma estimativa de distorção radial (Zhang, 1998) (Bouguet, 2009).

2.3- CONTROLE SERVO VISUAL

O princípio do controle servo visual baseia-se no uso da informação visual fornecida por uma ou mais câmeras dentro da malha de controle de movimento de um robô (Kim, et al., 1995) (Chaumette e Hutchinson, 2006); os primeiros sistemas de controle baseados em visão, reportados nos finais dos anos 1960 (McCarthy, et al., 1968), caracterizavam-se pelo fato de que as tarefas de interpretação da informação visual e de movimento do robô estavam totalmente separadas em forma de malha aberta, mas essas abordagens,

conhecidas como “ver e mover estático” (“*static look and move*”), embora sua simplicidade, sofrem de alta sensibilidade a erros de calibração da câmera e do robô e limitam-se a tarefas bem conhecidas. Já em Shirai e Inoue (1973), integra-se pela primeira vez a informação visual dentro da malha de realimentação do controle do robô, o que diminuiu significativamente o efeito destes erros. Mas é só no ano de 1979 que Hill e Park (1979) introduzem o termo de controle servo visual (“*visual servoing*”) para descrever o controle com realimentação visual em malha fechada (Figura 2.4). Sanderson e Weiss (1980) propuseram a primeira taxonomia de classificação dos tipos de sistemas de controle visual baseada na natureza da informação fornecida ao sistema de controle para realizar a tarefa de movimentação do robô, que se tornou um dos métodos clássicos de classificação do controle visual.

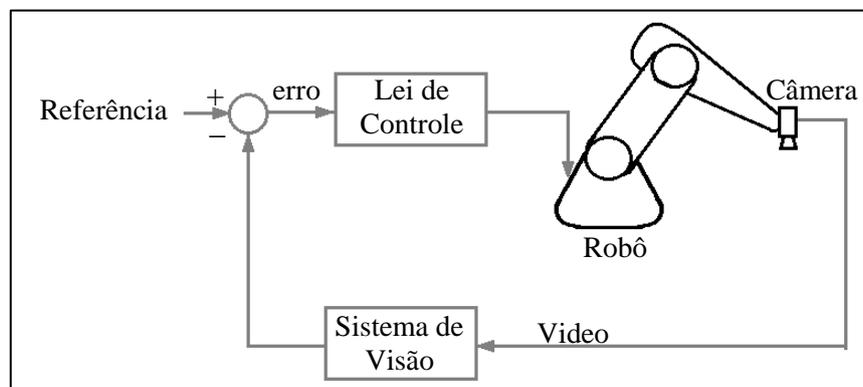


Figura 2.4. Controle Servo Visual

2.3.1- Controle Servo Visual Baseado em Imagem

Nos sistemas de controle baseados em imagem (“Image Based Visual Servoing” IBVS), conhecidos também como controle servo visual 2D, a definição do erro é feita a partir de valores calculados diretamente no espaço bidimensional das características da imagem (Figura 2.5). Pode-se definir estas últimas como qualquer elemento estrutural da imagem (canto, borda, etc.) que tipicamente corresponda a uma característica física de algum objeto dentro do campo visual da câmera (Hutchinson, et al., 1996).

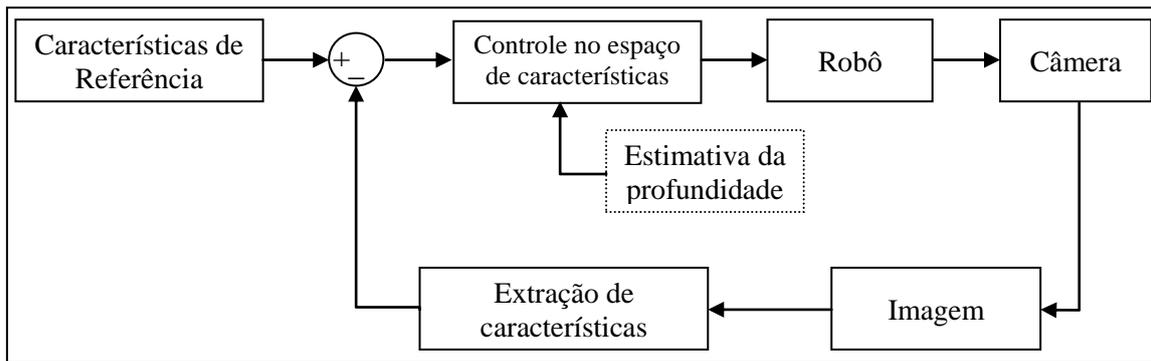


Figura 2.5 Diagrama de Blocos do IBVS

Geralmente, no IBVS, a tarefa de posicionamento é formulada como uma função do regulamento do erro usando uma abordagem do tipo “aprendizado por demonstração” (*teaching by showing*) (Malis, 2004), na qual é obtida uma imagem de referência na posição alvo e, posteriormente, durante o processo de controle, a cada ciclo, a imagem atual é comparada com a imagem de referência, de modo que o alvo seja atingido no instante em que as características da imagem coincidem com as características de referência.

Uma das etapas iniciais desta abordagem é a escolha das características (Shiroma, 2004), e os cálculos dos parâmetros associados a elas que dependem da aplicação para a qual é projetado o sistema. Podem-se utilizar diferentes tipos de informação fornecida pela imagem. A maior parte de trabalhos apresentados de controle servo visual 2D baseia-se no uso de coordenadas de pontos na imagem (Feddema, et al., 1989), (Espiau, 1994), (Snith e Papanikolopoulos, 1995), (Iwatsuki e Okiyama, 2005), mas podem-se utilizar outros tipos de primitivas geométricas; Espiau et al. (Espiau, et al., 1992) e Chaumette (Chaumette, 1990), por exemplo, fazem uma análise do controle servo visual com pontos, linhas, círculos, esferas e elipses. Já para cenas mais complexas, ou visando estabilizar o controle, podem ser utilizados outros tipos de características, como momentos (Liu, et al., 2009), descritores de Fourier ou invariantes projetivas (Hager, 1995). Em geral a natureza e representação dos parâmetros escolhidos dentro da imagem desempenham um papel importante dentro do comportamento do sistema e, a partir da sua definição, se faz possível o planejamento da lei de controle.

2.3.2- Controle Servo Visual Baseado em Posição

O controle servo visual baseado em posição (“Position Based Visual Servoing” PBVS) ou controle servo visual 3D, baseia-se na extração de características da imagem para calcular a postura (posição e orientação) do objeto alvo em relação à câmera. Desta forma, a função de erro é definida como a diferença entre a postura atual e a postura desejada no espaço de trabalho do robô (Figura 2.6).

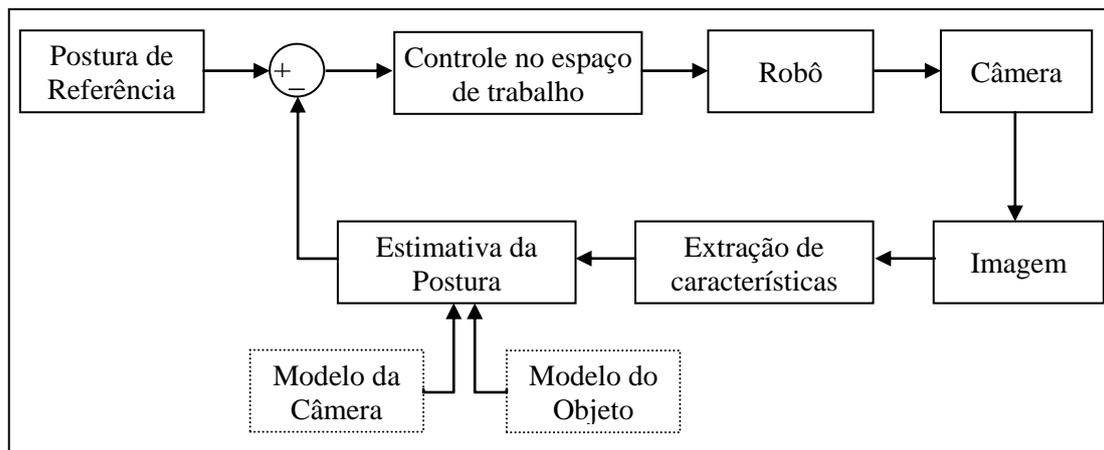


Figura 2.6. Diagrama de Blocos do PBVS

Um dos principais aspectos neste tipo de controle é a estimativa dos parâmetros empregados na definição do erro, ou seja, a reconstrução da postura de um ou mais objetos na cena, o que está fortemente relacionado com o problema da recuperação da geometria de objetos a partir de uma ou mais imagens. Geralmente este tipo de aproximação exige o conhecimento prévio da geometria do objeto alvo e das características internas da câmera. Dementhon e Davis (1995) descrevem uma metodologia para recuperar a postura de um objeto a partir de uma imagem e do modelo do objeto. Martinet, et al., (1996) e Martinet, et al., (1997) fazem uso dessa metodologia para planejar um controle visual para um robô móvel. Em Wilson, et al., (1996), além de um conhecimento prévio do objeto utilizam o filtro de Kalman Estendido para encontrar a postura do alvo a cada ciclo do controle. Outras abordagens, no entanto, não necessitam de informação da geometria do objeto e utilizam sistemas de visão estéreo para fazer a reconstrução (Hager, 1997) (Kragic, 2001).

A eficiência do controle PBVS está diretamente relacionada com a exatidão dos modelos empregados e depende em grande medida dos resultados da etapa de reconstrução.

Contudo, uma de suas principais vantagens é a capacidade de se ter um sistema de controle mais intuitivo, já que está em função das coordenadas cartesianas (Wilson, et al., 1996).

2.3.3- Definição da Lei de Controle

Os controles de tipo proporcional, que garantem o decréscimo do erro de forma exponencial, têm demonstrado seu bom funcionamento nas aplicações de controle nos sistemas robóticos com sensores visuais (Chaumette, 1990), (Espiau, 1994), (Comport, et al., 2003). Assim, então, a lei de controle num sistema de controle servo visual pode-se expressar como:

$$\dot{e} = -\lambda e \quad (2.25)$$

, onde λ é um escalar positivo.

Trabalhos apresentados no início dos anos 1990 (Chaumette, 1990) mostram que a lei de controle pode ser projetada na forma de regulação em zero de uma *função de tarefa* $e(\mathbf{q}, t)$ que representa o erro entre a configuração desejada e a configuração atual dos valores das variáveis de juntas do robô \mathbf{q} no instante t . Esta função de tarefa pode ter diferentes formas, dependendo do espaço de aplicação. Alguns exemplos são:

- $e(\mathbf{q}, t) = \mathbf{q} - \mathbf{q}^*(t)$: Onde $\mathbf{q}^*(t)$ é uma trajetória desejada no espaço das juntas.
- $e(\mathbf{q}, t) = \mathbf{r} - \mathbf{r}^*(t)$: Onde $\mathbf{r}^*(t)$ representa uma trajetória desejada no espaço operacional do manipulador, o espaço cartesiano.
- $e(\mathbf{q}, t) = \mathbf{s}(t) - \mathbf{s}^*$, onde $\mathbf{s}(t)$ define o vetor de características atual e \mathbf{s}^* é o vetor de características de referência.

Para uma câmera embarcada no manipulador robótico, o ponto de visão e, portanto, as características, estão em função da postura do robô \mathbf{r} . Assim, a função de tarefa baseada em características visuais é uma função da forma:

$$\mathbf{e} = (\mathbf{s}(\mathbf{r}, t) - \mathbf{s}^*) \quad (2.26)$$

2.3.4- Lei de Controle no Controle Servo Visual Baseado em Imagem (IBVS)

Como visto anteriormente, a principal diferença entre IBVS e PBVS está na definição do espaço da função de tarefa, \mathbf{s} , que para o caso do controle servo visual baseado em imagem é um vetor em coordenadas da imagem (seção 2.3.1). No que se refere a este trabalho as características relevantes de estudo são pontos descritos pelas suas coordenadas em píxeis dentro da imagem, faz-se necessário, então, encontrar a relação entre o movimento da câmera e a variação das características. Esta relação não é linear, já que movimentos na câmera resultam em movimentos complexos das características, mas pode-se linearizar (Corke, 1993) da seguinte forma:

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial t} = \mathbf{L}_s \dot{\mathbf{r}} \quad (2.27)$$

, onde

$$\mathbf{L}_s = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} = \begin{bmatrix} \frac{\partial s_1}{\partial r_1} & \dots & \frac{\partial s_1}{\partial r_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_k}{\partial r_1} & \dots & \frac{\partial s_k}{\partial r_m} \end{bmatrix} \quad (2.28)$$

\mathbf{L}_s é denominada a *Matriz de Interação da Imagem* ou o *Jacobiano da Imagem* e tem dimensões $[k \times m]$ sendo m a dimensão do espaço operacional do robô e k a dimensão do espaço das características. Se a matriz de interação é conhecida, e se existe sua inversa \mathbf{L}_s^{-1} , poder-se-ia calcular a velocidade do elemento terminal do manipulador $\dot{\mathbf{r}}$ a partir da equação (2.27), da forma:

$$\dot{\mathbf{r}} = \mathbf{L}_s^{-1} \dot{\mathbf{s}} \quad (2.29)$$

Derivando a equação (2.26)

$$\dot{\mathbf{e}} = \frac{d}{dt} (\mathbf{s}(\mathbf{r}, t) - \mathbf{s}^*) \quad (2.30)$$

, e com \mathbf{s}^* constante,

$$\dot{\mathbf{e}} = \frac{d}{dt} (\mathbf{s}(\mathbf{r}, t)) \quad (2.31)$$

Ou, da equação (2.27)

$$\dot{\mathbf{e}} = \mathbf{L}_s \dot{\mathbf{r}} \quad (2.32)$$

Igualando as equações (2.25) e (2.32), têm-se:

$$\mathbf{L}_s \dot{\mathbf{r}} = -\lambda \mathbf{e} \quad (2.33)$$

$$\dot{\mathbf{r}} = -\lambda \mathbf{L}_s^{-1} \mathbf{e} \quad (2.34)$$

, encontrando assim uma lei de controle em função da velocidade do manipulador $\dot{\mathbf{r}}$ e em função do erro $\mathbf{e} = (\mathbf{s}(\mathbf{r}, t) - \mathbf{s}^*)$.

Não entanto há situações onde é impossível calcular a matriz \mathbf{L}_s^{-1} , como quando $k \neq m$ ou quando \mathbf{L}_s não têm posto (*rank*) cheio, então faz-se uso da matriz pseudo-inversa *Moore-Penrose* \mathbf{L}_s^+ (Chaumette e Hutchinson, 2006), dada pela relação:

$$\mathbf{L}_s^+ = (\mathbf{L}_s^T \mathbf{L}_s)^{-1} \mathbf{L}_s^T \quad (2.35)$$

Em sistemas de controle visual real, é impossível encontrar o valor exato de \mathbf{L}_s ou de \mathbf{L}_s^+ , e portanto é feita uma aproximação destes valores resultando em uma lei de controle da forma:

$$\dot{\mathbf{r}} = -\lambda \widehat{\mathbf{L}}_s^+ \mathbf{e} \quad (2.36)$$

, com $\widehat{\mathbf{L}}_s^+$ sendo o valor aproximado da pseudo-inversas da *matriz de interação da imagem* \mathbf{L}_s .

2.3.4.1- Matriz de Interação \mathbf{L}_s de Pontos

Com o fim de implementar a lei de controle descrita em (2.36) é necessário definir a matriz de interação \mathbf{L}_s , que depende do tipo de primitivas estudadas. Neste caso as primitivas utilizadas são pontos, e como visto nas equações (2.27) e (2.28), a matriz de interação relaciona o movimento da câmera em função de um referencial global com as variações

das características no referencial da imagem. Assim, para descrevê-la, é feita uma análise da variação das posições no tempo tanto da câmera como dos pontos observados por ela.

Tendo em conta o modelo de câmera *pinhole* descrito na seção 2.1.3, um ponto \mathbf{P} de coordenadas espaciais (X, Y, Z) se projeta em uma imagem no ponto \mathbf{p} de coordenadas (x, y) :

$$x = fX/Z \quad y = fY/Z \quad (2.37)$$

, onde os parâmetros intrínsecos da câmera K são assumidos como constantes durante toda a tarefa de controle. As relações entre o sistema de coordenadas em píxeis da imagem (u, v) e as coordenadas espaciais (x, y) são:

$$\begin{aligned} x &= \left(\frac{u - u_0}{s_x} \right) \\ y &= \left(\frac{v - v_0}{s_y} \right) \end{aligned} \quad (2.38)$$

A velocidade da câmera \mathbf{v}_c em relação a um referencial externo esta dado por :

$$\begin{aligned} \dot{X} &= -T_x - \omega_y Z + \omega_z Y \\ \dot{Y} &= -T_y - \omega_z X + \omega_x Z \\ \dot{Z} &= -T_z - \omega_x Y + \omega_y X \end{aligned} \quad (2.39)$$

, onde $T(t) = [T_x(t), T_y(t), T_z(t)]^T$ é o componente translacional da velocidade e $\Omega(t) = [\omega_x(t), \omega_y(t), \omega_z(t)]$ o componente angular.

Para obter a variação das coordenadas na imagem a equação (2.33) é derivada em função do tempo:

$$\begin{aligned} \dot{x} &= \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{(\dot{X} - x\dot{Z})}{Z} \\ \dot{y} &= \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{(\dot{Y} - y\dot{Z})}{Z} \end{aligned} \quad (2.40)$$

, e substituindo (2.36) em (2.37),

$$\begin{aligned}\dot{x} &= -\frac{T_x}{Z} + \frac{xT_z}{Z} + xy\omega_x - (1+x^2)\omega_y + y\omega_z \\ \dot{y} &= -\frac{T_y}{Z} + \frac{yT_z}{Z} + (1+y^2)\omega_x - xy\omega_y - x\omega_z\end{aligned}\tag{2.41}$$

, encontra-se a velocidade dos pontos na imagem, em função da velocidade da câmera.

Assumindo como o vetor de características \mathbf{s} um conjunto de pontos \mathbf{p} na imagem, de coordenadas (x, y) , e de acordo à equação (2.27), a relação das velocidades da câmera e dos pontos na imagem pode-se escrever de forma matricial como:

$$\dot{\mathbf{p}} = \mathbf{L}_s \mathbf{v}_c\tag{2.42}$$

, em que \mathbf{L}_s representa a matriz de interação de cada ponto característico e $\mathbf{v}_c = [\dot{X}, \dot{Y}, \dot{Z}]$ é a velocidade da câmera.

Desta forma, como se deduz das equações (2.41) e (2.42), a matriz de interação para pontos numa imagem define-se de forma explícita como:

$$\mathbf{L}_s = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & (1+y^2) & -xy & -x \end{bmatrix}\tag{2.43}$$

Esta matriz está composta das coordenadas em píxeis de cada ponto, do seu valor de profundidade Z em referência à câmera e dos parâmetros intrínsecos da câmera. Estes dois últimos valores são geralmente estimados a partir de aproximações, portanto o valor da matriz de interação é também uma estimativa $\hat{\mathbf{L}}_s$ do valor real. Para sua aplicação no controle de um sistema de seis graus de liberdade a matriz deve ter no mínimo seis linhas, ou seja, três pontos, mas na prática geralmente escolhem-se mais pontos com a finalidade de eliminar singularidades e mínimos globais não desejados (Chaumette e Hutchinson, 2006).

2.3.4.2- Estimativa de $\widehat{\mathbf{L}}_s$

A estimativa de $\widehat{\mathbf{L}}_s$ desempenha um papel importante na estabilidade do controle e por tal razão tem-se proposto em trabalhos anteriores diferentes métodos de escolha. Marey e Chaumette (2008) apresentam os mais relevantes:

- $\widehat{\mathbf{L}}_s = \mathbf{L}_{s^*}$: Neste caso considera-se $\widehat{\mathbf{L}}_s$ constante durante a tarefa de controle e o seu valor calcula-se a partir da matriz de interação na configuração desejada. A estabilidade neste tipo de abordagem só é garantida na vizinhança da posição desejada e a trajetória do sistema é pouco restrita e pode apresentar-se o caso de perda de pontos no campo visual.
- $\widehat{\mathbf{L}}_s = \mathbf{L}_{s(t)}$: neste caso o valor de \mathbf{L}_s é estimado a cada iteração em função dos parâmetros atuais da configuração da câmera e a estimativa da profundidade, que pode ser conhecida a partir do modelo do objeto alvo e a calibração da câmera. No entanto, o cálculo da profundidade pode levar um tempo considerável com um número grande de primitivas. Por esta razão, às vezes é conveniente dar um valor aproximado de Z , que geralmente se aproxima ao valor na posição desejada
- $\widehat{\mathbf{L}}_s = (\mathbf{L}_{s^*} + \mathbf{L}_{s(t)})/2$: neste caso, considera-se o valor médio entre os dois casos anteriores. O valor da profundidade Z deve ser conhecido, uma vez que $\mathbf{L}_{s(t)}$ está envolvido.

Neste trabalho o valor da matriz de interação foi calculado a cada iteração, contando com um valor aproximado da profundidade.

2.3.5- Lei de Controle no Controle Servo Visual Baseado em Posição (PBVS)

No controle visual baseado em posição, o comando se faz diretamente sobre a postura da câmera, procurando reduzir a diferença entre sua posição atual e sua posição desejada. Isto implica que a definição do vetor de características \mathbf{s} seja em função da postura da câmera com respeito de um referencial.

Para definir corretamente este erro consideram-se os três referenciais apresentados na Figura 2.7, onde \mathcal{F}_{c^*} representa o referencial associado à câmera em sua posição desejada, \mathcal{F}_c é o referencial da câmera em sua posição atual e \mathcal{F}_o é o referencial do objeto, as transformações entre os respectivos origens dos referenciais estão dados por cT_o e ${}^{c^*}T_o$.

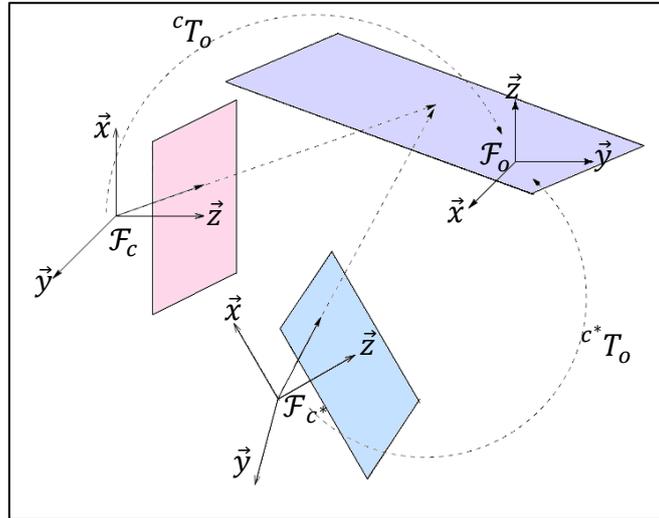


Figura 2.7. Modelo do movimento da câmera em PBVS, adaptado de (Malis, et al., 1998)

Para que o referencial da câmera passe à posição desejada teria que experimentar uma transformação ${}^{c^*}T_c = [{}^{c^*}\mathbf{R}_c | {}^{c^*}\mathbf{t}_c]$, que poderia encontrar-se a partir do conhecimento da postura do objeto nas duas posições da câmera (${}^{c^*}T_o$ e cT_o).

Com o conhecimento da postura relativa das câmeras e sua decomposição, pode-se obter uma lei de controle em função da velocidade da câmera, como será visto nos seguintes parágrafos.

Neste trabalho, utilizou-se o fato de que qualquer matriz de rotação \mathbf{R} pode ser tomada como uma rotação de um ângulo θ ao redor de um eixo fixo \mathbf{u} (Ma, et al., 2001), tal que:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.44)$$

, e

$$\theta = \cos^{-1} \left(\frac{\text{traço}(R) - 1}{2} \right) \quad \mathbf{u} = \frac{1}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \quad (2.45)$$

Desta forma o vetor de características, \mathbf{s} , poder-se-ia definir como:

$$\mathbf{s} = f(\mathbf{t}, \mathbf{u}\theta) \quad (2.46)$$

, onde \mathbf{t} representa a translação e $\mathbf{u}\theta$ a rotação entre os referenciais das câmeras, ou seja, o vetor de características esta em função da postura das câmeras em sua posição atual e a posição desejada.

Seguindo o mesmo esquema de desenvolvimento do controle IBVS, uma relação entre o movimento da câmera e a mudança da sua posição está dada pela equação (2.27)

$$\dot{\mathbf{s}} = \mathbf{L}_s \dot{\mathbf{r}}$$

Segundo Chaumette e Hutchinson (2006) \mathbf{L}_s depende da escolha do referencial no qual se defina a translação, neste caso escolheu-se $\mathbf{t} = {}^{c^*}\mathbf{t}_c$, desta forma

$$\mathbf{s}^* = 0, \quad \mathbf{s} = ({}^{c^*}\mathbf{t}_c, \mathbf{u}\theta) \quad (2.47)$$

$$\mathbf{e} = \mathbf{s}, \quad \dot{\mathbf{e}} = \dot{\mathbf{s}} \quad (2.48)$$

A equação (2.27)

$$\dot{\mathbf{r}} = \mathbf{L}_s^{-1} \dot{\mathbf{e}} \quad (2.49)$$

, faz a relação entre a variação das características e a velocidade da câmera $\dot{\mathbf{r}} = \mathbf{v}_c$.

Aplicando uma lei de controle proporcional, tal que $\dot{\mathbf{e}} = -\lambda \mathbf{e}$, se obtém

$$\mathbf{v}_c = -\lambda \mathbf{L}_s^{-1} ({}^{c^*}\mathbf{t}_c, \theta \mathbf{u}) \quad (2.50)$$

\mathbf{L}_s define-se como (Chaumette e Hutchinson, 2006):

$$\mathbf{L}_s = \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{L}_{\theta \mathbf{u}} \end{bmatrix} \quad (2.51)$$

, com a característica que $\mathbf{L}_{\theta\mathbf{u}}^{-1}(\theta\mathbf{u}) = \theta\mathbf{u}$, desta forma obtém-se a seguinte lei de controle:

$$\mathbf{v}_c = \begin{bmatrix} T \\ \Omega \end{bmatrix} = \begin{bmatrix} -\lambda \mathbf{R}^{T c*} \mathbf{t}_c \\ -\lambda \theta\mathbf{u} \end{bmatrix} \quad (2.52)$$

Todas as variáveis aqui envolvidas são obtidas das relações de postura entre a câmera em sua posição atual e a câmera em sua posição desejada.

3 - CONTROLE SERVO VISUAL

Neste trabalho se apresentam duas metodologias de controle servo visual para manipuladores robóticos, o controle servo visual baseado em imagem (Figura 2.5) e o controle servo visual baseado em posição (Figura 2.6), sob uma configuração *eye in hand* (câmera embarcada) para um manipulador robótico de seis graus de liberdade. Estas abordagens baseiam-se principalmente em técnicas de processamento digital de imagens para a extração das características relevantes na cena e para sua interpretação, a partir dos quais é projetado o sistema de controle que utiliza a técnica de “aprendizagem por demonstração” (ver seção 2.3.1).

O sistema consiste de duas etapas, uma etapa *off-line*, na qual se extrai toda a informação das configurações invariáveis do sistema como o modelo do objeto, os parâmetros de calibração da câmera e a imagem do alvo na posição desejada, e uma etapa *on-line*, na qual se faz o processamento da imagem e a execução do controle.

Nas Figuras 3.3 e 3.4 apresentam-se os esquemas das abordagens de controle propostas, os quais serão descritos passo a passo durante este capítulo, descrevendo as metodologias e algoritmos empregados no seu desenvolvimento. Posteriormente será feita uma referência às etapas de modelagem cinemática do manipulador.

3.1- EQUIPAMENTO

Para contar com uma maior flexibilidade nos algoritmos desenvolvidos, as imagens foram adquiridas com dois tipos de câmera: uma câmera *web*, de fabricante *Microsof*® e uma câmera *CCD Monochrome Pulnix TM 1020-15 CL*, com as seguintes características, fornecidas pelo fabricante:

- ✓ Resolução Espacial Máxima 1008 (H) x 1018 (V) a 15 fps
- ✓ Píxeis quadrados de 9,0µm
- ✓ Saída Digital *Câmera Link*
- ✓ 30 fps
- ✓ Placa de aquisição NI PCI 1428

Todos os programas de processamento de imagens e de modelagem cinemático foram totalmente desenvolvidos no ambiente Matlab ® e Simulink ®. Para a etapa de calibração da câmera foi utilizado o toolbox de calibração de câmeras fornecido por Bouguet (2009) o qual foi ligeiramente modificado para ser utilizado com um padrão de calibração de pontos como o apresentado na Figura 3.1.

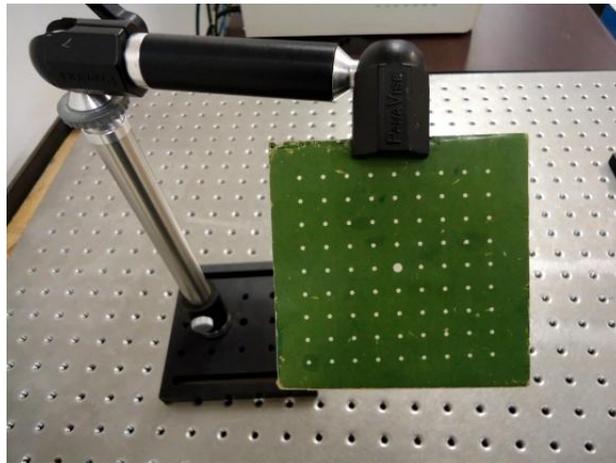


Figura 3.1. Padrão de calibração

Neste projeto, optou-se por objetos alvos de superfícies planas (Figura 3.2) ou bidimensionais que facilitaram em grande medida a aplicação das técnicas de análise de imagens já que mantêm uma relação homográfica entre suas imagens. Além disso, este tipo de objetos podem ser encontrados em cenas reais nas quais poder-se-ia estender esta aplicação e, por essa razão, não há perda de generalidade na aplicação. Outra característica da montagem experimental foi que as condições de luz não fossem controladas, mas evitando-se variações bruscas entre duas cenas consecutivas, o que se garante em condições normais de iluminação.



Figura 3.2. Objetos de Superfície Plana

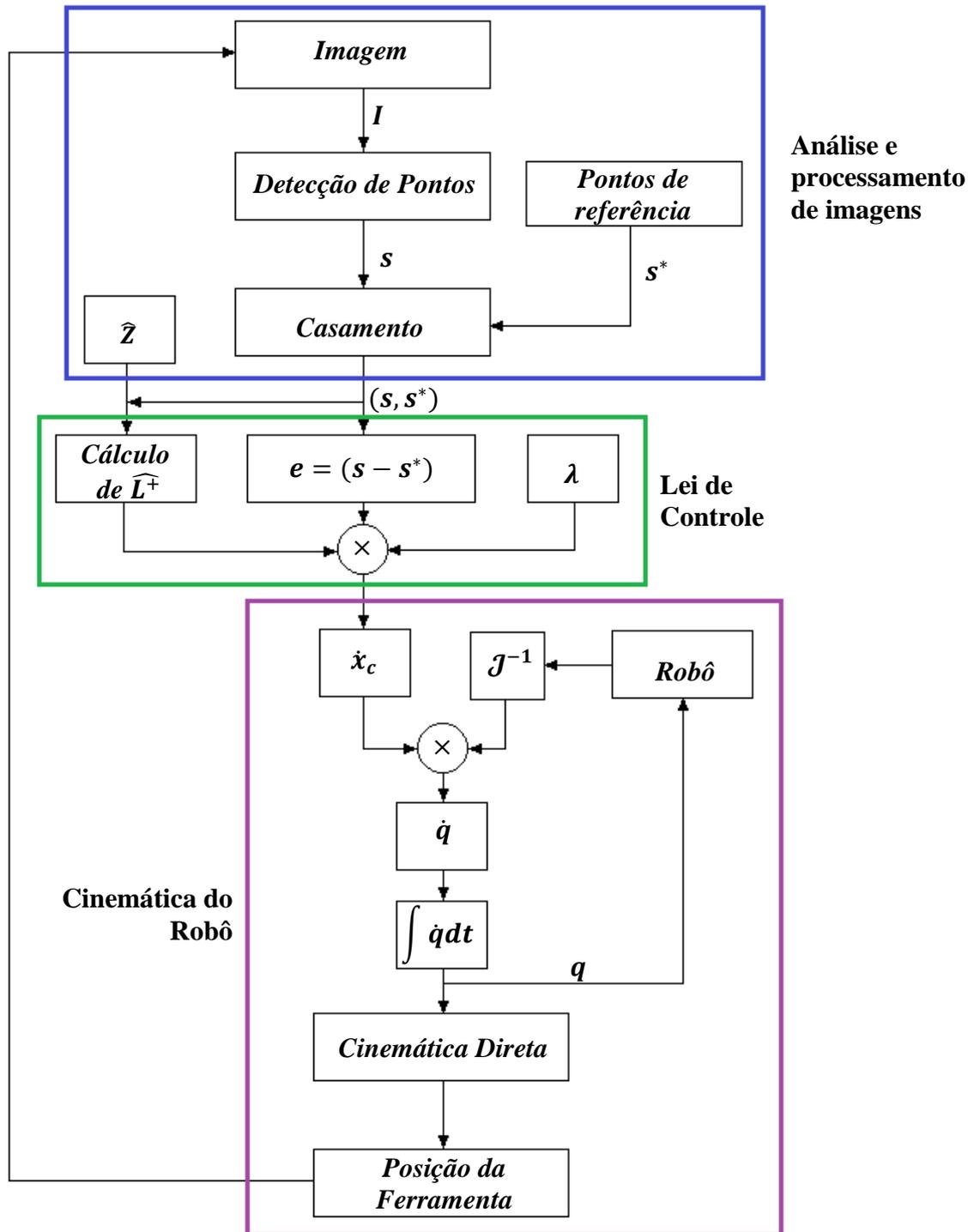


Figura 3.3. Esquema do Controle Servo Visual Baseado em Imagem

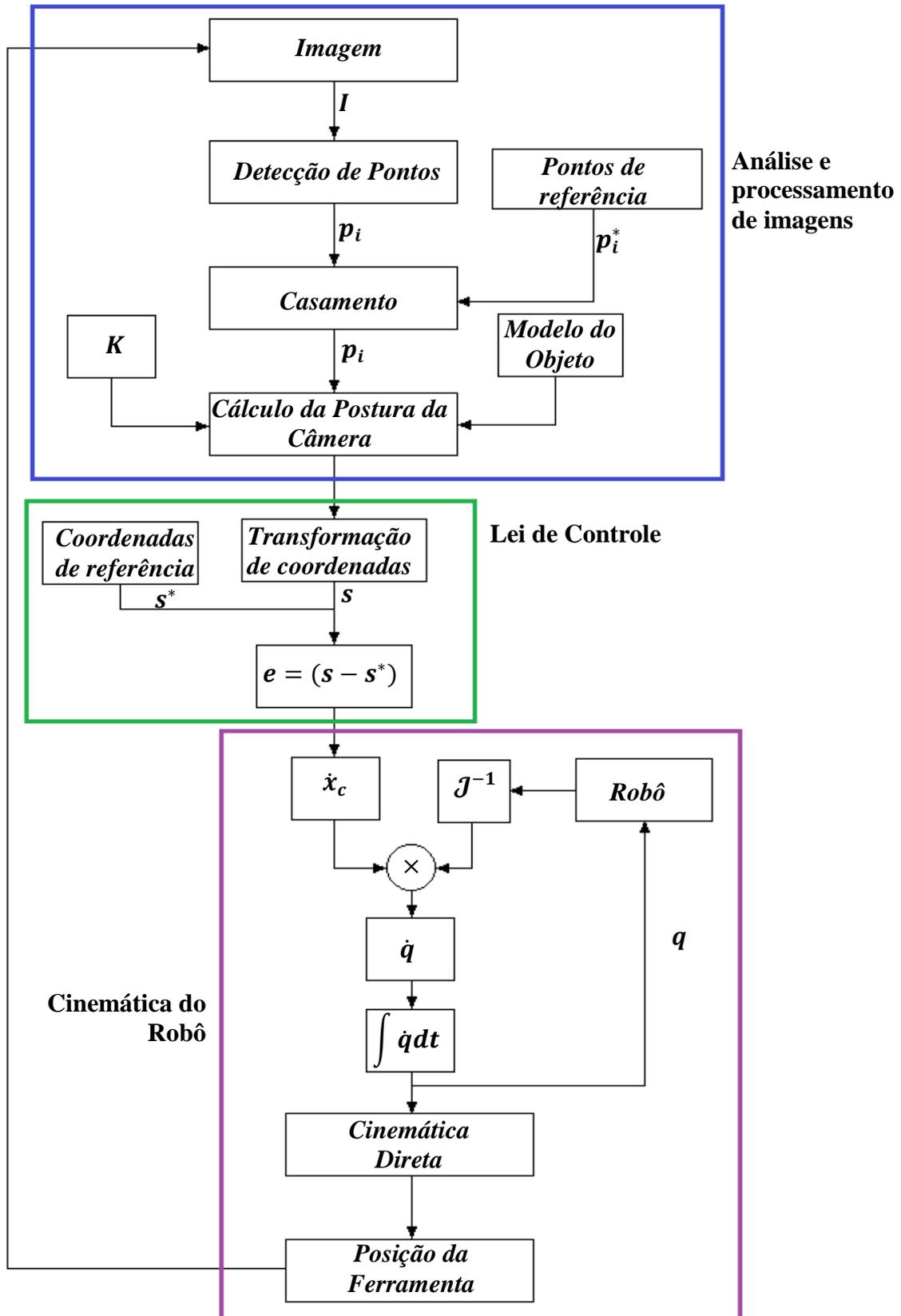


Figura 3.4. Esquema de Controle Servo Visual Baseado em Posição

3.2- PROCESSAMENTO DE IMAGENS

Um dos principais objetivos das técnicas de processamento de imagens é a extração de informação relevante de uma cena a partir de sua imagem e, em alguns casos, da informação da configuração do sistema visual (câmera-cena) para sua posterior interpretação. O termo “relevante” faz referência à importância das informações que dependem exclusivamente da aplicação.

No caso particular deste trabalho, têm-se dois problemas específicos:

- No primeiro, aplicado para o controle IBVS, devem-se comparar as características da imagem de um alvo numa posição r_i com as características da imagem do mesmo alvo numa posição r_0 .
- No segundo caso, aplicado no controle PBVS, a comparação se faz diretamente entre as duas posições do alvo, r_i e r_0 , extraídas a partir de suas imagens.

3.2.1- Extração de características

Apesar da natureza da informação das duas abordagens propostas ser diferente, a base para sua obtenção é a mesma, ou seja, a partir de duas imagens de um alvo extrair características tais que permitam, mediante sua comparação, encontrar a variação da posição do objeto com respeito à câmera. Para isso foi desenvolvida uma etapa de extração de pontos de imagens, consistente na aplicação do algoritmo de Harris (Harris e Stephens, 1988) sobre imagens de objetos planos (Figura 3.5).

O algoritmo de Harris baseia-se no princípio da existência de grande variação da intensidade de píxeis nos cantos em todas as direções e faz o cálculo da *Matriz de Co-variância* da imagem a partir dos valores do gradiente da intensidade da imagem. Posteriormente analisa a mudança nos valores de intensidade ao redor do ponto mediante o deslocamento de uma janela gaussiana.



Figura 3.5. Aplicação do detector de cantos de Harris

A seguir, apresenta-se o algoritmo de Harris adaptado para este trabalho (Algoritmo 1). Os valores de gradiente são obtidos a partir da técnica de diferenciação direcional de sinais discretos proposta por Farid e Simoncelli (2004). A matriz de co-variância M da imagem I , define-se como:

$$M = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \quad (3.1)$$

e a função de resposta do detector R :

$$R = [\det(M) - k * \text{traço}(M)^2] \quad (3.2)$$

Objetivo

Dada uma imagem I, e os seguintes parâmetros:

- n : Tamanho do filtro Gaussiano
- σ : Desvio Padrão do Filtro Gaussiano
- w : Tamanho da Janela onde só pode haver um ponto (canto)

Limiar: Mínimo valor de auto-correlação que um ponto deve ter para se considerar um canto, determinar as posições em I dos cantos.

Algoritmo

- i. Calcular $(\partial I/\partial x)^2$, $(\partial I/\partial y)^2$ e $(\partial I/\partial x)(\partial I/\partial y)$ (Farid e Simoncelli, 2004)
- ii. Aplicar filtro gaussiano $g(n, \sigma)$ aos valores calculados em i.
- iii. Com os valores obtidos de ii calcular a matriz M usando a equação (3.1)
- iv. Calcular R, com a equação (3.2)
- v. Escolher os pontos que cumprem:

$$\max(R, [w]) \wedge R \geq \text{limiar}$$

- vi. Retornar as posições do conjunto de pontos escolhidos em v.

Algoritmo 1. Detecção de Cantos de Harris

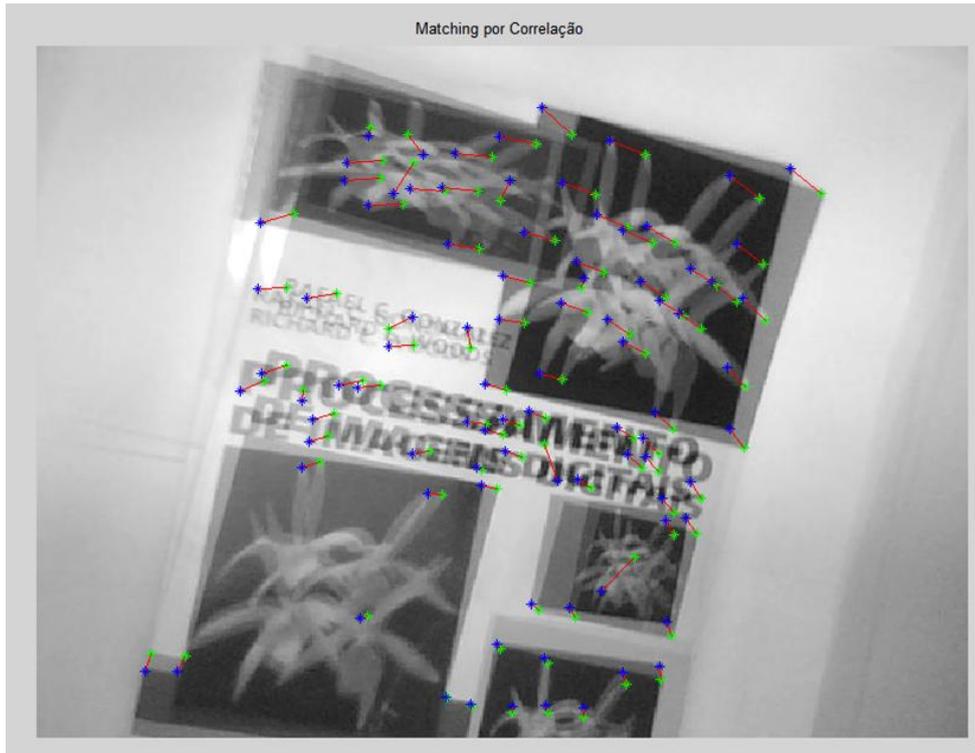
3.2.2- Casamento ou *Matching*

Com a finalidade de detectar a variação na posição de um ponto entre duas imagens se faz necessária uma etapa de casamento, isto é, encontrar o mesmo ponto em duas imagens diferentes. No presente trabalho se desenvolveu a etapa de casamento de pontos ou *Matching* entre duas imagens a partir da correlação normalizada da variância (VNC “Variance Normalized Correlation”) (Vincent e Laganière, 2001).



(a)

(b)



(c)

Figura 3.6. Resultado da aplicação do algoritmo de casamento (a) Imagem do Alvo na posição inicial (b) Imagem do alvo na posição final (c) Imagens superpostas, casamento de pontos, as linhas vermelhas indicam o movimento de cada ponto entre as duas imagens

Segundo este método, a correlação entre dois pontos p e p' , de duas imagens I e I' respectivamente, está dada por (Andreasson, et al., 2007):

$$VNC(p, p') = \frac{\sum_{i, i'}^{\kappa} (I(i) - \overline{I(p)})(I'(i') - \overline{I'(p')})}{N \sqrt{\sigma_I(p)^2 \sigma_{I'}(p')^2}} \quad (3.3)$$

onde, κ representa a janela da vizinhança da correlação, $i \in \kappa(p)$ e $i' \in \kappa(p')$, N é o número de píxeis na vizinhança, $\sigma_I(p)^2$ e $\overline{I(p)}$ são a variância e a média sobre a intensidade na vizinhança, respectivamente, o algoritmo de casamento desenvolvido neste trabalho se apresenta no Algoritmo 2 (Casamento por Correlação).

Objetivo

Dados dois vetores $p = \{p_i\}$ e $p' = \{p_j'\}$ de pontos detectados em duas imagens, determinar as correspondências $\{p_n \leftrightarrow p_n'\}$, $n \leq \min(i, j)$, i, j, n são escalares

Algoritmo

- i. Determinar o tamanho da janela de correlação N e a janela de busca D
- ii. Para cada ponto em p_i , calcular:
 - a. $I(p_i)$: imagem centrada em p_i com tamanho igual a $N \times N$
 - b. $\overline{I(p_i)}$: média
 - c. $\sigma_I(p_i)^2$: variância
- iii. Identificar os pontos em p' que estão dentro da janela de busca D ao redor de p_i
- iv. Criar um vetor de candidatos na segunda imagem: p^* com o resultado de iii
- v. Para cada ponto p_j^* , calcular ii. a. b. c.
- vi. Determinar VNC (equação 3.3) e armazenar para cada par p_i, p_j^*
- vii. Escolher para cada p_i , o par $p_i \leftrightarrow p_j^*$ com maior valor VNC
- viii. Retornar as posições de $p_i \leftrightarrow p_j^*$

Algoritmo 2. Casamento por Correlação

3.2.3- Casamento Robusto

Embora o comportamento do algoritmo de casamento seja satisfatório para casos nos quais o deslocamento entre imagens seja pouco (Figura 3.6), em sistemas de controle visual pode-se dar o caso em que duas imagens representem posições do objeto afastadas em grandes medidas. Para este tipo de situação o algoritmo de casamento deve ser robusto e conduzir a uma solução de correspondência o suficientemente aceitável para cumprir a tarefa de controle. Com este objetivo, propõe-se neste trabalho implementar um algoritmo de casamento baseado no cálculo da homografia H relativa às duas vistas da cena tirando vantagem da natureza plana do objeto alvo. Como o cálculo da homografia se faz só com a informação das correspondências iniciais, o método precisa de uma otimização para encontrar a melhor solução, como se verá a seguir.

3.2.3.1- Cálculo da Matriz de Homografia \mathbf{H}

A homografia é uma relação geométrica da posição de um objeto plano em duas imagens diferentes (Hartley e Zisserman, 2003) (Faugeras e Luong, 2001) e que pode-se expressar como:

$$\alpha \mathbf{p}_i = \mathbf{H} \mathbf{p}_i^* \quad (3.4)$$

onde \mathbf{p}_i são pontos do objeto na imagem I e \mathbf{p}_i^* são esses mesmos pontos na imagem I^* , α é um escalar positivo e \mathbf{H} é a matriz de homografia de dimensões (3×3) .

Para uma descrição detalhada do cálculo da matriz de Homografia ver a seção 2.1.4- e o Apêndice A. A matriz \mathbf{H} , em função das coordenadas na imagem dos pontos se obtém a partir da equação (3.4):

$$\begin{bmatrix} \mathbf{0}^T & -z_i^* \mathbf{p}_i^T & -y_i^* \mathbf{p}_i^T \\ z_i^* \mathbf{p}_i^T & \mathbf{0}^T & -x_i^* \mathbf{p}_i^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \quad (3.5)$$

, ou:

$$\mathbf{A}_i \mathbf{h} = 0$$

, onde

$$\mathbf{p}_i^* = \begin{bmatrix} x_i^* \\ y_i^* \\ z_i^* \end{bmatrix}, \mathbf{h} = \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix}, \mathbf{0}^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

A matriz de homografia se encontra a partir do vetor \mathbf{h} , na relação

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (3.6)$$

Para que este sistema tenha solução, precisam-se pelo menos quatro pares de pontos correspondentes da forma $(\mathbf{p}_i, \mathbf{p}_i^*)$, o método de solução linear direto (*DLT*: “Direct

Linear Transformation”) proposto por Hartley e Zisserman (2003) para resolver o cálculo de \mathbf{H} se resume no Algoritmo 3 (cálculo da Homografia).

Se o cálculo da matriz \mathbf{H} for feito com qualquer quádrupla de pares $\{p_n \leftrightarrow p_n^*\}$ será apresentada uma numerosa quantidade de falsos positivos no processo de casamento. Desta forma, a estimação de \mathbf{H} tem que passar por um método de escolha robusto (Hartley e Zisserman, 2003). Neste caso, utilizou-se o método RANSAC (Fischler e Bolles, 1981) (“*Random Sample Consensus*”) para determinar de uma maneira robusta a homografia correspondente entre dois planos.

Objetivo

Dado quatro ou mais pares de correspondências $\{p_n \leftrightarrow p_n'\}$ entre duas imagens, determinar a Homografia H tal que $p_n' = Hp_n$.

Algoritmo

- i. Para cada par de correspondências $\{p_n \leftrightarrow p_n'\}$ determinar a matriz \mathbf{A}_i .
- ii. Empilhar as n matrizes obtidas em i. numa única matriz \mathbf{A} de tamanho $2n \times 9$.
- iii. Calcular a decomposição em valores singulares (SVD) de \mathbf{A} (Ver Apêndice B).
- iv. O vetor singular unitário correspondente ao menor valor singular em de \mathbf{A} , é o valor de \mathbf{h} (ultima coluna de \mathbf{V})
- v. Determinar \mathbf{H} com (3.6)

Algoritmo 3. Cálculo da Matriz de Homografia

3.2.3.2- RANSAC

O método RANSAC introduzido por Fischler e Bolles (1981) tem como objetivo ajustar um modelo para um conjunto de dados experimentais que possam ter uma porcentagem elevada de erros. Este método, ao contrário dos métodos convencionais de ajuste, procura obter o modelo a partir da menor quantidade de dados possível. A seguir, apresenta-se o algoritmo RANSAC (Algoritmo 4) descrito em Fischler e Bolles (1981).

Objetivo

Dado o mínimo número de dados (n) que requer um modelo M para ser definido, e um conjunto P de dados cujo número é maior do que n , encontrar o modelo que melhor se ajuste ao conjunto P .

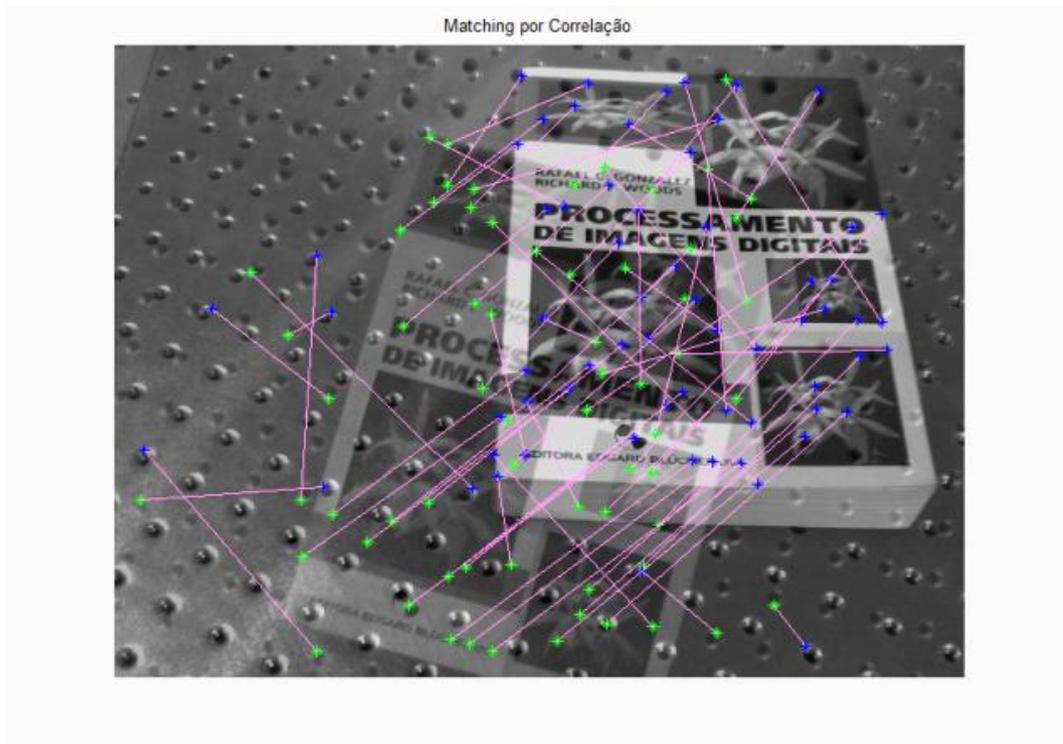
Algoritmo

- i. Escolher um subconjunto S_1 de n dados de P
- ii. Definir o modelo M_1 com o subconjunto S_1
- iii. Com o modelo M_1 determinar um subconjunto S_1^* de pontos de P que se ajustem a M_1 com certo valor de tolerância t . (S_1^* é chamado de conjunto de consenso de S_1).
- iv. Se o número de pontos em S_1^* é maior que um limiar T , calcula-se um novo modelo M_1^* com esses pontos S_1^* .
- v. Se o número de pontos em S_1^* é menor que um limiar T , escolhe-se de novo um subconjunto S_2 e repete o processo de ii até v durante N tentativas.
- vi. Depois de N tentativas escolhe-se o maior conjunto de consenso e o modelo se re-estima com ele.

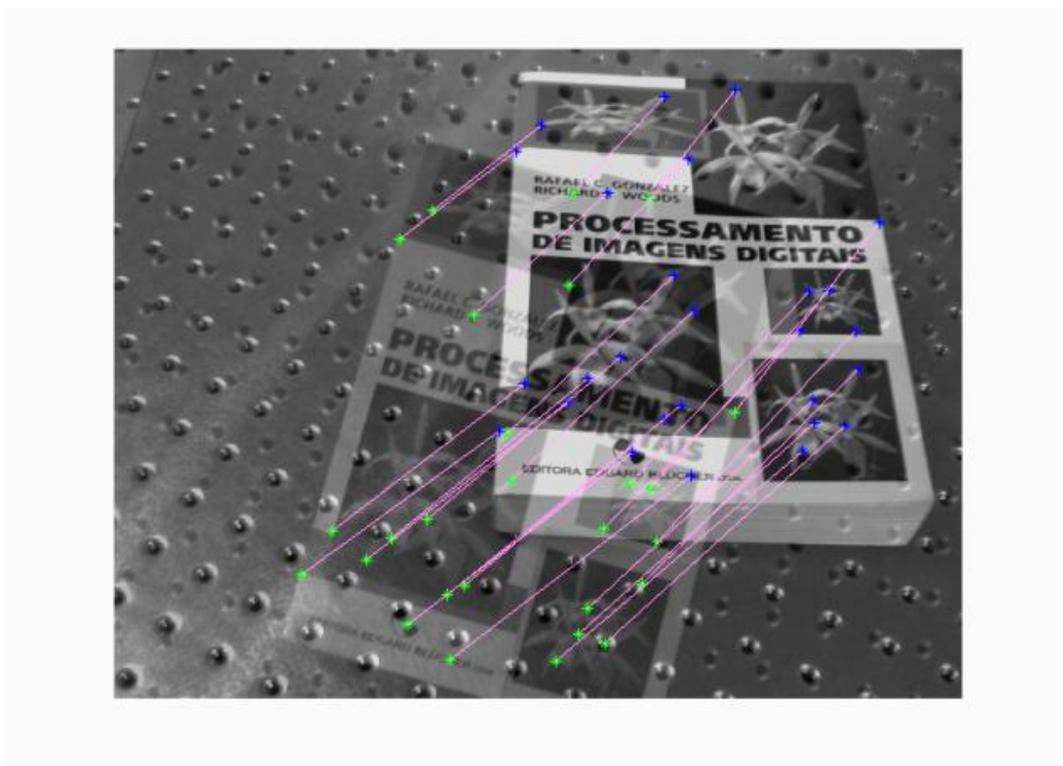
Algoritmo 4. RANSAC

Os valores de t , T e N , são funções do desvio padrão de P , assim como da exatidão requerida para o modelo. Em Hartley e Zisserman (2003) se apresenta detalhadamente a obtenção destes valores.

A estimativa da matriz de Homografia obtém-se da aplicação do método RANSAC ao conjunto de P de correspondências de pontos obtidos do algoritmo de casamento, e o modelo M calcula-se a partir do algoritmo de homografia. Na Figura 3.7 se apresentam as imagens correspondentes à aplicação do Casamento por correlação (Algoritmo 2) e o Casamento Robusto (Algoritmo 5) a duas imagens deslocadas de uma mesma cena.



(a)



(b)

Figura 3.7. Aplicação do casamento (a) método VNC (b) método casamento Robusto

Objetivo

Dado um vetor de correspondências entre duas imagens $\{p_n \leftrightarrow p_n'\}$, obtidos com o método de casamento por correlação (Algoritmo 2), escolher aqueles pares que cumprem com a restrição dada por a matriz de Homografia que relaciona as duas imagens.

Algoritmo

- i. Determinar com o método RANSAC (Algoritmo 4) a matriz de Homografia H que melhor se ajuste aos pontos $\{p_n \leftrightarrow p_n'\}$
- ii. Identificar os pontos de $\{p_n \leftrightarrow p_n'\}$ que cumprem com esta restrição H
- iii. Criar um novo vetor de correspondências a partir dos pontos identificados em ii.

Algoritmo 5. Casamento Robusto

3.2.4- Cálculo da Postura do Objeto Alvo

O objetivo primário desta etapa é achar a posição e orientação da câmera com respeito a um objeto fixo a partir de uma única imagem. Para este fim, faz-se necessário munir o sistema com algumas informações a respeito da configuração do arranjo visual, como o modelo do objeto alvo e os parâmetros intrínsecos da câmera.

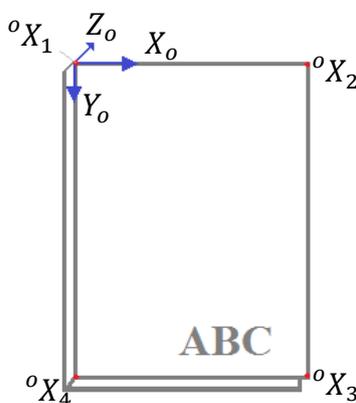


Figura 3.8. Referencial Associado ao Objeto

No caso particular deste trabalho, contam-se com objetos planos (ou de superfícies planas) que permitem obter uma relação entre eles e suas imagens mediante um mapeamento linear (Faugeras e Luong, 2001) dado pela matriz de homografia H . Na equação (3.4) expressa-se este mapeamento entre pontos de duas imagens. Esta relação pode ser feita de forma análoga entre pontos correspondentes do modelo do objeto à sua imagem.

$$\alpha \mathbf{x} = \mathbf{H} \mathbf{X} \quad (3.7)$$

Neste trabalho o modelo do objeto está composto por pontos \mathbf{X}_i cuja posição, dada em coordenadas do objeto, é conhecida. Na Figura 3.8 mostra-se uma representação de um objeto de superfície plana com seu referencial associado. Os pontos vermelhos ${}^o\mathbf{X}_i$ indicam as posições dos pontos utilizados para descrever o modelo em função de suas coordenadas.

Para encontrar a postura da câmera deve-se então decompor a matriz de homografia para achar as matrizes de rotação e translação associadas à postura da câmera. Da equação (2.7) temos que a projeção de um ponto a través de uma câmera *pinhole* é da forma

$$\beta \mathbf{x} = \mathbf{P} \mathbf{X} \quad (3.8)$$

, com \mathbf{P} sendo a matriz de projeção da câmera que é dada pela relação:

$$\mathbf{P} = \mathbf{K} \mathbf{R} [\mathbf{I}_3 | \mathbf{t}] \quad (3.9)$$

, onde \mathbf{K} é a matriz de calibração da câmera, \mathbf{R} é a matriz de rotação que determina a orientação da câmera, \mathbf{I}_3 é a matriz identidade de $[3 \times 3]$ e \mathbf{t} é o vetor de posição da câmera.

No caso particular de objetos planos um ponto pode definir-se, sem perda de generalidade da forma $\mathbf{X} = (x, y, k)$ (Johansson e Cipolla, 2002). A projeção, então, é da forma:

$$\mathbf{P} = \mathbf{K} \mathbf{R} \begin{bmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & t_3 \end{bmatrix} \quad (3.10)$$

Em forma de vetores coluna

$$\mathbf{R} \begin{bmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & t_3 \end{bmatrix} = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] \quad (3.11)$$

Igualando com a equação (3.7)

$$\mathbf{H} = \lambda \mathbf{K} [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] \quad (3.12)$$

Desenvolvendo

$$\mathbf{K}^{-1}[\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] = \lambda [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] \quad (3.13)$$

Neste ponto se têm em conta as propriedades de ortonormalidade da matriz \mathbf{R}

$$\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1 \quad \text{e} \quad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \quad (3.14)$$

Assim

$$\|\mathbf{K}^{-1}\mathbf{h}_1\| = \|\mathbf{K}^{-1}\mathbf{h}_2\| = \lambda \quad (3.15)$$

uma solução é da forma

$$\lambda = \frac{\|\mathbf{K}^{-1}\mathbf{h}_1\| + \|\mathbf{K}^{-1}\mathbf{h}_2\|}{2} \quad (3.16)$$

, e,

$$[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] = \frac{\mathbf{K}^{-1}\mathbf{H}}{\lambda} \quad (3.17)$$

Objetivo

Dadas as coordenadas de 4 pontos de um objeto plano no referencial do objeto, suas correspondentes coordenadas na imagem $\{X_i \leftrightarrow x_i\}$, e a matriz de calibração \mathbf{K} da câmera

Encontrar a postura \mathbf{P} da câmera com respeito do objeto.

Algoritmo

- i. Encontrar a matriz de homografia \mathbf{H} que relaciona os pontos $\{X_i \leftrightarrow x_i\}$ utilizando o Algoritmo 3.
- ii. Calcular o valor de λ utilizando a equação (3.16)
- iii. Encontrar $[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}]$ a partir da equação (3.17)
- iv. Calcular \mathbf{r}_3 de (3.14)
- v. Compor a matriz de projeção \mathbf{P} com a equação (3.9)
- vi. Adicionar a restrição: se $\mathbf{t} < 0$, rotar π a matriz \mathbf{P} na direção de Z

Algoritmo 6. Estimativa da Postura da Câmera

Com a equação (3.17) é possível encontrar duas soluções da postura da câmera. Então se adiciona uma restrição de positividade na direção do eixo z . O Algoritmo 6 (Estimativa da Postura da câmera), desenvolvido neste trabalho, descreve o processo de obtenção da postura da câmera em relação a um objeto plano.

3.3- LEI DE CONTROLE

No presente trabalho se aplica um controle proporcional com um decaimento idealmente exponencial do erro, (equação 2.25). Esta aproximação permite a projeção de uma lei de controle simples, no espaço da imagem ou no espaço de trabalho, que estabelece o comando diretamente em função da velocidade da ferramenta num manipulador robótico, ignorando a dinâmica do sistema. Apesar da sua simplicidade, esta aproximação tem sido testada com sucesso em grande número de publicações.

Como estabelecido anteriormente, a diferença entre as duas aproximações de controle visual propostas está na definição do espaço do comando $\mathbf{s}(\mathbf{r}, t)$. Para o caso do IBVS o $\mathbf{s}(\mathbf{r}, t)$ está definido como as posições em coordenadas da imagem das características detectadas no tempo t pelos algoritmos descritos anteriormente.

$$\mathbf{s}(\mathbf{r}, t) = (p_1(t), p_2(t), \dots, p_n(t)), n \geq 4 \quad (3.18)$$

onde p_i representa um ponto em uma imagem com coordenadas em píxeis, (u_i, v_i) , e n é o número de pontos. Analogamente o vetor \mathbf{s}^* , de posições desejadas, está também definido como sendo as coordenadas em píxeis dos n pontos na posição alvo.

Para o controle PBVS o vetor $\mathbf{s}(\mathbf{r}, t)$ está composto pela postura da câmera em função de um referencial externo no tempo t . Esta postura indica tanto a posição como a orientação da câmera no espaço cartesiano e é obtida a partir dos algoritmos de detecção e recuperação da postura descritos anteriormente:

$$\mathbf{s}(\mathbf{r}, t) = ({}^o\mathbf{r}_c(t)) \quad (3.19)$$

No Algoritmo 7 (Lei de Controle Visual), desenvolvido para esta pesquisa, se faz uma descrição dos passos necessários para encontrar a variável de controle em cada uma das abordagens de controle servo visual estudadas neste trabalho.

Objetivo

Dados o conjunto $\mathbf{s}(r, t)$, \mathbf{s}^* e os parâmetros intrínsecos da câmera, calcular o valor de variável de controle

Algoritmo

i. Para o Controle Visual baseado em Imagem:

- a. Estimar o valor da Matriz de Interação \hat{L} dos pontos detectados na imagem com a equação (2.43) a partir dos parâmetros intrínsecos da câmera e assumindo uma profundidade Z constante igual a z .
- b. Calcular a Pseudo-inversa Moore-Penrose da matriz \hat{L} : \hat{L}^+
- c. Encontrar o erro tal que

$$e = (s(r, t) - s^*)$$

- d. Estimar a variável de controle como a velocidade da câmera com a relação dada em (2.42) e $\lambda > 0$.

$$\dot{r} = -\lambda \hat{L}_s^+ e$$

ii. Para o Controle Visual Baseado em Posição

- a. A partir do cálculo da postura do objeto com respeito da câmera em sua posição atual cT_o e com respeito da câmera e em sua posição ideal ${}^{c^*}T_o$ utilizando o Algoritmo 6, encontrar a transformação homogênea entre as posições da câmera

$${}^{c^*}T_c = [R | {}^{c^*}t_c]$$

- b. Com estes valores de rotação e translação encontrar a velocidade da câmera, utilizando a equação (2.52)

Algoritmo 7. Lei de Controle Visual

3.4- CONTROLE CINEMÁTICO DO MANIPULADOR

Depois da análise das imagens, o seguinte passo na elaboração do controle visual foi efetivar a movimentação do manipulador robótico (e portanto da câmera) a partir do parâmetro de velocidade da câmera calculado, (ver Figura 3.3 e Figura 3.4), que neste caso por tratar-se de uma configuração *eye-in-hand* seria a mesma velocidade da ferramenta ou elemento terminal do manipulador. Para cumprir esta tarefa é preciso o conhecimento das relações entre o movimento do robô, em função de um referencial externo, com as posições de seus elos. Estas relações são dadas pelo modelo cinemático do robô, que faz uma descrição analítica da relação entre a posição e orientação do elemento terminal (ferramenta do robô) e os valores de suas coordenadas articulares.

A cinemática direta é uma ferramenta que permite determinar qual a posição e orientação do elemento terminal do robô, com respeito a um sistema de coordenadas de referência, em função dos valores das posições das juntas e dos parâmetros geométricos dos elos do robô, e a cinemática inversa resolve a configuração que deve adotar o robô para atingir uma posição e orientação específica do elemento terminal (Figura 3.9).

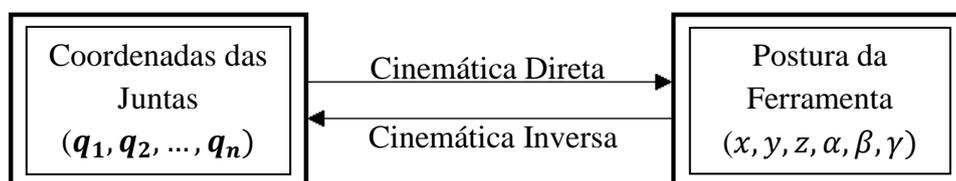


Figura 3.9. Relação entre cinemática direta e cinemática inversa, adaptado de (Barrientos, et al., 2007)

3.4.1- Cinemática Direta do Robô

O interesse desta pesquisa é modelar um sistema de controle visual que possa ser implantado num manipulador robótico industrial de 6 graus de liberdade (6DOF). Com esta finalidade escolheu-se o modelo do robô ABB IRB2000, chamado IRB2000 daqui para frente, o qual tem sete elos, incluindo a base, unidos por seis juntas.

A cinemática direta do robô relaciona a postura do elemento terminal com a postura de cada junta, valendo-se do produto das matrizes de transformação homogênea de cada elo. Então, para um robô de 6 graus de liberdade, como o IRB2000, tem-se a seguinte relação:

$${}^0T_6 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6 \quad (3.20)$$

, onde as matrizes jA_i expressam a posição e a orientação do referencial j respeito ao referencial i .

Uma forma de se representar (3.20) é mediante a definição dos parâmetros de Denavit e Hartenberg (D-H) (Craig, 1989) (Apêndice C). Para o manipulador IRB2000, estes parâmetros são apresentados na Tabela 3.1 e foram obtidos mediante as relações dos referenciais associados às juntas como se apresenta na Figura 3.10.

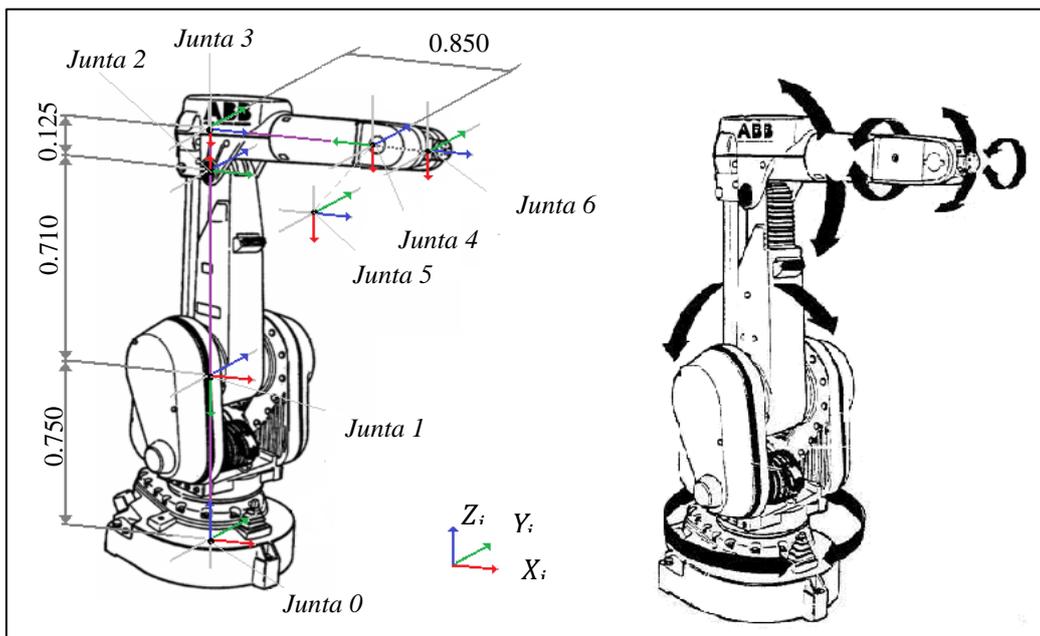


Figura 3.10. Referenciais das juntas na obtenção dos parâmetros D-H, (unidades em metros)

Tabela 3.1. Parâmetros D-H do Manipulador IRB2000

Junta	θ_i (rad)	d_i (m)	a_i (m)	α_i (rad)
1	q_1	0.750	0	$\pi/2$
2	q_2	0	0.710	0
3	q_3	0	0.125	$\pi/2$
4	q_4	0.850	0	$-\pi/2$
5	q_5	0	0	$\pi/2$
6	q_6	0	0	0

Com os parâmetros *D-H* a matriz de transformação *A* para cada junta define-se:

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.21)$$

Tendo então os valores θ_i das articulações e os parâmetros D-H de um manipulador é possível encontrar a transformação homogênea entre o elemento terminal e a base do robô utilizando as equações (3.21) e (3.20).

Objetivo

Dado:

- Os parâmetros de D-H para o manipulador IRB2000
- O vetor de configuração das juntas $q = [q_1, q_2, \dots, q_6]$

Encontrar a Matriz de Transformação 0T_6 da postura do elemento terminal em função da base do robô.

Algoritmo

- i. Se a posição inicial da ferramenta 0T_f não é conhecida, fazer ${}^0T_f = [I_{[4 \times 4]}]$.
- ii. Para cada eixo encontrar o valor da matriz ${}^{i-1}A_i$ com a equação (3.21) e os seus parâmetros D-H.
- iii. Ir formando a transformação como o produto $T = T * {}^{i-1}A_i$ com $i = [1, 2, \dots, 6]$
- iv. Finalmente aplicar a transformação da posição inicial ${}^0T_6 = T * {}^0T_f$

Algoritmo 8. Cinemática Direta do Manipulador IRB2000

3.4.2- Jacobiano do robô *J*

Como visto, para encontrar a postura do elemento terminal do manipulador e portanto da câmera, é preciso contar com o vetor dos ângulos das juntas $q = [\theta_i]$. Mas o controle visual fornece a informação da velocidade da câmera (equações (2.32) e (2.42)), a qual está no espaço cartesiano.

A forma de relacionar as velocidades das articulações de um robô com a velocidade de seu elemento terminal é mediante a matriz Jacobiana do manipulador (ou simplesmente Jacobiano), a qual é uma função da postura do manipulador que permite conhecer as velocidades do elemento terminal do robô a partir das velocidades de cada junta. Ao contrário, a matriz Jacobiana inversa permite conhecer as velocidades necessárias nas juntas para atingir uma velocidade determinada no elemento terminal do robô (Figura 3.11).

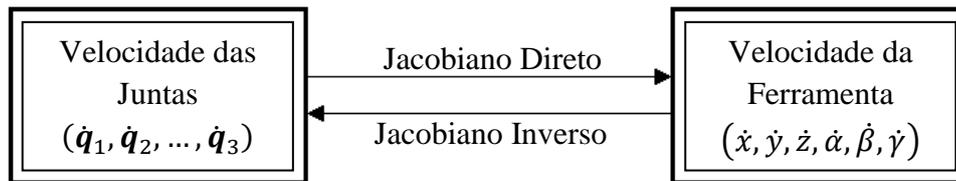


Figura 3.11. Relação entre o Jacobiano e o Jacobiano Inverso, adaptado de (Barrientos, et al., 2007)

Para encontrar a matriz Jacobiana de um robô faz-se necessária uma análise diferencial das posições, tanto das juntas como do elemento terminal do robô (Craig, 1989). Da cinemática direta têm-se as relações:

$$\begin{aligned} x &= f_x(q_1, \dots, q_n) & y &= f_y(q_1, \dots, q_n) & z &= f_z(q_1, \dots, q_n) \\ \alpha &= f_\alpha(q_1, \dots, q_n) & \beta &= f_\beta(q_1, \dots, q_n) & \gamma &= f_\gamma(q_1, \dots, q_n) \end{aligned} \quad (3.22)$$

, derivando em função do tempo, obtém-se :

$$\begin{aligned} \dot{x} &= \sum_1^n \frac{\partial f_x}{\partial q_i} \dot{q}_i & \dot{y} &= \sum_1^n \frac{\partial f_y}{\partial q_i} \dot{q}_i & \dot{z} &= \sum_1^n \frac{\partial f_z}{\partial q_i} \dot{q}_i \\ \dot{\alpha} &= \sum_1^n \frac{\partial f_\alpha}{\partial q_i} \dot{q}_i & \dot{\beta} &= \sum_1^n \frac{\partial f_\beta}{\partial q_i} \dot{q}_i & \dot{\gamma} &= \sum_1^n \frac{\partial f_\gamma}{\partial q_i} \dot{q}_i \end{aligned} \quad (3.23)$$

o que em forma matricial, pode-se escrever:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{q}_n \end{bmatrix}; \quad \mathbf{J} = \begin{bmatrix} \partial f_x / \partial q_1 & \dots & \partial f_x / \partial q_n \\ \vdots & \ddots & \vdots \\ \partial f_y / \partial q_1 & \dots & \partial f_y / \partial q_n \end{bmatrix} \quad (3.24)$$

, sendo \mathbf{J} a matriz Jacobiana do Manipulador.

Para encontrar o valor do Jacobiano para um conjunto de configurações articulares q_i é preciso ter uma descrição do manipulador em função destes parâmetros como na equação (3.21). Paul, et al., (1981) propõem um método sistemático do cálculo do Jacobiano a partir das mudanças diferenciais da translação e rotação do manipulador e das coordenadas da juntas, cuja aplicação ao manipulador IRB2000 se apresenta no Algoritmo 9.

3.4.3- Jacobiano Inverso \mathbf{J}^{-1}

No controle visual a informação obtida do processamento das imagens é a velocidade da câmera, ou seja, do elemento terminal do robô, em coordenadas cartesianas. Para imprimir este movimento ao manipulador, precisa-se da sua transformação em coordenadas de juntas para serem utilizadas na equação (3.21). Esta transformação de coordenadas cartesianas em coordenadas de juntas é realizada mediante o Jacobiano inverso como mostrado na Figura 3. Da equação (3.24):

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{q}_n \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad (3.25)$$

Para a aplicação apresentada neste trabalho é possível obter, a partir da velocidade do elo final, as velocidades das juntas mediante a aplicação do Jacobiano inverso do robô, o qual permite efetuar a movimentação do manipulador até a posição desejada.

Objetivo

Dado:

- O vetor de configuração das articulações $q = [q_1, q_2, \dots, q_6]$
- A Transformação Homogênea de cada elo ${}^{i-1}A_i$

Encontrar o Jacobiano \mathbf{J} do manipulador IRB2000.

Algoritmo

- Se a posição inicial da ferramenta 0T_f não é conhecida, fazer ${}^0T_f = [I_{[4 \times 4]}]$.
- Definir uma transformação $T_i = {}^0T_f$
- Para cada um dos elos fazer mudar T_i da forma:

$$T_i = {}^{i-1}A_i * T_i$$

- Como T_i é uma transformação homogênea da forma $\begin{bmatrix} \mathbf{R} | \mathbf{T} \\ \mathbf{0} | 1 \end{bmatrix}$ escolher as componentes da rotação:

$$n = [R_{11} \ R_{21} \ R_{31}]^T \quad o = [R_{12} \ R_{22} \ R_{32}]^T \quad a = [R_{13} \ R_{23} \ R_{33}]^T$$
$$p = [T_1 \ T_2 \ T_3]^T$$

- Com cada Transformação T_i formar as colunas do Jacobiano:

$$J_i = \begin{bmatrix} (n_y p_x - n_x p_y) \\ (o_y p_x - o_x p_y) \\ (a_y p_x - a_x p_y) \\ n_z \\ o_z \\ a_z \end{bmatrix}$$

- Obter o Jacobiano completo a partir de suas colunas: $\mathbf{J} = [J_1 \ J_2 \ J_3 \ J_4 \ J_5 \ J_6]$

Algoritmo 9. Jacobiano do IRB2000

4 - RESULTADOS E ANÁLISE

O sucesso das técnicas de controle visual está definido pela sua capacidade de cumprir integralmente a tarefa para a qual foi planejado. Neste contexto, cada etapa do processo deve desempenhar seu papel com a maior precisão possível. Para identificar as fraquezas e as virtudes dos sistemas de controle implementados durante esta pesquisa foram feitos testes de cada uma das etapas envolvidas. Além disso, devido a se tratar de um sistema baseado em visão, acredita-se que muitas das dificuldades do processo encontram-se justamente nestes estágios de processamento das imagens, o que pode ser verificado com os resultados obtidos e que são apresentados neste capítulo.

4.1- ANÁLISE E PROCESSAMENTO DE IMAGENS

Como já comentado em capítulos anteriores, os objetos utilizados para o desenvolvimento da aplicação continham superfícies planas que tivessem pontos característicos como cantos ou quinas. Uma vez definido o tipo de objetos, decidiu-se adquirir suas imagens em diferentes condições de iluminação, posição e de fundo para observar o comportamento dos algoritmos propostos. Os resultados aqui apresentados foram obtidos com a aplicação dos algoritmos em um grupo de 10 pares de imagens, com rotações, variação de escala e translações entre si, este conjunto de imagens se apresenta na Tabela 4.1. A maior parte dos testes foi feita com duas câmeras diferentes com o objetivo de comprovar a flexibilidade dos algoritmos.

4.1.1- Detecção de Pontos

A detecção de pontos foi feita com a aplicação do detector de cantos de Harris (Algoritmo 1). Para testar seus resultados, o algoritmo foi aplicado em imagens do mesmo objeto em diferentes posições procurando encontrar sua invariância ante a rotação, translação e variação de escala das imagens. Nas figuras da Tabela 4.1 se apresentam os resultados da detecção de pontos nas imagens de teste. Na Figura 4.1 se destacam num círculo vermelho os pontos que forem detectados nas duas imagens (coincidências) dos pares de imagens 2, 3, 6, e 8 da Tabela 4.1.

Tabela 4.1. Imagens de Teste

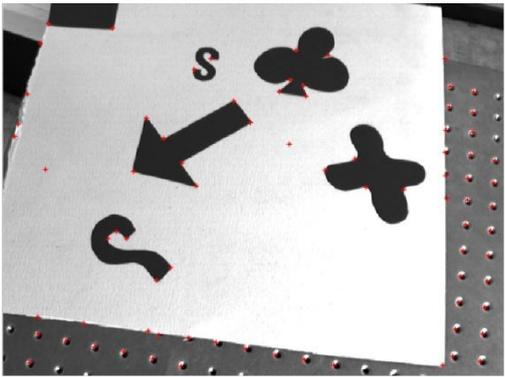
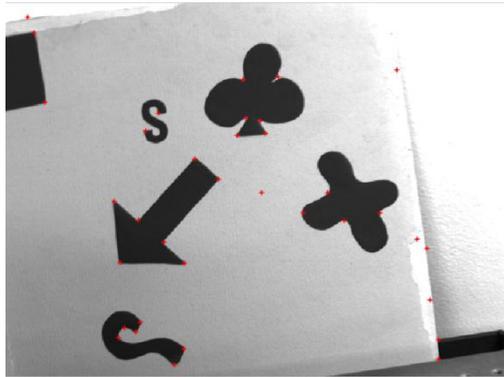
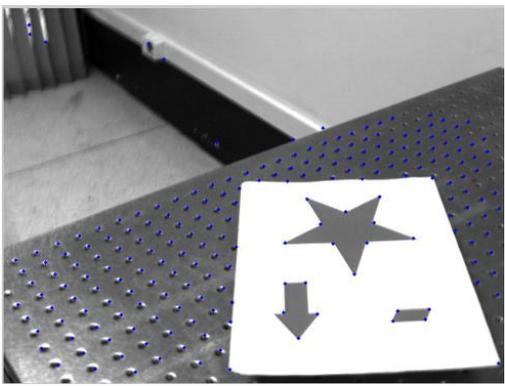
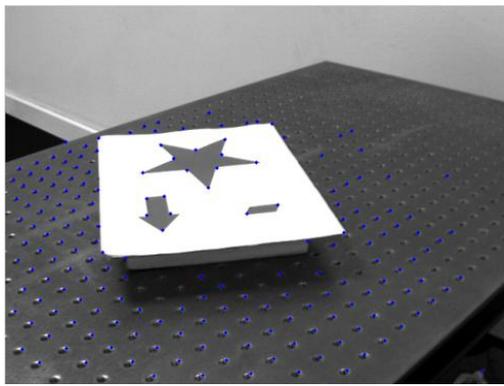
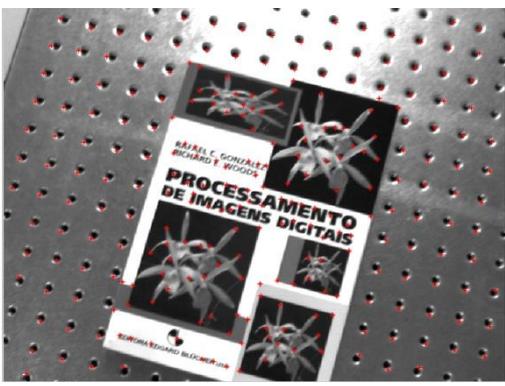
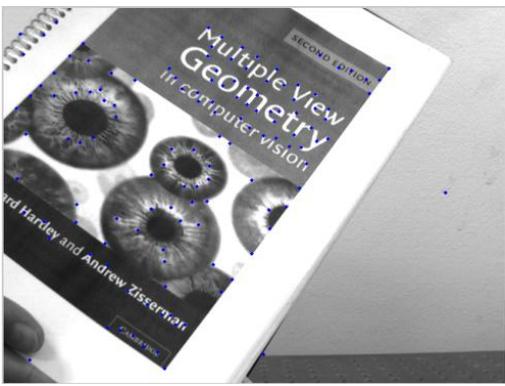
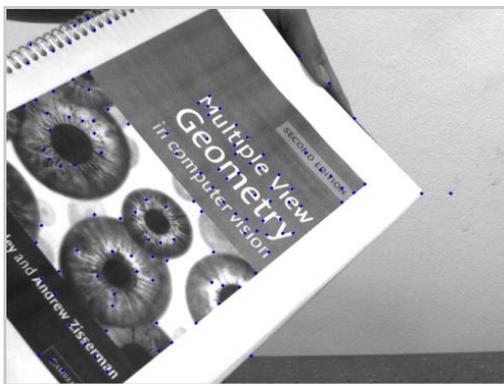
Nome	Imagem 1	Imagem 2
Par 1		
Par2		
Par3		
Par4		

Tabela 4.1. Imagens de Teste (Continuação)

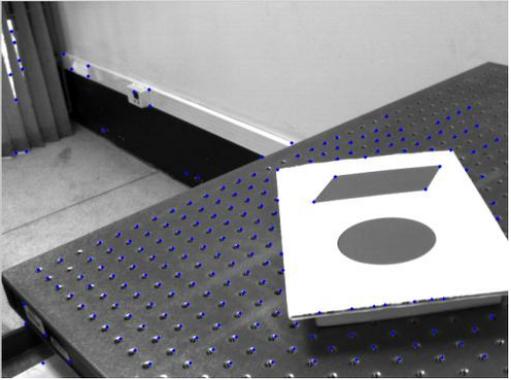
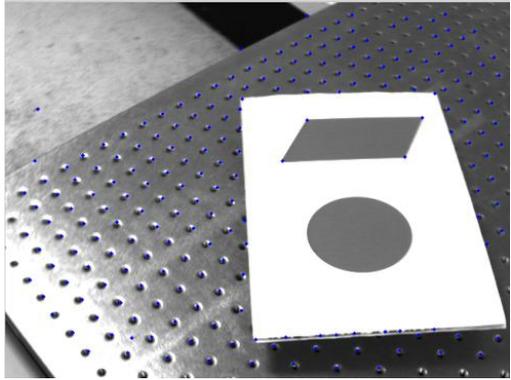
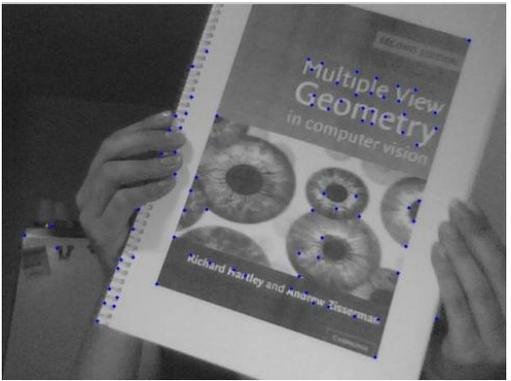
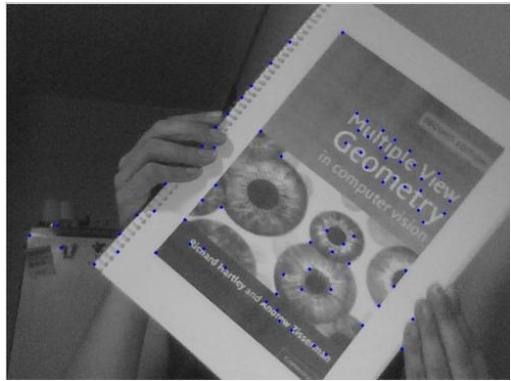
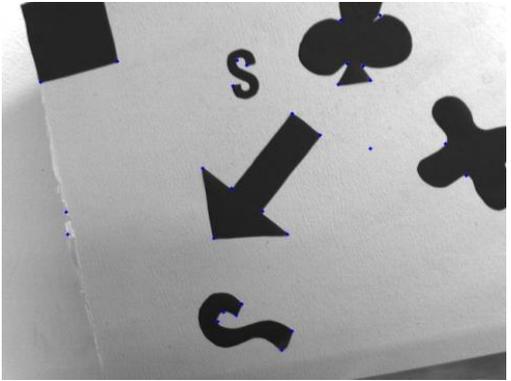
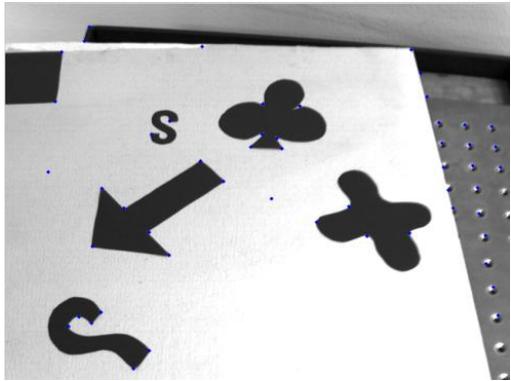
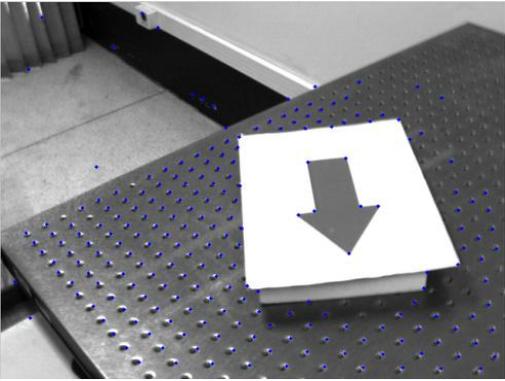
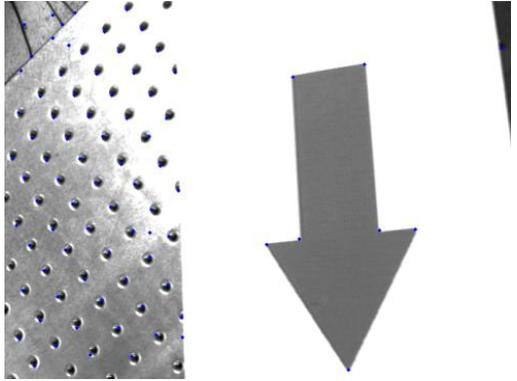
Nome	Imagem 1	Imagem 2
Par5		
Par6		
Par7		
Par8		

Tabela 4.1. Imagens de Teste (Continuação)

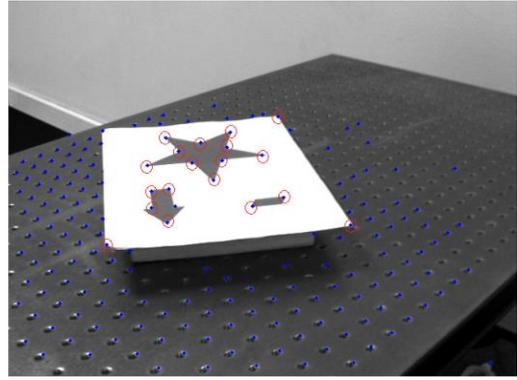
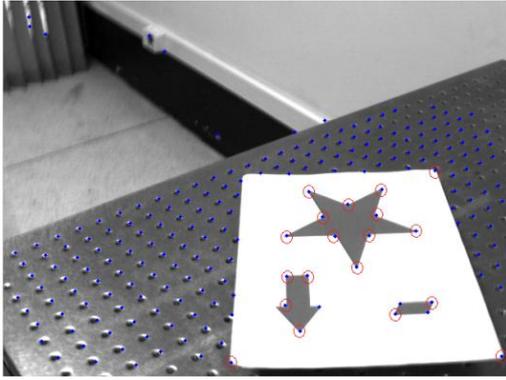
Nome	Imagem 1	Imagem 2
Par9		
Par10		

O detector de Harris foi implementado com os seguintes parâmetros (Algoritmo 1):

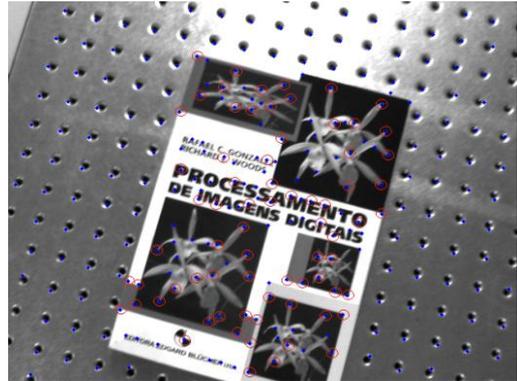
$$n = 5; \quad \sigma = 2,5; \quad w = 10; \quad \text{Limiar} = 3000$$

Os anteriores valores foram obtidos depois de uma etapa de testes, nos quais procurou-se encontrar um número consistente entre pontos detectados e o número real de cantos e quinas em todas as imagens. Estes valores foram utilizados para todas as imagens, encontrando os melhores resultados entre as diferentes combinações testadas.

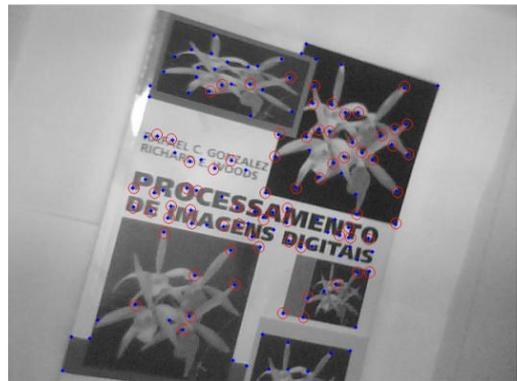
Para cada par de imagens foram comprovados o número de coincidências e foi obtida a relação entre estes valores e o número total de pontos detectados. Esclarece-se que nestas medições foi levado em conta o fato de que, na mudança da posição dos objetos, alguns pontos saem do campo visual que na vista anterior estavam presentes. Na Tabela 4.2 apresenta-se um resumo dos resultados em vários pares de imagens, e como pode-se observar, embora sejam imagens do mesmo objeto, nem sempre os pontos detectados são iguais.



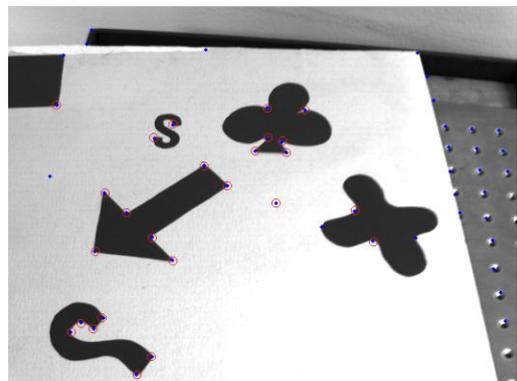
(a)



(b)



(c)



(d)

Figura 4.1. Detecção de Cantos com Algoritmo de Harris

Tabela 4.2. Resultados da Detecção de Cantos

<i>Imagens</i>	<i>Cantos Detectados, presentes em ambas visuais</i>		<i>Coincidências</i>	<i>Porcentagem de acerto</i>	
	<i>Imagem 1</i>	<i>Imagem 2</i>		<i>Imagem 1</i>	<i>Imagem 2</i>
Par 1	34	28	26	76,47 %	92,86 %
Par 2	22	21	19	86,36 %	90,48 %
Par 3	120	133	72	60,00 %	54,00%
Par 4	131	171	94	71,76%	54,97%
Par 5	14	15	12	85,71%	80,00%
Par 6	191	136	75	39,27%	55,15%
Par 7	83	75	56	67,47%	74,67%
Par 8	26	28	25	96,15%	89,28%
Par 9	8	8	8	100%	100%
Par 10	94	103	30	31,91%	29,12%

Nas imagens texturizadas (pares 3, 4, 6, 7, 10) a quantidade de pontos detectados aumenta consideravelmente em relação àquelas imagens que têm cores sólidas com fundos claros (Figura 4.1 (a) e (d)), mas na mesma medida estas últimas imagens apresentam uma maior quantidade de acertos. Segundo estes resultados, as mudanças de posição afetam em maior medida a detecção dos pontos em imagens altamente texturizadas como nos pares 10 e 6. No entanto, a quantidade de pontos ainda é válida para a seguinte etapa do sistema proposto.

Os pares de imagens 6 e 9 apresentam uma maior variação da escala e pode-se deduzir dos resultados, que enquanto os objetos se encontrem claramente definidos nas duas imagens, esta variação não afeta à detecção. O detector de cantos apresentou bons resultados em imagens com grandes translações e rotações, mas de novo a qualidade das imagens tem uma grande influência nos resultados.

Em geral o detector de Harris apresenta um bom comportamento na detecção de cantos e quinas, elementos que exibem invariância frente à translação, variação de escala e rotação. Por este motivo, o detector de Harris foi utilizado como o método de identificação das características nas imagens durante o desenvolvimento desta pesquisa.

4.1.2- Casamento por correlação

Uma vez obtidas as coordenadas dos pontos característicos de cada imagem implementou-se o casamento por correlação (Algoritmo 2), no qual procurou-se corresponder pontos em

duas imagens diferentes da mesma cena. Na Figura 4.2 apresenta-se o resultados da execução do casamento de pontos entre as imagens do par 8 da Tabela 4.1. Um dos objetivos da correspondência ou casamento dentro deste trabalho é quantificar o deslocamento dos pontos entre duas imagens para posteriormente definir uma medida do erro entre a posição desejada e a posição atual do objeto alvo. Medições erradas de correspondências conduzem a medidas incorretas do deslocamento, o qual poderia causar um comportamento indesejado do controle visual. Na Figura 4.2 podem-se observar dentro dos círculos vermelhos algumas inconsistências no emparelhamento dos pontos fornecidos pelo algoritmo de Casamento por Correlação.

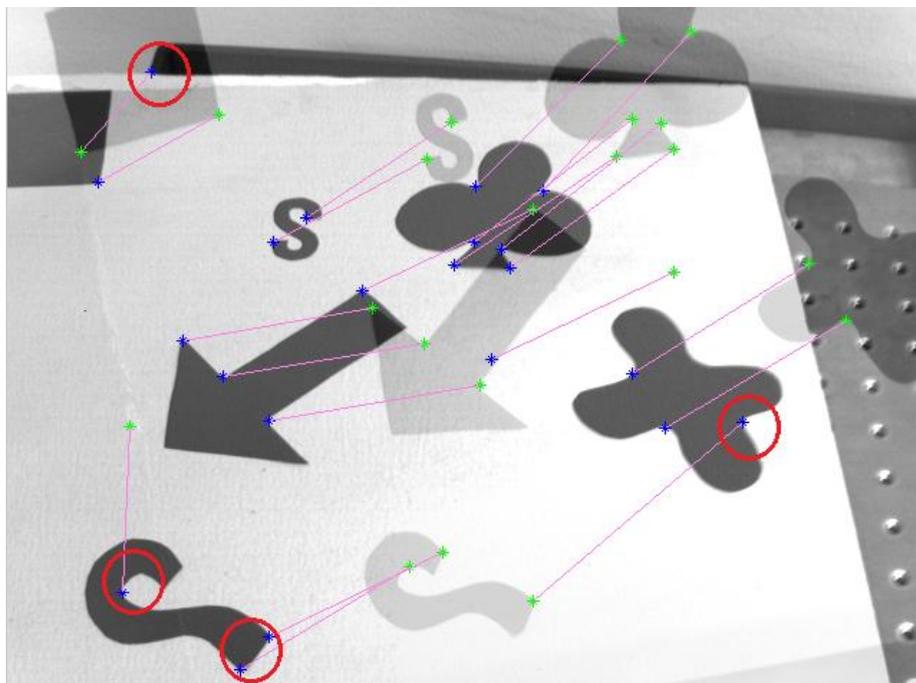


Figura 4.2. Casamento por Correlação

Nas Figuras 4.3 e 4.4 apresentam-se os resultados da aplicação do algoritmo de casamento em duas das imagens da Tabela 4.1 (Par 6 e Par 9). A análise de acerto para estas imagens se apresenta na Tabela 4.3. Nota-se que as imagens mais texturizadas apresentam uma maior quantidade de emparelhamentos incorretos, porém a porcentagem de acerto também aumenta em relação aos pontos detectados.



Figura 4.3. Casamento por correlação do Par 6 (Ver Tabela 4.1.), no círculo vermelho se destacam os casamentos incorretos

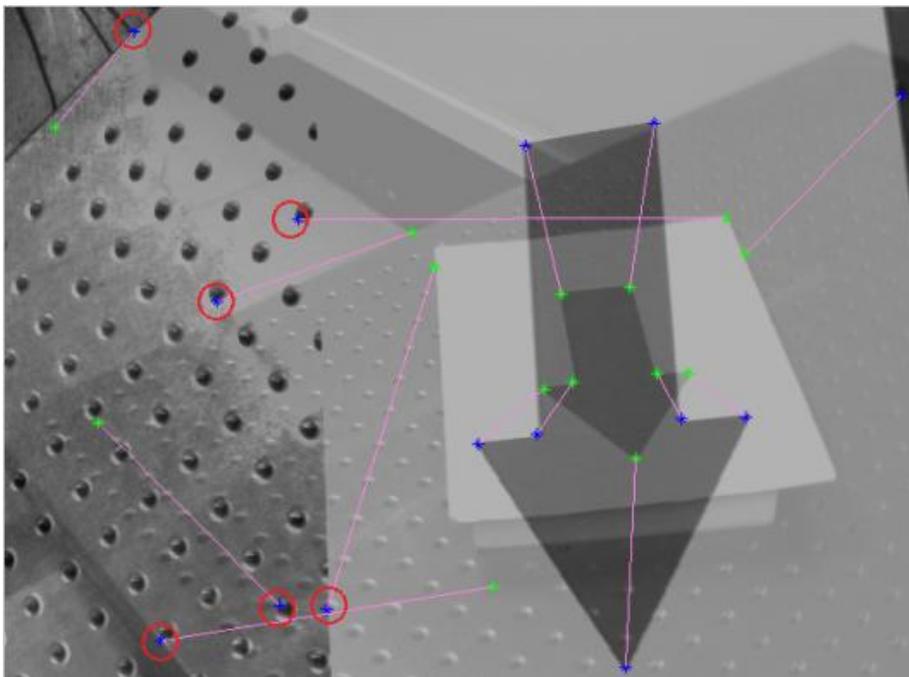


Figura 4.4. Casamento por correlação do Par 9 (Ver Tabela 4.1.), no círculo vermelho se destacam os casamentos incorretos

O algoritmo de casamento por correlação apresenta resultados muito satisfatórios em situações onde as imagens do objeto não apresentam grandes diferenças na posição ou onde os objetos estão bem definidos, pares 1, 2, 4, 8 e 9. No entanto, em alguns casos onde há variação brusca na posição e orientação do objeto, e além, os objetos estão bem

definidos (não têm boa aparência), par 3 e 10, os resultados não são confiáveis já que detecta uma grande porcentagem de correspondências erradas. Estas últimas condições são muito comuns em situações reais nas que pretende-se aplicar o controle visual. Para garantir o maior número possível de correspondências acertadas foi implementado o algoritmo de casamento robusto com a restrição da matriz de homografia (Algoritmo 5).

Tabela 4.3. Resultados do Algoritmo de Casamento por Correlação

<i>Imagens</i>	<i>Pontos Detectados</i>		<i>Correspondências</i>		<i>Porcentagem de acerto</i>
	<i>Imagem 1</i>	<i>Imagem 2</i>	<i>Corretas</i>	<i>Incorretas</i>	
Par 1	34	28	21	4	84,00 %
Par 2	22	21	18	2	90,00 %
Par 3	120	133	29	33	46,77 %
Par 4	131	171	51	12	80,95%
Par5	14	15	7	5	58,33%
Par6	191	136	31	24	56,36%
Par7	83	75	21	14	60,00%
Par8	26	28	16	4	80,00%
Par9	8	8	7	1	87,5%
Par10	94	103	9	23	31,03%

4.1.3- Casamento Robusto

A procura de pontos correspondentes ou casamento de uma forma robusta, em duas vistas de um objeto, sugere-se a imposição de restrições nos pares de pontos considerados iguais nas imagens (correspondências). Neste trabalho, sugere-se a restrição dada pela matriz de homografia, resultando em pontos correspondentes se, além de apresentar uma alta correlação nos valores de intensidade de sua vizinhança, satisfazem a transformação homográfica que relaciona as imagens.

Infere-se imediatamente que o comportamento do casamento surge do cálculo correto da matriz de homografia entre as duas imagens, mas isto não pode ser garantido na presença de uma grande quantidade de correspondência erradas como as apresentadas na Tabela 3. Por este motivo foi utilizado o método de RANSAC (Algoritmo 4) para encontrar o melhor valor da homografia, dentre os dados iniciais do casamento fornecidos pelo algoritmo de casamento por correlação.

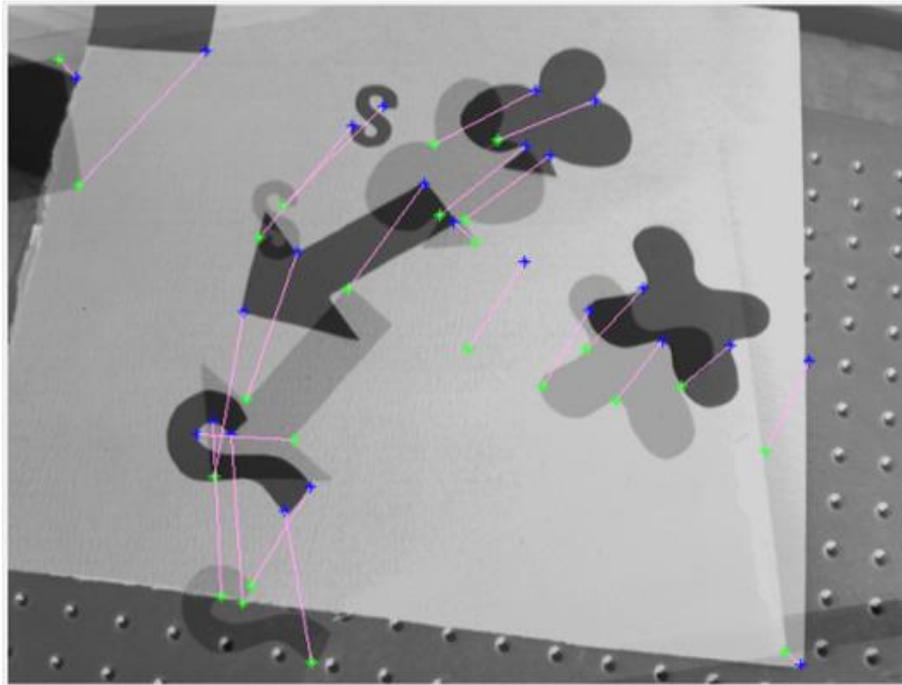
A seguir, na Tabela 4.4, apresentam-se os resultados obtidos do cálculo da Homografia H (Algoritmo 3), tanto com todos os pontos, como utilizando o método RANSAC (Algoritmo 5). Os valores de erro forem calculados a partir da reprojeção dos pontos com a homografia obtida por cada um dos métodos.

Tabela 4.4. Comparação do cálculo da Homografia

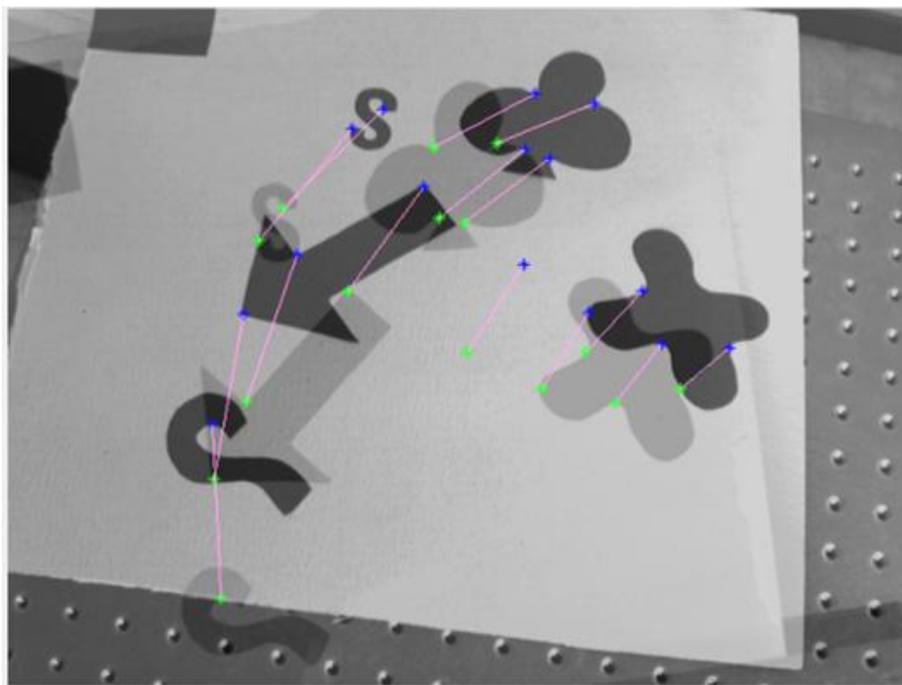
<i>Imagens</i>	<i>Número de correspondências em toda a imagem</i>		<i>Desvio Padrão [píxeis]</i>	
	Sem RANSAC	Com RANSAC	Sem RANSAC	Com RANSAC
Par 1	25	15	31.9874	0.5610
Par 2	54	11	127.7160	0.7681
Par 3	62	29	597.2767	0.6995
Par 4	63	37	81.8465	0.5727
Par 5	51	12	507.4117	0.7691
Par 7	55	17	559.6182	1.0103
Par 6	35	7	399.7412	0.4385
Par 8	20	11	88.8480	0.5090
Par 9	18	5	243.62	0.6938
Par 10	32	9	109.7905	0.3940

Como esperado, o fato da Matriz de Homografia, H , ser obtido a partir de casamentos errados (casamento só com correlação) faz com que seu valor seja pouco confiável e não garante um mapeamento correto dos pontos nas duas imagens. Pelo contrário, a Homografia calculada com o método RANSAC elimina as correspondências erradas e, em grande medida, reproduz o mapeamento dos pontos nas imagens corretamente.

Nas Figura 4.5 à 4.8. são apresentados os resultados da aplicação do casamento com correlação (Algoritmo 2) e com a restrição imposta pela homografia (Algoritmo 4), para alguns pares de imagens da Tabela 4.1.

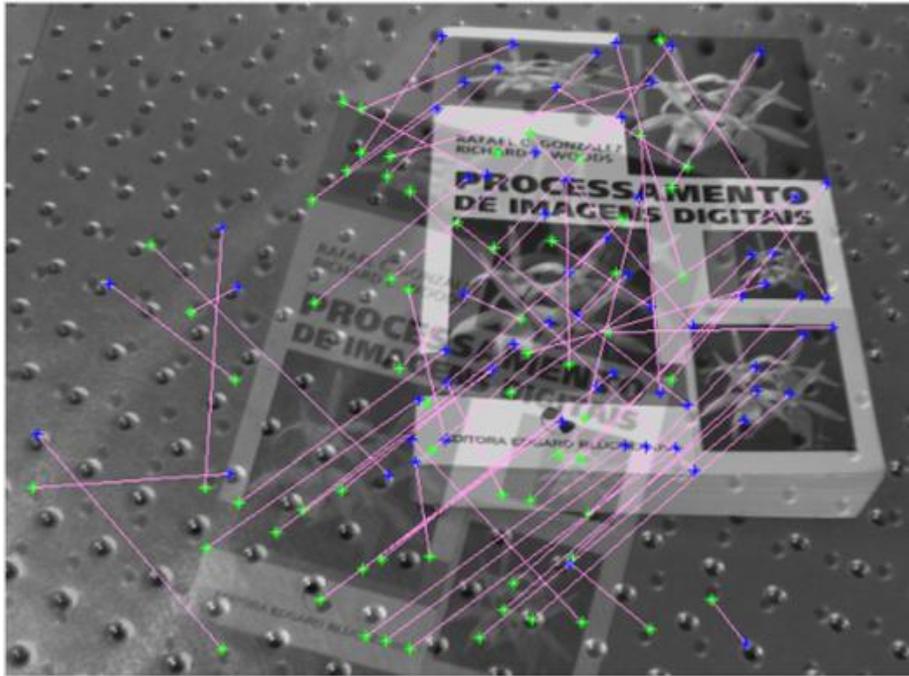


(a)



(b)

Figura 4.5. Casamento de Pontos Par 1 (Ver Tabela 4.1) (a) com só Correlação (b) com o cálculo da Homografia por RANSAC

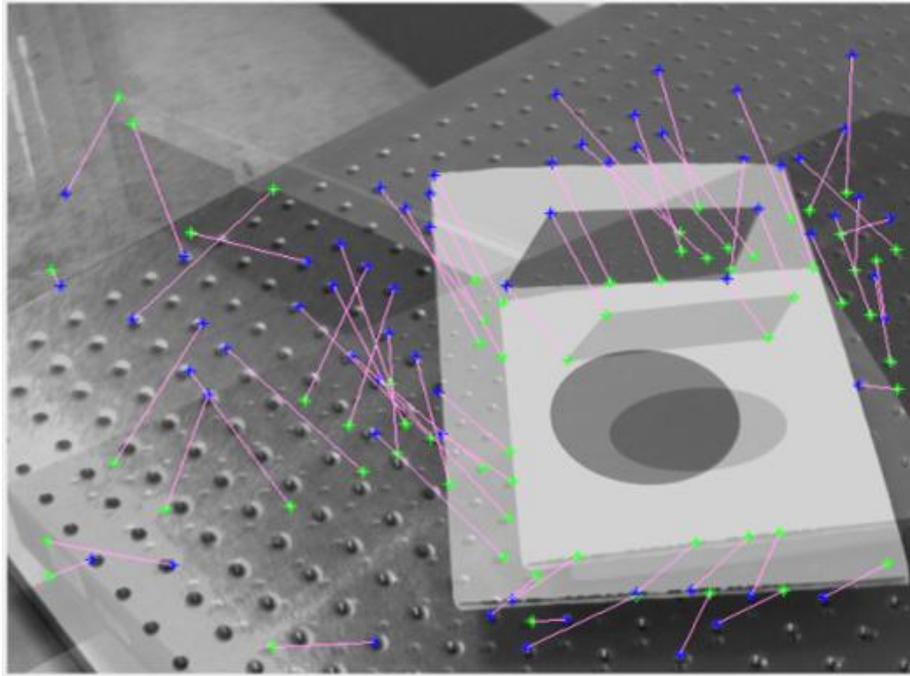


(a)

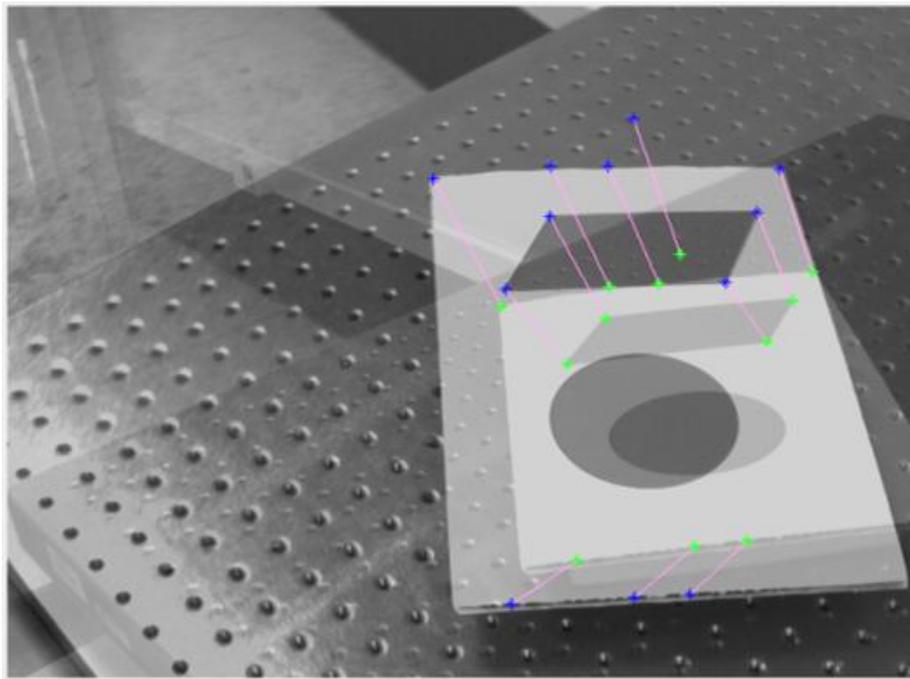


(b)

Figura 4.6. Casamento de Pontos Par 3 (Ver Tabela 4.1.) (a) com só Correlação (b) com o cálculo da Homografia por RANSAC

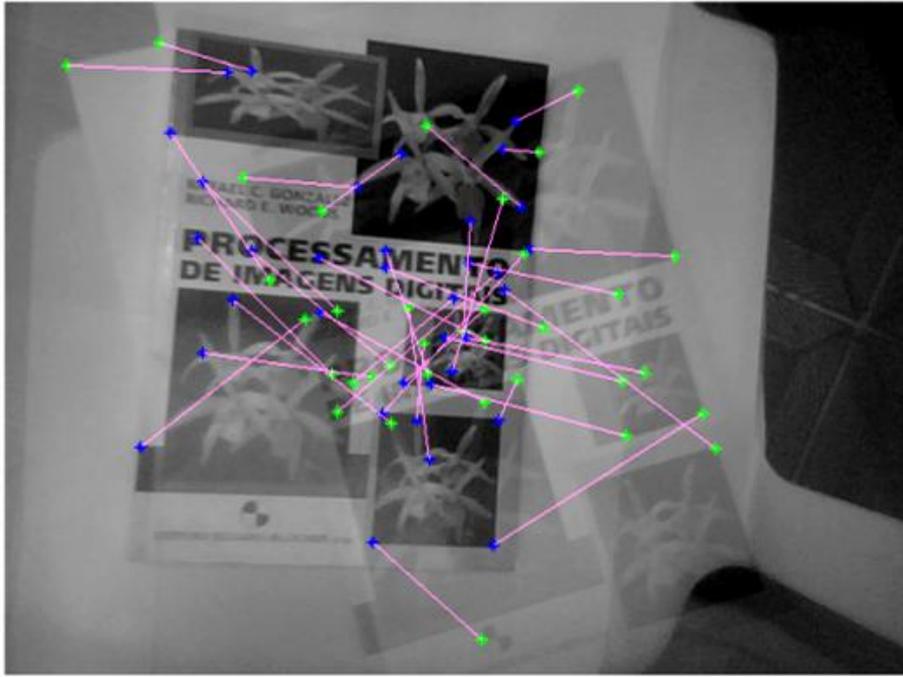


(a)

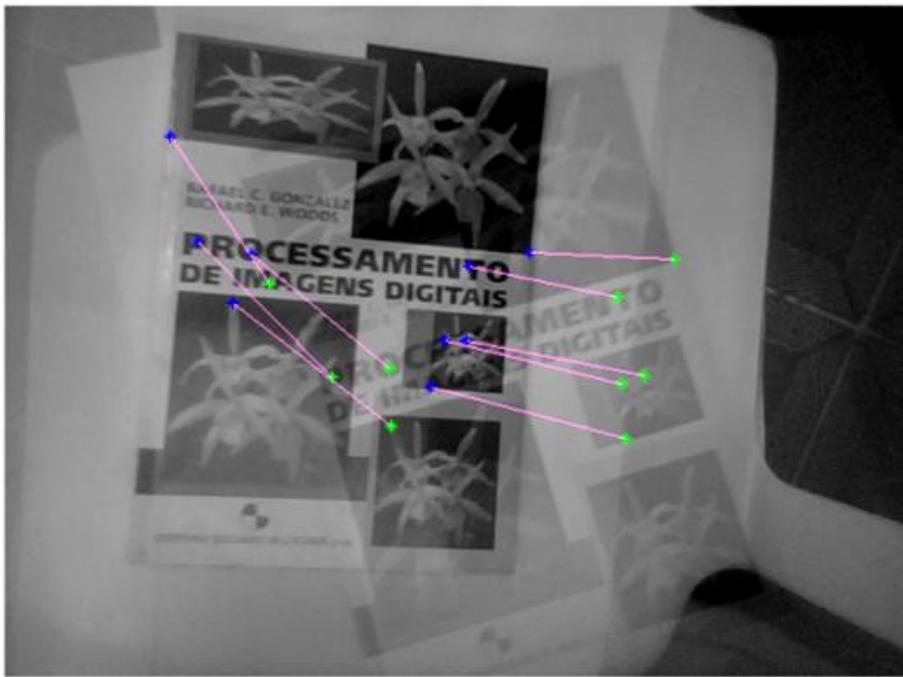


(b)

Figura 4.7. Casamento de Pontos Par 5 (Ver Tabela 4.1.) (a) com só Correlação (b) com o cálculo da Homografia por RANSAC



(a)



(b)

Figura 4.8. Casamento de Pontos Par 10 (Ver Tabela 4.1) (a) com só Correlação (b) com o cálculo da Homografia por RANSAC

Observa-se nas figuras anteriores que a grande maioria das correspondências, fornecidas pelo algoritmo de casamento com a restrição dada pela homografia (Algoritmo 5), são corretas. Resultados similares foram obtidos para todas as imagens de teste da Tabela 4.1, como se apresenta na Tabela 4.5, provando que o algoritmo de casamento com restrição de

Homografia proposto cumpre com a condição de robustez necessária para sua aplicação em sistemas de controle servo-visual.

Tabela 4.5. Resultados do Casamento Robusto

<i>Imagens</i>	<i>Correspondências Encontradas</i>	<i>Correspondências Corretas</i>	<i>Porcentagem de acerto</i>
Par 1	15	15	100%
Par 2	11	10	90,9%
Par 3	29	29	100%
Par 4	37	34	91,9%
Par 5	12	11	91,6%
Par 7	17	17	100%
Par 6	7	7	100%
Par 8	11	11	100%
Par 9	5	5	100%
Par 10	9	9	100%

Os resultados do Casamento Robusto em todas as imagens testadas, embora reduzam em grande medida a quantidades de pontos correspondentes, ainda fornecem um número suficiente de correspondências para calcular o erro na lei de controle que, como visto na seção 2.3, precisa para sua definição quatro ou mais correspondências. De acordo com isto, o algoritmo de casamento robusto foi escolhido como método de obtenção de correspondência de pontos no presente trabalho.

4.1.4- Estimativa da Postura da Câmera

O objetivo da implementação do algoritmo da estimativa da postura da câmera (Algoritmo 6) foi recuperar a posição e orientação da câmera, em função do referencial do objeto observado, a partir de uma imagem, da informação dos parâmetros intrínsecos da câmera e do modelo do objeto. O modelo do objeto, neste caso particular está definido por quatro de seus pontos característicos (quatro quinas) cujas coordenadas no referencial do objeto estão totalmente determinadas, como se apresenta na Figura 4.9.

As distâncias entre os pontos ${}^oP_{mi}$ foram medidas diretamente sobre o objeto, em milímetros. As coordenadas destes pontos são:

$${}^oP_{m1} = [0,0,0,1] \quad {}^oP_{m2} = [153, 0, 0,1] \quad {}^oP_{m3} = [0, 147, 0,1] \quad {}^oP_{m4} = [153, 147, 0,1]$$

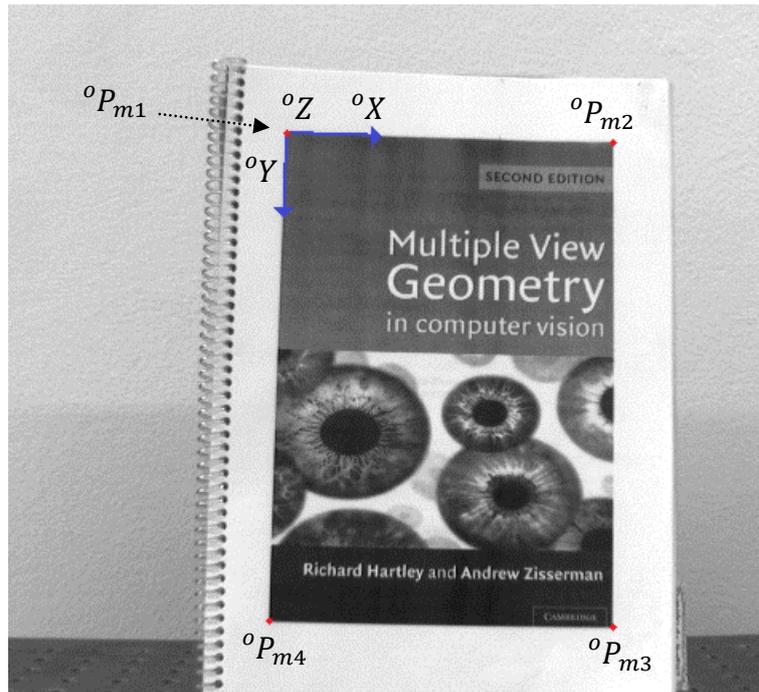


Figura 4.9. Referencial do Objeto

As correspondências destes pontos na imagem foram obtidas manualmente para evitar erros nas etapas posteriores do cálculo da postura. Já para uma aplicação real, deve-se garantir que os pontos observados na imagem e os pontos do modelo tenham uma relação inequívoca que permita uma única correspondência, para o qual pode-se implementar o casamento robusto utilizando a matriz de homografia.

As respostas fornecidas pelo programa são as matrizes de rotação e translação da câmera, com respeito ao referencial do objeto, e, com essa informação, é possível calcular a matriz de projeção para cada uma das imagens utilizando a equação (2.15). A matriz de projeção P é a que permite testar o desempenho do processo de cálculo da postura. Da equação (2.15), repetida aqui, sabe-se que:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

, onde $[X Y Z]$ representam as coordenadas de um ponto no espaço, neste caso no referencial do objeto, e $[u v]$ representam sua projeção em píxeis sobre a imagem. Desta

forma conhecendo a matriz \mathbf{P} , é possível projetar qualquer ponto sobre a imagem cujas coordenadas no referencial do objeto sejam conhecidas.

Para comprovar a premissa anterior, planejou-se modelar virtualmente um cubo de dimensões conhecidas (com as coordenadas de seus vértices) que fosse localizado de forma que um de seus vértices coincidissem com a origem do sistema de coordenadas do objeto e que sua face frontal sempre estivesse na direção positiva de \mathbf{Z} . Como se apresenta na Figura 4., a cada movimento da câmera, uma projeção P é calculada e, aplicando esta projeção às coordenadas do cubo, este deve-se movimentar junto com o objeto na posição em que foi localizado inicialmente. O cubo modelado tem um comprimento de aresta de 50 mm e as coordenadas de seus vértices no referencial do objeto se mostram na Tabela 4.6.

Tabela 4.6. Coordenadas dos oito vértices do cubo virtual no referencial do objeto

${}^oP_{c1} = [0,0,0,1]$	${}^oP_{c2} = [50,0,0,1]$	${}^oP_{c3} = [0,50,0,1]$	${}^oP_{c4} = [50,50,0,1]$
${}^oP_{c5} = [0,0,50,1]$	${}^oP_{c6} = [50,0,50,1]$	${}^oP_{c7} = [0,50,50,1]$	${}^oP_{c8} = [50,50,50,1]$

Os resultados obtidos para duas câmeras diferentes se apresentam nas Figura 4 e 4.12, a única diferença entre as duas câmeras é a matriz de calibração. Os resultados mostram que o comportamento do algoritmo cumpre com sua função de calcular corretamente a matriz de projeção de cada uma das câmeras e, portanto, de obter sua posição e orientação em função do referencial do objeto.

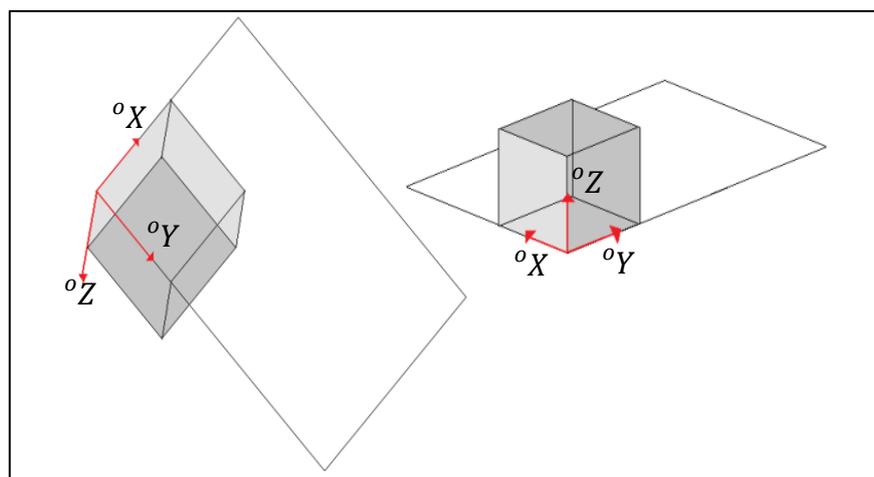
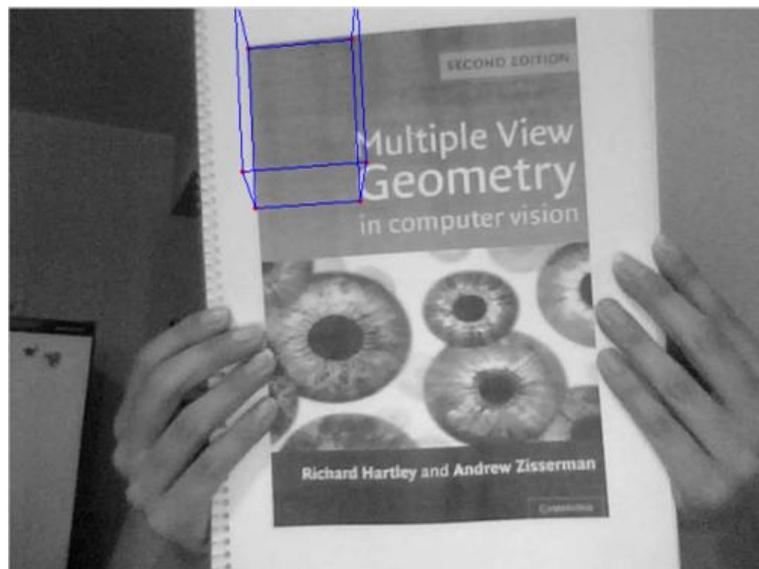
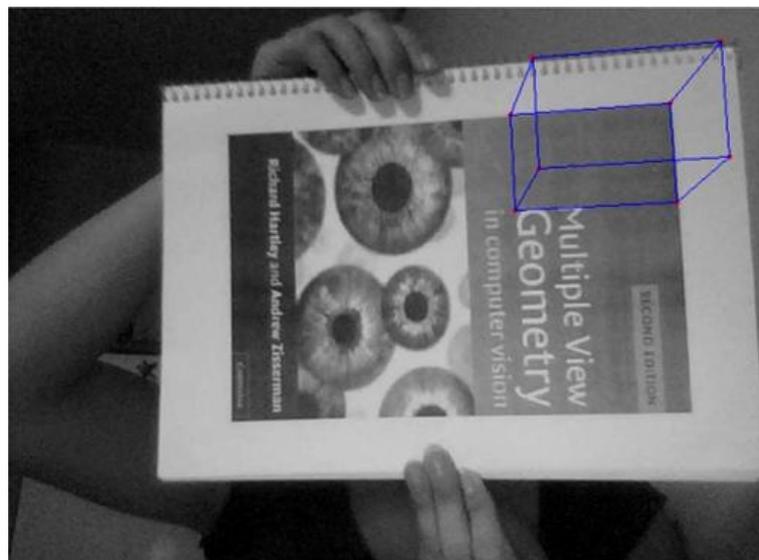


Figura 4.10. Comportamento desejado do movimento do Cubo Virtual

A matriz de projeção da câmera em cada imagem, obtida a partir do Algoritmo 6, foi aplicada diretamente sobre as coordenadas dos vértices do cubo virtual (Tabela 4.6) sem nenhuma informação adicional além das coordenadas destes pontos na imagem e no objeto, e a matriz de calibração da câmera. Desta forma, tendo a informação da câmera e um conjunto de correspondências, entre pontos em coordenadas do objeto e pontos em coordenadas da imagem, é possível obter a postura (orientação e posição) da câmera em função ao referencial do objeto.

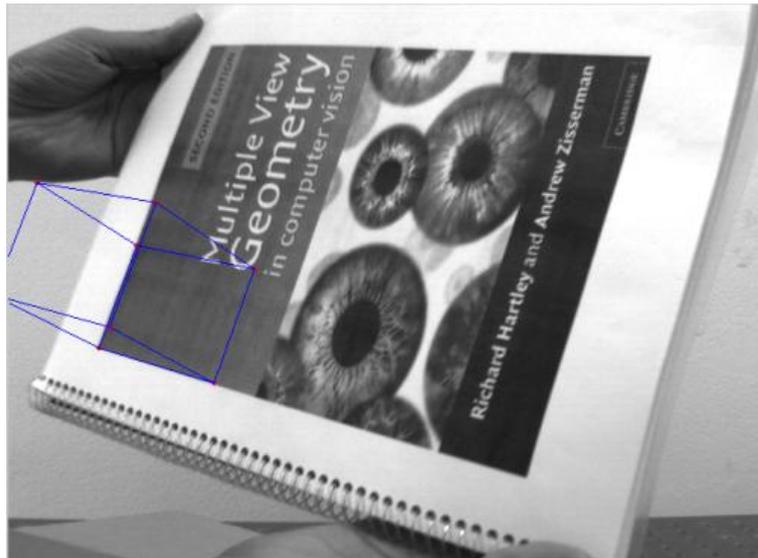


(a)

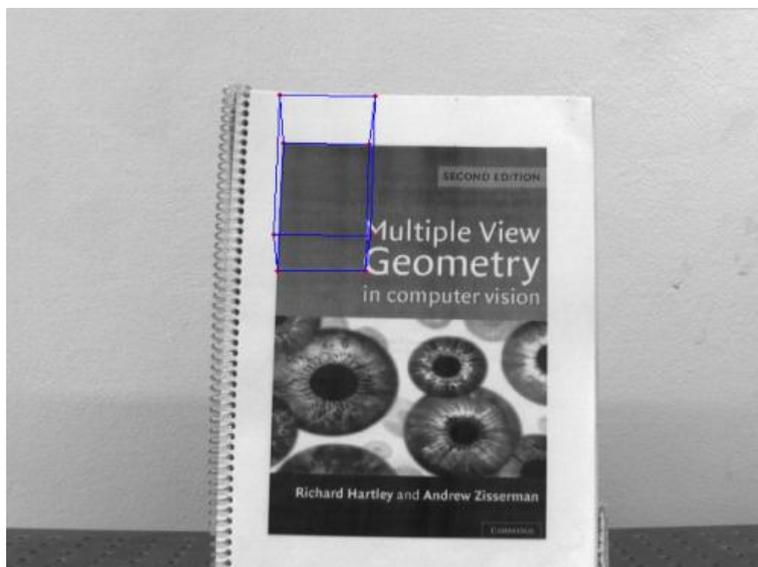


(b)

Figura 4.11. Testes do Cálculo da Postura da Webcam com o objeto em duas posturas diferentes



(a)



(b)

Figura 4.12. Testes do Cálculo da Postura da Câmera CCD com o objeto em duas posturas diferentes

Os resultados do cálculo da postura para os exemplos apresentados nas Figuras 4.11 e 4.12 se apresentam na Tabela 4.7.

Uma vez obtida a validação qualitativa do algoritmo de recuperação da postura (Figuras 4.11 e 4.12), optou-se por complementá-la com dados quantitativos, para o qual foi feito um teste com movimentos controlados da câmera com respeito a um objeto fixo, cujo modelo fosse conhecido.

Tabela 4.7. Resultados do algoritmo de cálculo da postura das imagens das Figuras 4.11 e 4.12

<i>Parâmetros Intrínsecos K</i>	<i>Imagem</i>	<i>Rotação R da câmera</i>	<i>Translação T da câmera</i>
$\begin{bmatrix} 684.6 & 0 & 314.05 \\ 0 & 684.4 & 196.47 \\ 0 & 0 & 1 \end{bmatrix}$	Figura 4 (a)	$\begin{bmatrix} 0.050 & -1.215 & -0.083 \\ -0.770 & -0.068 & -0.122 \\ 0.096 & 0.116 & -0.934 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 146.6 \\ 20.75 \\ -469.8 \\ 1 \end{bmatrix}$
	Figura 4 (b)	$\begin{bmatrix} 0.785 & 0.051 & -0.233 \\ 0.026 & -1.139 & -0.268 \\ -0.212 & 0.328 & -0.896 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -113.28 \\ 129.83 \\ 452.75 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 935.05 & 0 & 337.8 \\ 0 & 936.64 & 263.4 \\ 0 & 0 & 1 \end{bmatrix}$	Figura 4 (a)	$\begin{bmatrix} 0.758 & -0.265 & -0.335 \\ -0.262 & -0.817 & -0.578 \\ -0.149 & 0.815 & -0.688 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 16.347 \\ 77.159 \\ -394.10 \\ 1 \end{bmatrix}$
	Figura 4 (b)	$\begin{bmatrix} -0.021 & -1.205 & -0.116 \\ -0.785 & 0.030 & 0.006 \\ 0.003 & 0.147 & -0.947 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 121.26 \\ 16.90 \\ -504.40 \\ 1 \end{bmatrix}$

4.1.5- Calibração da Câmera

As matrizes de calibração da câmera foram obtidas mediante o *toolbox* fornecido por Bouguet (Bouguet, 2009), salvo algumas modificações feitas para que se pudesse adaptar ao nosso padrão de calibração composto de pontos (Figura 4.13), a diferença do padrão utilizado no *toolbox* que é de tipo xadrez, a seguir, se apresentam os valores das matrizes de calibração das duas câmeras utilizadas nos testes dos algoritmos de processamento de imagens aqui implementados.

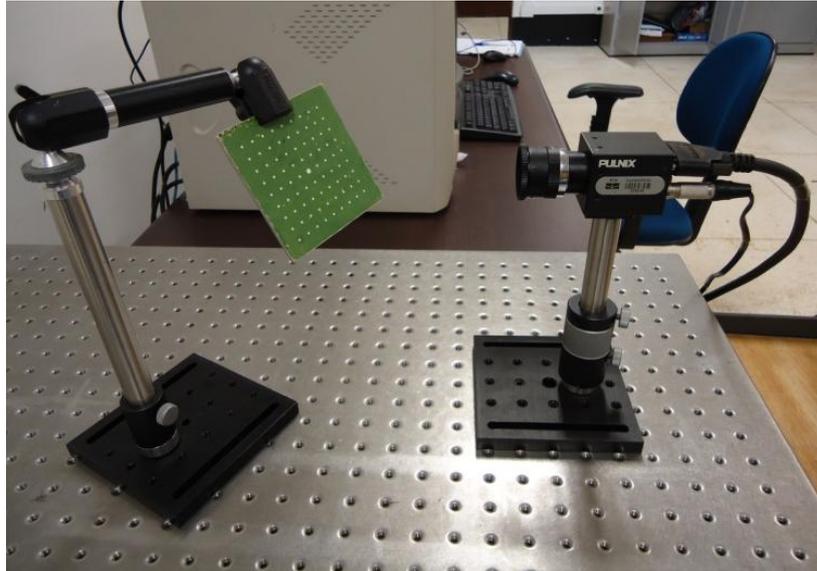


Figura 4.13. Padrão de Calibração

- WEBCam:
$$K_1 = \begin{bmatrix} 684.56565 & 0 & 314.052 \\ 0 & 684.38276 & 196.468 \\ 0 & 0 & 1 \end{bmatrix}$$
- Câmera CCD:
$$K_2 = \begin{bmatrix} 935.05279 & 0 & 337.78432 \\ 0 & 936.63692 & 263.37690 \\ 0 & 0 & 1 \end{bmatrix}$$

4.2- LEI DE CONTROLE

Com o objetivo de testar o correto funcionamento da lei de controle de cada uma das metodologias, o Algoritmo 7 foi implementado e, posteriormente foram comparados os valores de velocidade da câmera. Para encontrar o erro em cada um dos resultados, o algoritmo foi testado com duas imagens iguais, ou seja, com o movimento relativo entre seus pontos nulo, os resultados das velocidades, que em princípio deveriam ser iguais a zero, foram comparados entre si e se apresentam na Tabela 4.8.

Tabela 4.8. Cálculo das Velocidades de Câmera

PBVS		IBVS	
Velocidade $T \left[\frac{cm}{s} \right]; \omega \left[\frac{rad}{s} \right]$	Erro	Velocidade $T \left[\frac{cm}{s} \right]; \omega \left[\frac{rad}{s} \right]$	Erro
$T = \begin{bmatrix} -0.0042 \\ -0.0743 \\ 0.0110 \\ 0.0001 \end{bmatrix}$ $\omega = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$e_T = 0.0752$ $e_\omega \approx 0$	$T = \begin{bmatrix} 0.0039 \\ -0.0006 \\ 0.0010 \\ -0.0008 \end{bmatrix}$ $\omega = \begin{bmatrix} -0.0039 \\ -0.0020 \end{bmatrix}$	$e_T = 0.0041$ $e_\omega = 0.0045$
$T = \begin{bmatrix} -0.0169 \\ 0.1072 \\ 0.0262 \\ 0.0003 \end{bmatrix}$ $\omega = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$e_T = 0.116$ $e_\omega \approx 0$	$T = \begin{bmatrix} -0.0251 \\ -0.0221 \\ 0.0002 \\ 0.0218 \end{bmatrix}$ $\omega = \begin{bmatrix} 0.0252 \\ 0.0001 \end{bmatrix}$	$e_T = 0.0334$ $e_\omega = 0.0333$
$T = \begin{bmatrix} -0.0143 \\ 0.0798 \\ -0.0464 \\ 0.0002 \end{bmatrix}$ $\omega = \begin{bmatrix} -0.0001 \\ 0 \end{bmatrix}$	$e_T = 0.0934$ $e_\omega \approx 0$	$T = \begin{bmatrix} -0.0199 \\ -0.0101 \\ 0.0052 \\ 0.0104 \end{bmatrix}$ $\omega = \begin{bmatrix} 0.0199 \\ -0.0024 \end{bmatrix}$	$e_T = 0.0229$ $e_\omega = 0.0226$
$T = \begin{bmatrix} -0.0139 \\ -0.2765 \\ -0.0227 \\ 0.0007 \end{bmatrix}$ $\omega = \begin{bmatrix} -0.0001 \\ 0 \end{bmatrix}$	$e_T = 0.2778$ $e_\omega \approx 0$	$T = \begin{bmatrix} 0.0834 \\ 0.0167 \\ 0.0150 \\ 0.0177 \end{bmatrix}$ $\omega = \begin{bmatrix} -0.0799 \\ -0.0073 \end{bmatrix}$	$e_T = 0.0864$ $e_\omega = 0.0822$
$T = \begin{bmatrix} -0.0054 \\ 0.0190 \\ -0.0111 \\ -0.0001 \end{bmatrix}$ $\omega = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$e_T = 0.0227$ $e_\omega \approx 0$	$T = \begin{bmatrix} -0.0108 \\ 0.0089 \\ 0.0025 \\ 0.0092 \end{bmatrix}$ $\omega = \begin{bmatrix} 0.0100 \\ 0.0039 \end{bmatrix}$	$e_T = 0.0142$ $e_\omega = 0.0141$
$T = \begin{bmatrix} -0.0152 \\ -0.0266 \\ -0.0353 \\ 0.0001 \end{bmatrix}$ $\omega = \begin{bmatrix} -0.0001 \\ 0.0001 \end{bmatrix}$	$e_T = 0.0467$ $e_\omega \approx 0$	$T = \begin{bmatrix} 0.0230 \\ 0.0215 \\ 0.0031 \\ 0.0204 \end{bmatrix}$ $\omega = \begin{bmatrix} -0.0231 \\ -0.0040 \end{bmatrix}$	$e_T = 0.0316$ $e_\omega = 0.0311$
$T = \begin{bmatrix} 0.0618 \\ -0.0056 \\ 0.0320 \\ 0 \end{bmatrix}$ $\omega = \begin{bmatrix} 0 \\ 0.0002 \\ 0 \end{bmatrix}$	$e_T = 0.0698$ $e_\omega \approx 0$	$T = \begin{bmatrix} -0.0033 \\ -0.0261 \\ 0.0009 \\ -0.0259 \end{bmatrix}$ $\omega = \begin{bmatrix} 0.0023 \\ -0.0006 \end{bmatrix}$	$e_T = 0.0263$ $e_\omega = 0.0260$
$T = \begin{bmatrix} -0.0300 \\ 0.0064 \\ -0.0490 \\ -0.0001 \end{bmatrix}$ $\omega = \begin{bmatrix} -0.0001 \\ -0.0002 \\ 0 \end{bmatrix}$	$e_T = 0.0578$ $e_\omega = 0$	$T = \begin{bmatrix} 0.0027 \\ 0.0331 \\ 0.0081 \\ 0.0327 \end{bmatrix}$ $\omega = \begin{bmatrix} -0.0037 \\ 0.0047 \end{bmatrix}$	$e_T = 0.0342$ $e_\omega = 0.0332$

Os resultados anteriores mostram que o cálculo da velocidade da câmera apresenta erros que aparentemente não são muito significativos em cenas onde a precisão não seja muito alta. Porém, estes resultados foram obtidos a partir de somente duas imagens e sem entrar no laço de realimentação do controle. Contudo, os algoritmos para o cálculo da velocidade da câmera são suficientes para se implementar dentro dos sistemas de controle visual propostos.

5 - SIMULAÇÃO

Com o fim de testar o comportamento das duas técnicas de controle servo visual sob estudo, foram desenvolvidas simulações de cada uma das abordagens na plataforma *simulink*. Para tal fim, utilizou-se a informação real da matriz de calibração da câmera CCD e as coordenadas de quatro pontos nas imagens estudadas no capítulo anterior. Os três componentes do processo de controle visual, a análise das imagens, a lei de controle e o movimento do sistema robô-câmera foram implementados de tal forma que seu desempenho fora satisfatório para atingir os dois casos de controle. O esquema do sistema simulado apresenta-se na Figura 5.1, a única variação entre os dois modelos de controle servo visual estudados nesta pesquisa, está no bloco correspondente ao controle.

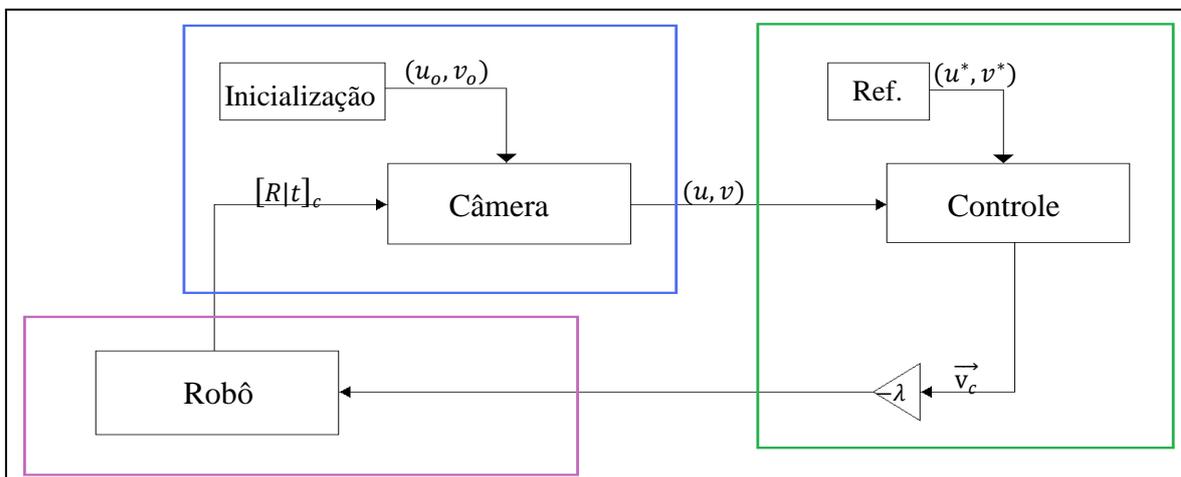


Figura 5.1. Diagrama do Sistema Simulado. O sub-índice c refere-se à câmera

Cada um dos blocos da Figura 5.1. conta com diferentes componentes que reproduzem o comportamento dos elementos do sistema. Cabe ressaltar neste ponto que sendo estes modelos ideais, estão livres de perturbações externas como erros nas coordenadas dos pontos ou erros na execução do movimento no manipulador. A seguir será descrito cada um dos componentes desenvolvidos na simulação.

5.1- INICIALIZAÇÃO E MODELAGEM DA CÂMERA

Como foi mencionado anteriormente, o objetivo do controle servo-visual, implementado neste trabalho, é controlar a posição de um manipulador robótico de seis graus de liberdade em relação a um objeto alvo, baseando-se na informação visual do objeto, fornecida por

uma câmera embarcada no elemento terminal do manipulador. O objeto alvo caracterizado por ter uma superfície plana, encontra-se fixo no espaço, assim como se encontra fixa a base do manipulador robótico. Na Figura 5.2. são apresentados os elementos envolvidos no sistema e as relações entre seus referenciais.

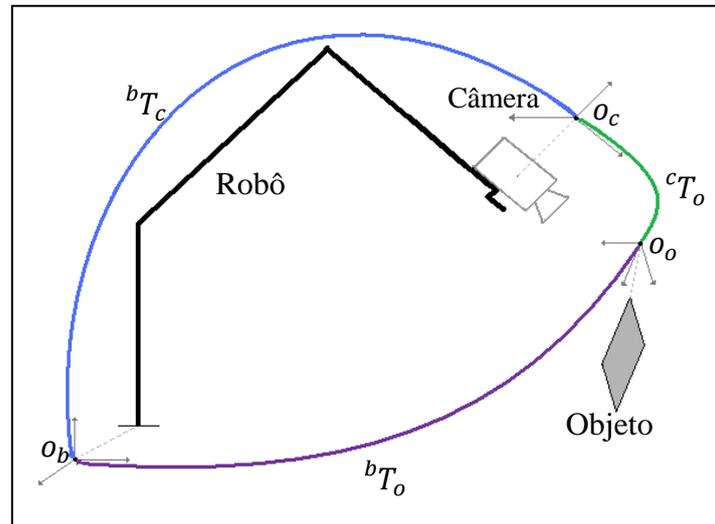


Figura 5.2. Referenciais associados ao sistema (Robô, Câmera e Objeto Alvo) com suas respectivas transformações

A inicialização do sistema tem duas tarefas específicas; a primeira é determinar a transformação entre o referencial do objeto e a base do manipulador, esta relação se mantém constante durante todo o processo; e a segunda é estimar, a partir das informações iniciais, as coordenadas dos pontos característicos do objeto, em coordenadas da imagem, vistos desde a posição inicial da câmera (esta posição inicial deve ser diferente da posição alvo). As informações fornecidas ao sistema na sua inicialização são:

- Imagem do objeto na posição alvo.
- Coordenadas homogêneas, no referencial do objeto, dos quatro pontos característicos do objeto que determinam o modelo do objeto.
- Valores das coordenadas de junta na posição inicial.
- Parâmetros de D-H do manipulador.
- Matriz de calibração da câmera **K**.

Tendo em conta que tanto o objeto como a base do manipulador se encontram fixos, sua relação geométrica, como mostra a Figura 5.2, é dada em qualquer ponto por:

$${}^bT_o = {}^bT_c {}^cT_o \quad (5.1)$$

Escolheu-se a posição alvo para determinar estes valores. A transformação entre o objeto e a câmera, cT_o , obteve-se a partir da relação homográfica entre o objeto e sua imagem nesta posição (seção 3.2.4-). A relação entre o elemento terminal e a base do robô, bT_c , obteve-se a partir da cinemática direta do manipulador (seção 3.4.1-). O diagrama do sistema implementado é apresentado a seguir.

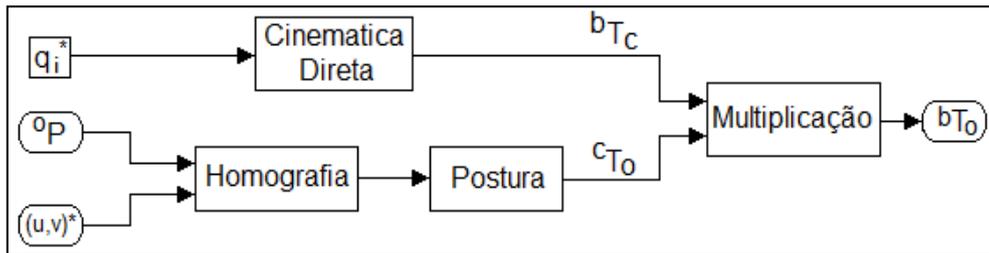


Figura 5.3. Inicialização do sistema de controle visual

Na Figura 5.3. q_i^* representa as coordenadas de junta na posição alvo, oP é o modelo do objeto, $(u, v)^*$ são as coordenadas dos pontos na imagem adquirida na posição alvo. Estes dados são necessários nesta etapa da simulação porque não existe uma câmera real que forneça a imagem atual e a imagem alvo. Estas devem ser geradas a partir da modelagem da câmera baseada no modelo de projeção central (seção 2.1.3-) mediante a relação:

$$(u, v, 1)^T = \mathbf{K} {}^cP \quad (5.2)$$

, onde \mathbf{K} é a matriz de calibração da câmera e cP são as coordenadas dos pontos (o modelo oP) nas coordenadas da câmera.

$${}^cP = {}^cT_o {}^oP \quad (5.3)$$

$${}^cT_o = ({}^bT_c)^{-1} {}^bT_o \quad (5.4)$$

A transformação bT_o da equação (5.4) é conhecida de (5.1), e a transformação bT_c se obtém da cinemática direta do manipulador conhecendo os valores de junta na posição inicial q_{i_0} como apresenta a Figura 5.4.

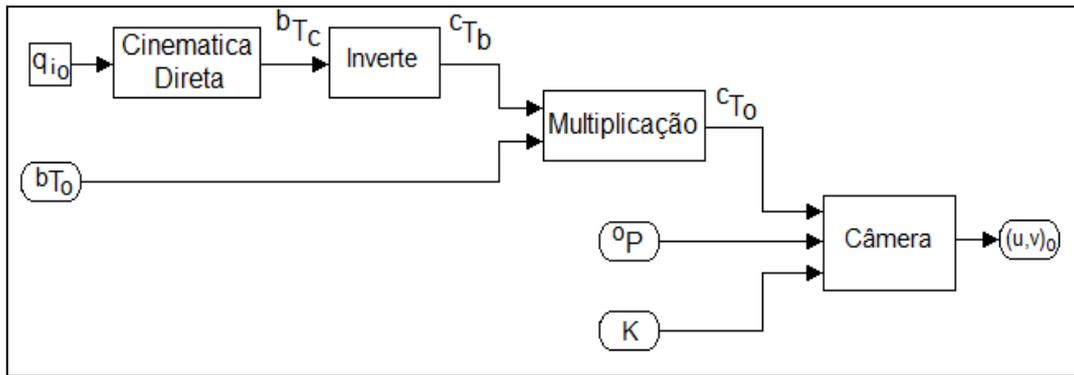


Figura 5.4. Obtenção da imagem atual dos pontos do objeto

5.2- ROBÔ

A informação fornecida pelo sistema de controle é a velocidade que deve atingir o elemento terminal do robô, e por isso é necessário implementar uma etapa de transformação dessa velocidade em parâmetros das juntas e, desta forma, mediante o modelo da cinemática direta do manipulador, encontrar a postura desejada do elemento terminal do robô e da câmera a cada iteração. Assim, no esquema apresentado na Figura 5.5, determina-se a postura do elemento terminal em relação à base do manipulador em função da velocidade do elemento terminal, tendo em conta a equação (3.25):

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{q}_n \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad (5.5)$$

onde $[\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\gamma}]^T$ representam as componentes de translação (três) e angulares (três) da velocidade do elemento terminal respectivamente e \mathbf{J}^{-1} é a matriz Jacobiana inversa do manipulador. Integrando a equação (5.5) obtém-se os parâmetros de junta para a velocidade especificada.

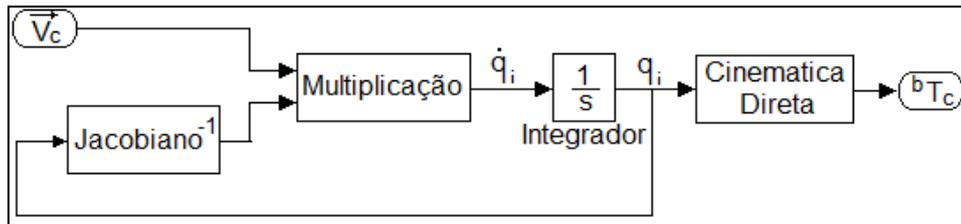


Figura 5.5. Determinação da postura do elemento terminal a partir de sua velocidade

Uma vez obtida a postura do elemento terminal, e portanto da câmera, é adquirida uma nova imagem do alvo nesta nova posição, utilizando o modelo da câmera. Esta nova imagem do alvo (posição atual) junto com a imagem do alvo na posição desejada são fornecidas ao bloco de controle que procura achar, a partir das diferenças entre estas imagens, a velocidade do elemento terminal como se detalha a seguir.

5.3- CONTROLE

A etapa de controle é a encarregada de calcular a velocidade que deve atingir o elemento terminal do robô a partir da comparação entre a imagem do objeto na posição alvo e a imagem atual. O cálculo da velocidade varia entre os dois esquemas de controle servo visual estudados neste trabalho.

5.3.1- Controle Servo Visual baseado em Imagem

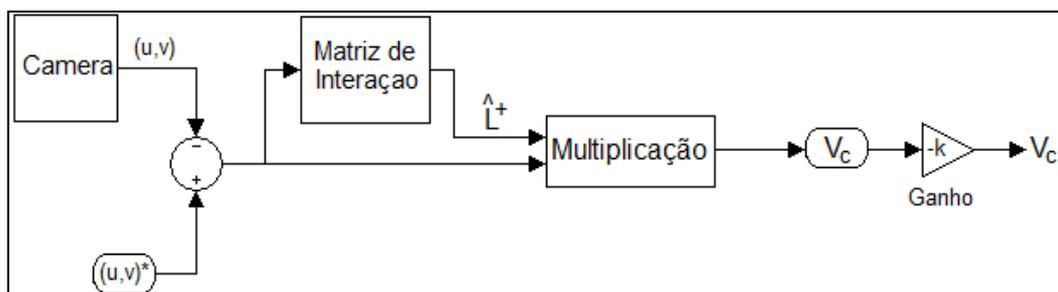


Figura 5.6. Controle Visual Baseado em Imagem

O controle baseado em imagem baseia-se no cálculo da matriz de interação (seção 2.3.4.1-) para encontrar a velocidade da câmera, que neste caso é a mesma velocidade do elemento terminal do manipulador robótico.

Como visto na seção 2.3.4.1-, o valor da matriz de interação depende não só das diferenças nas imagens como também dos valores estimados da profundidade da cena e dos

parâmetros intrínsecos da câmera. Da equação (2.42) tem-se que a velocidade da câmera \mathbf{v}_c está dada por:

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}}_s^+ \mathbf{e} \quad (5.6)$$

, onde \mathbf{e} é o erro entre as posições dos pontos nas duas imagens, $\widehat{\mathbf{L}}_s^+$ é a estimativa da matriz de interação e λ é o valor do ganho. A velocidade da câmera assim obtida é o sinal de realimentação do laço de controle do sistema como se apresenta na Figura 5.1.

5.3.2- Controle Servo Visual baseado em Posição

O cálculo da velocidade no controle visual baseado em posição se faz a partir do conhecimento da postura da câmera a cada iteração com respeito à postura da câmera na posição desejada, de acordo com a relação dada na equação (2.52):

$$\mathbf{v}_c = \begin{bmatrix} T \\ \Omega \end{bmatrix} = \begin{bmatrix} -\lambda \mathbf{R}^T \mathbf{c}^* \mathbf{t}_c \\ -\lambda \theta \mathbf{u} \end{bmatrix} \quad (5.7)$$

O diagrama de blocos da Figura 5.7. resume o cálculo da velocidade da câmera, como visto, este é o sinal de realimentação do laço de controle.

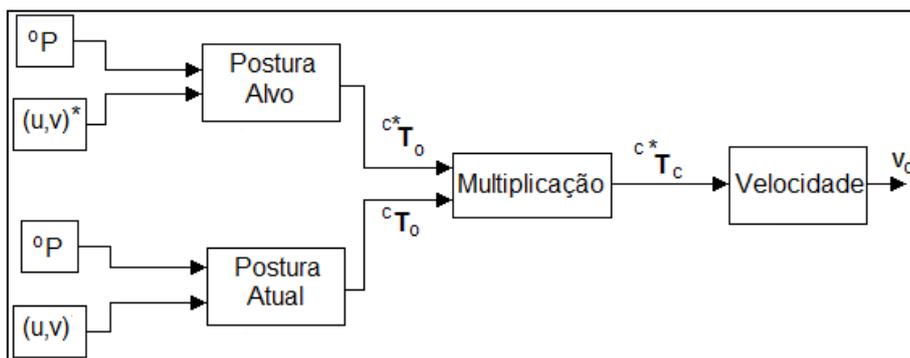


Figura 5.7. Controle Visual Baseado em Posição

5.4- RESULTADOS DE SIMULAÇÃO DO CONTROLE SERVO VISUAL

5.4.1- Controle Visual Baseado em Imagem

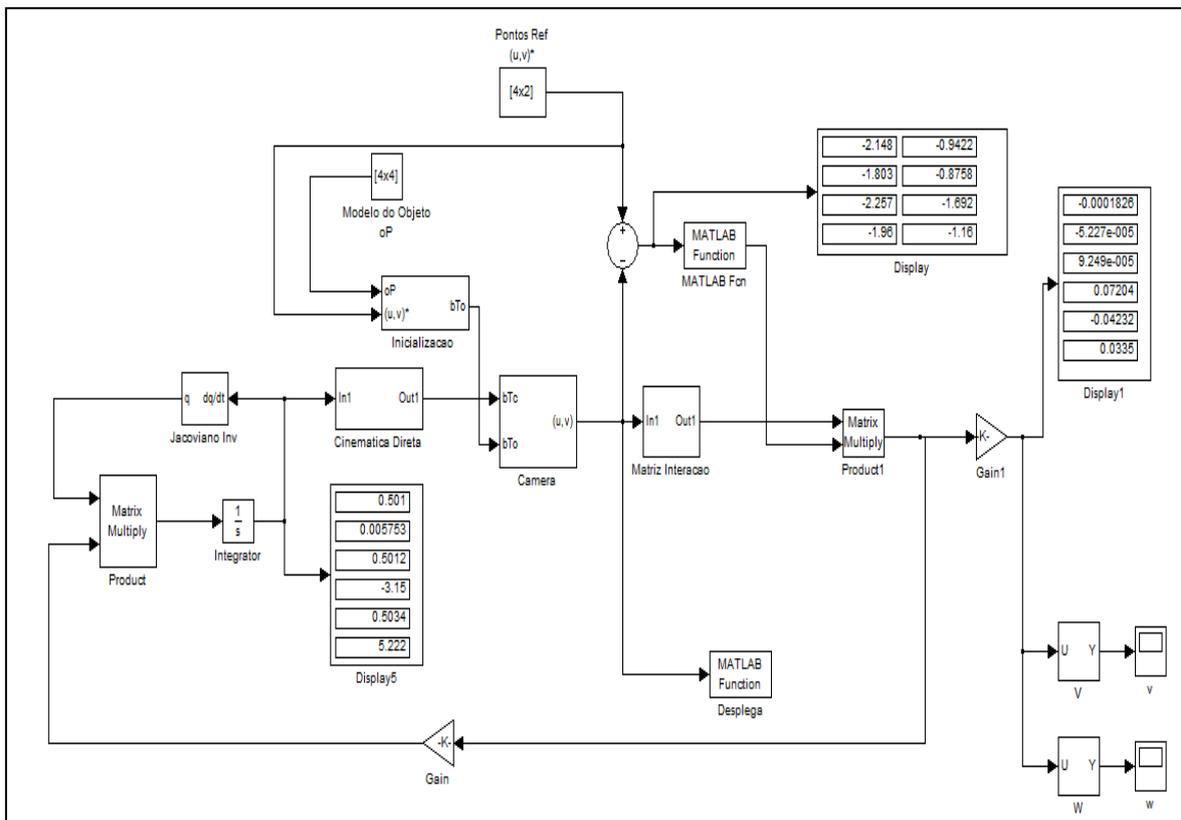
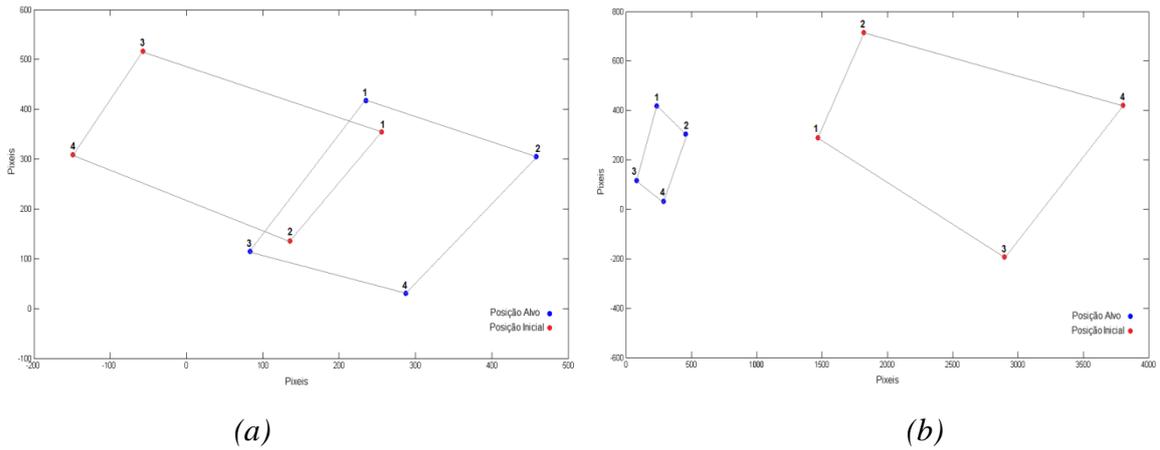


Figura 5.8. Modelo em Simulink do IBVS

Para analisar o desenvolvimento do sistema de controle, foi estudado o comportamento de diferentes variáveis envolvidas no modelo em dois casos específicos, isto é, caso 1 e caso 2, apresentados na Figura 5.9. As variáveis analisadas foram:

- A velocidade da câmera;
- As posições dos pontos na imagem;
- As variáveis de junta do manipulador.

Nas Figuras 5.10 e 5.11 são apresentados os resultados da implementação do sistema de controle baseado em imagem para uma posição inicial do objeto próxima à posição alvo (caso 1). A Figura 5.10 mostra a evolução da trajetória dos pontos desde sua posição inicial até atingir a posição alvo, destaca-se que as coordenadas dos pontos em pixels são dadas com respeito à posição final da câmera.



(a) (b)
 Figura 5.9. Casos de Estudo (a) Caso 1: Posição Inicial próxima à Posição Alvo
 (b) Caso 2: Posição Inicial afastada da Posição Alvo

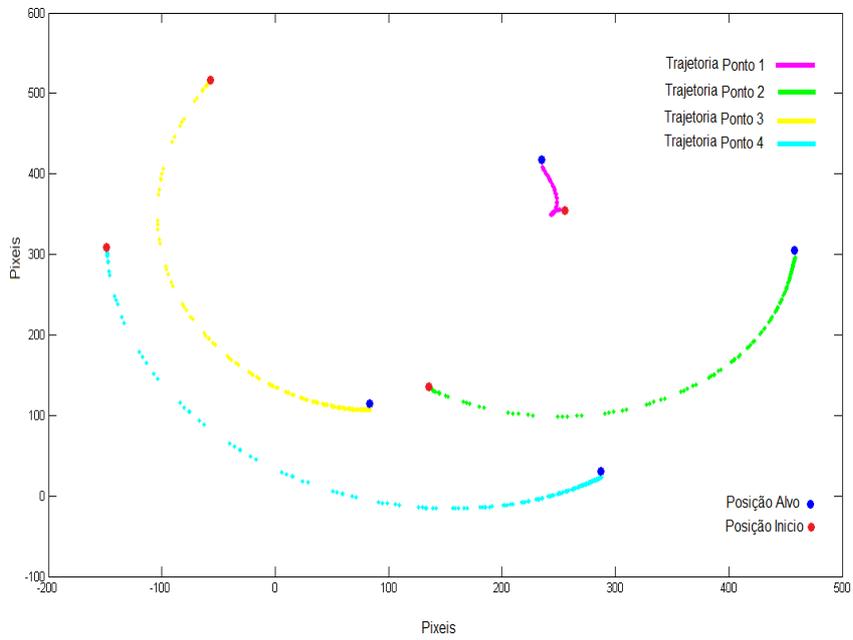


Figura 5.10. Evolução da Trajetória dos pontos durante o controle IBVS (caso 1)

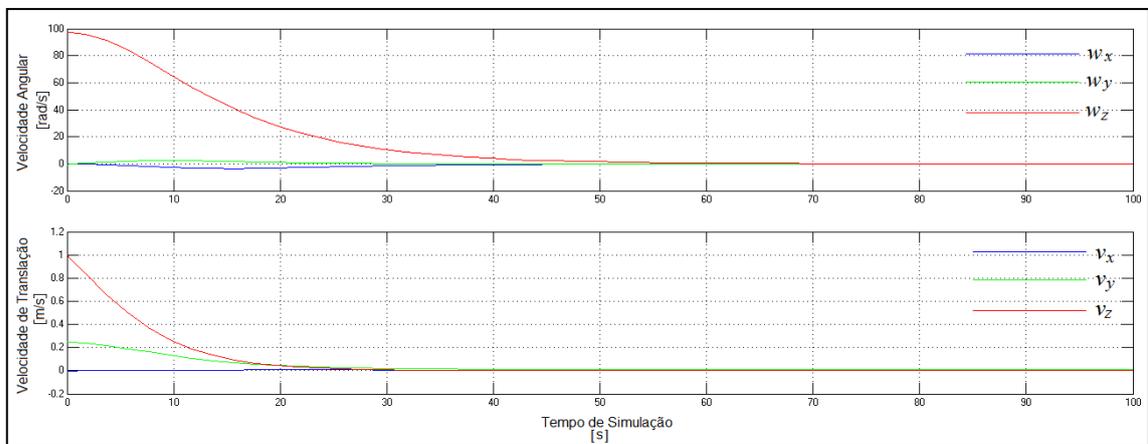


Figura 5.11. Evolução da Velocidade da câmera durante o controle IBVS (caso 1)

Neste caso particular, observa-se como os pontos inicialmente (em vermelho) atingem a posição alvo (em azul), ou seja, o controle cumpre com o seu objetivo. Como esperado, a velocidade da câmera, apresentada na Figura 5.11, encontra o valor de convergência zero durante o tempo de execução do controle.

No exemplo apresentado nas Figuras 5.12 e 5.13, as posições iniciais dos pontos encontram-se consideravelmente afastadas da posição alvo (~ 3000 píxeis) (caso 2), porém o controle consegue atingir o objetivo de movimentar o manipulador até a posição desejada. A velocidade da câmera converge ao valor esperado embora o tempo de convergência seja um pouco maior que no caso anterior.

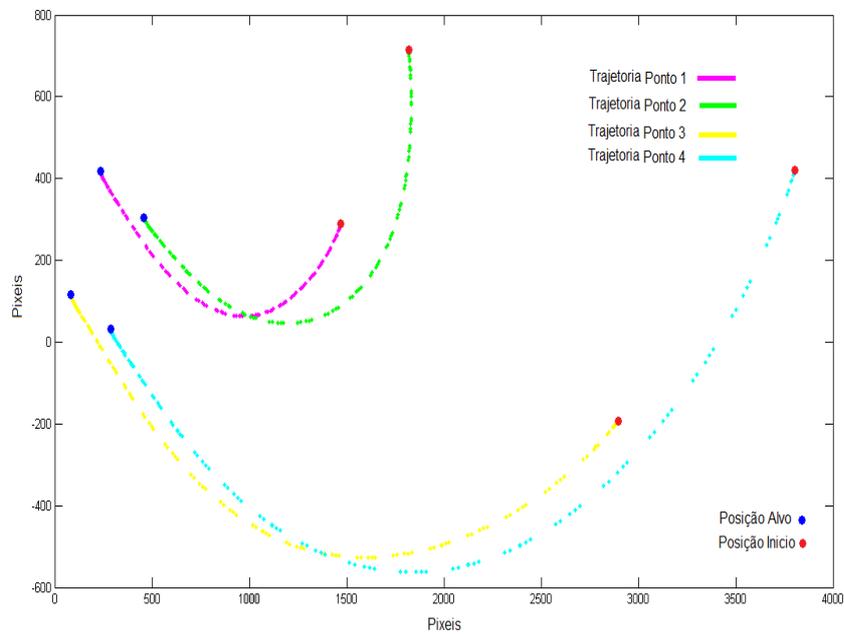


Figura 5.12. Evolução da Trajetória dos pontos durante o controle IBVS (caso 2)

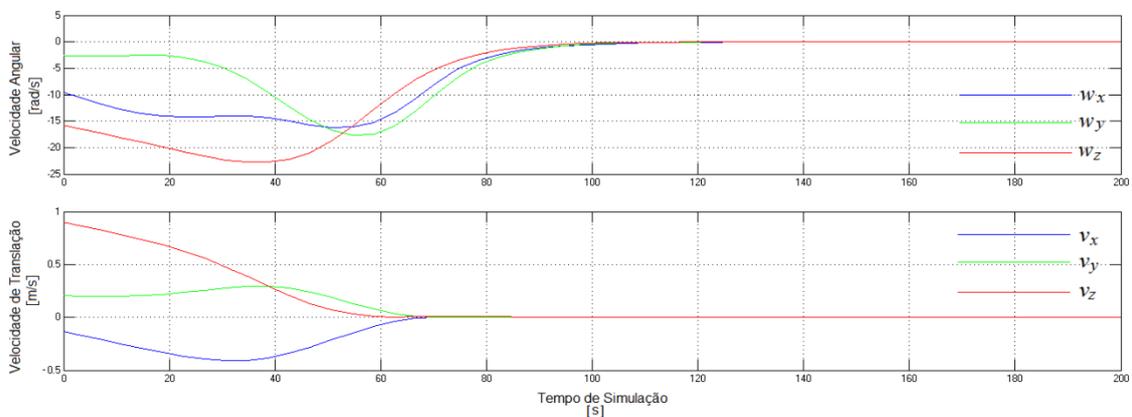


Figura 5.13. Evolução da Velocidade da câmera durante o controle IBVS (caso 2)

5.4.2- Controle Visual Baseado em Posição

Na Figura 5.14. apresenta-se o modelo em Simulink implementado para o controle baseado em posição. O cálculo da velocidade da câmera, o fator diferenciador dos dois esquemas de controle, implica a estimação a cada iteração da postura da câmera, pelo que poder-se-ia pensar que a convergência, se houvesse, tardaria mais tempo. Porém, como apresentam as Figuras 5.15 a 5.18, não só o tempo de convergência nos dois casos estudados é menor, mas também as trajetórias dos pontos apresentam menores curvaturas.

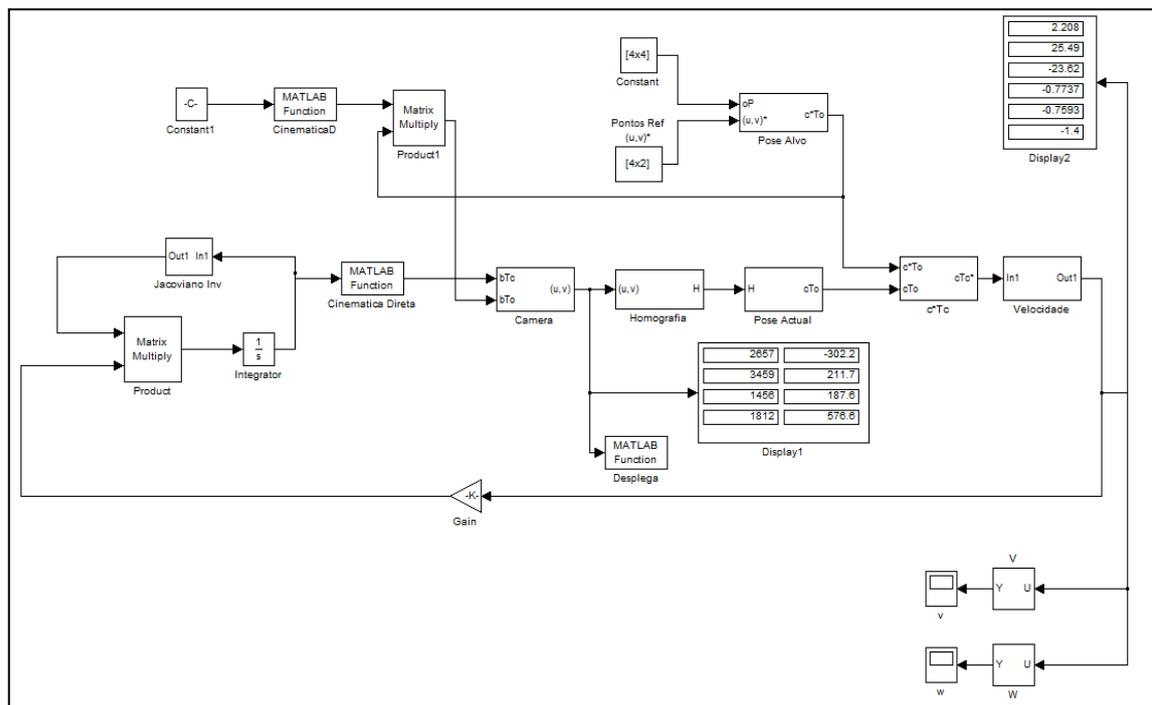


Figura 5.14. Modelo em Simulink do PBVS

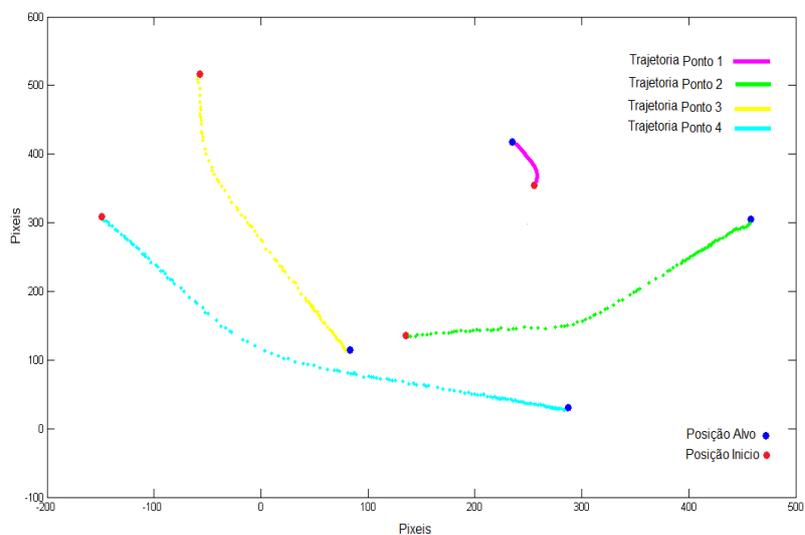


Figura 5.15. Evolução da Trajetória dos pontos durante o controle PBVS (caso 1)

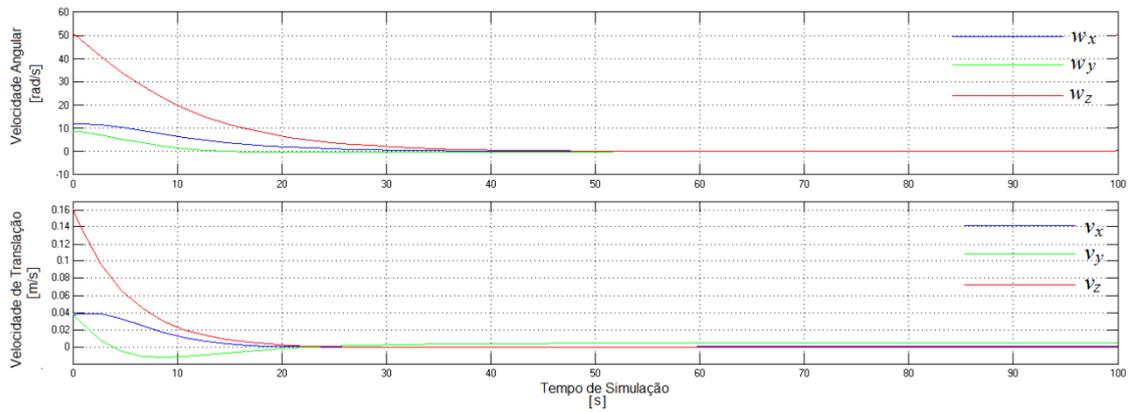


Figura 5.16. Evolução da Velocidade da câmera durante o controle PBVS (caso 1)

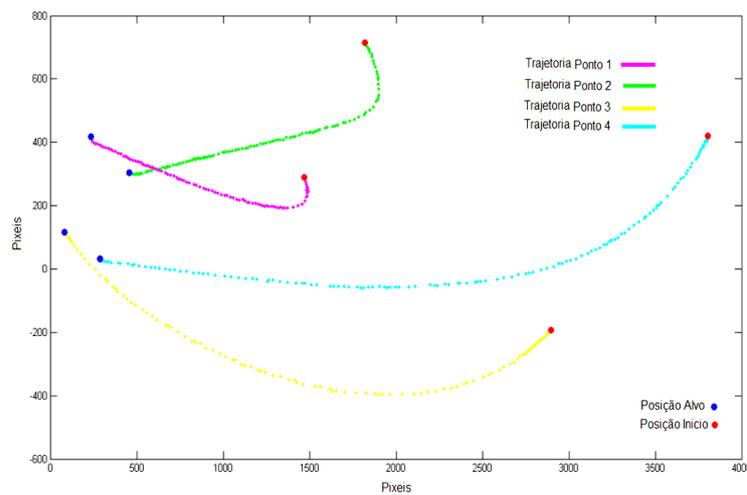


Figura 5.17. Evolução da Trajetória dos pontos durante o controle PBVS (caso 2)

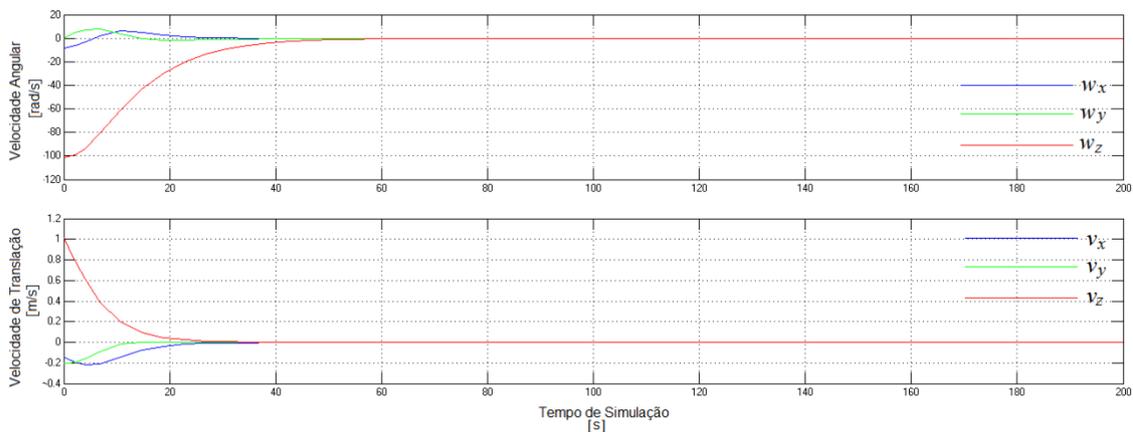


Figura 5.18. Evolução da Velocidade da câmera durante o controle PBVS (caso 2)

Os resultados da simulação apresentados até este ponto mostram que em condições ideais, livres de erros de calibração e de modelagem do objeto, o controle servo visual baseado em posição tem um melhor comportamento, tanto em tempo de execução da tarefa como em

deslocamento relativo da câmera, evitando em maior medida, a perda de informação visual. Este aspecto é importante em situações reais onde o campo de visão da câmera é limitado e movimentos bruscos do robô poderiam levar o objeto para fora do campo visual da câmera. Porém, em ambientes reais não é possível garantir condições livres de erros de calibração ou de modelagem do objeto. Para fazer uma análise mais profunda deste aspecto foram adicionados erros de calibração e do modelo do objeto aos sistemas de controle implementados. Os resultados destes testes são apresentados a seguir.

5.5- ADIÇÃO DE ERROS NO SISTEMA

Para testar o comportamento de cada uma das metodologias implementadas em condições não ideais, foram adicionados diferentes erros nos parâmetros de contorno de forma sistemática.

5.5.1- Erros de Calibração

Foram introduzidos dentro do modelo erros nos parâmetros de calibração da câmera nas etapas do cálculo da matriz de interação para o controle baseado em imagem e no cálculo das posturas no controle baseado em posição. Cada um dos parâmetros foi modificado em 20% do valor real obtido mediante a calibração da câmera (seção). Os resultados são apresentados nas figuras 5.19 e 5.20.

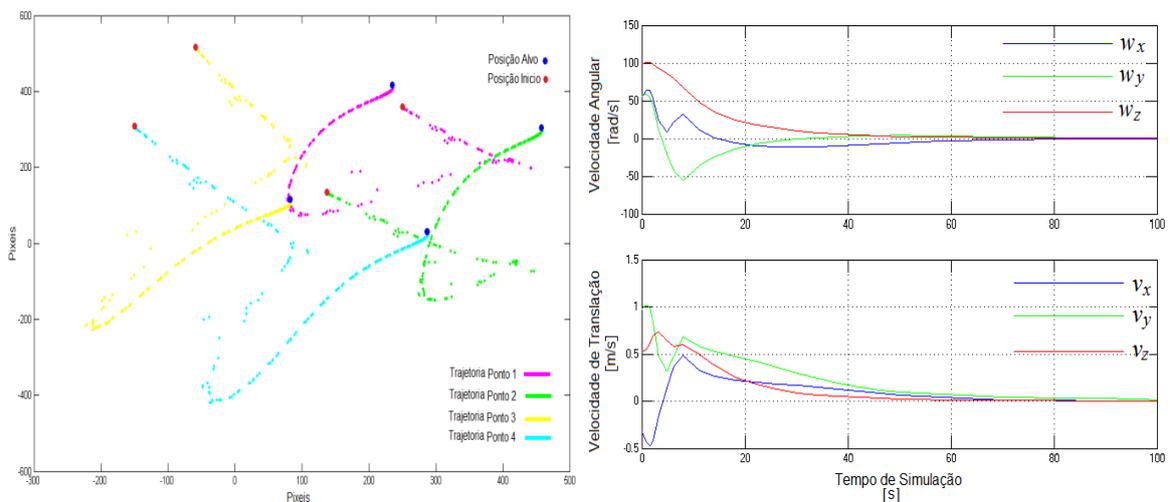


Figura 5.19. Evolução da Trajetória dos pontos e da Velocidade da câmera durante o controle IBVS em presença de erros de calibração do 20% (caso 1)

No controle visual baseado em imagens observa-se que as trajetórias seguidas pelos pontos, na presença de erros de calibração, têm um comportamento errático, mas conseguem atingir a posição desejada. A velocidade apresenta igualmente instabilidade no início da tarefa, mas encontra o ponto de convergência igual a zero embora o tempo convergência seja quase o dobro do empregado em condições ideais. A robustez aos erros de calibração, apresentada neste tipo de controle, deve-se ao fato de utilizar no cálculo da variável de controle a informação adquirida diretamente das imagens a cada iteração o que permite que o sistema corrija sua posição.

Ao contrário do comportamento apresentado no controle baseado em imagem, para caso 1 apresentado na Figura 5.19, o controle baseado em posição não consegue movimentar a câmera ate a posição desejada (Figura 5.20) e, até mesmo, se afasta rapidamente do alvo. Igualmente a velocidade da câmera não converge ao valor zero esperado em todas suas componentes. Este resultado é justificado dado que os parâmetros de calibração são utilizados para calcular a postura dos objetos com respeito à câmera a cada iteração do sistema, de tal forma que poder-se-ia afirmar que o erro de calibração influencia em grande medida o comportamento do sistema de controle.

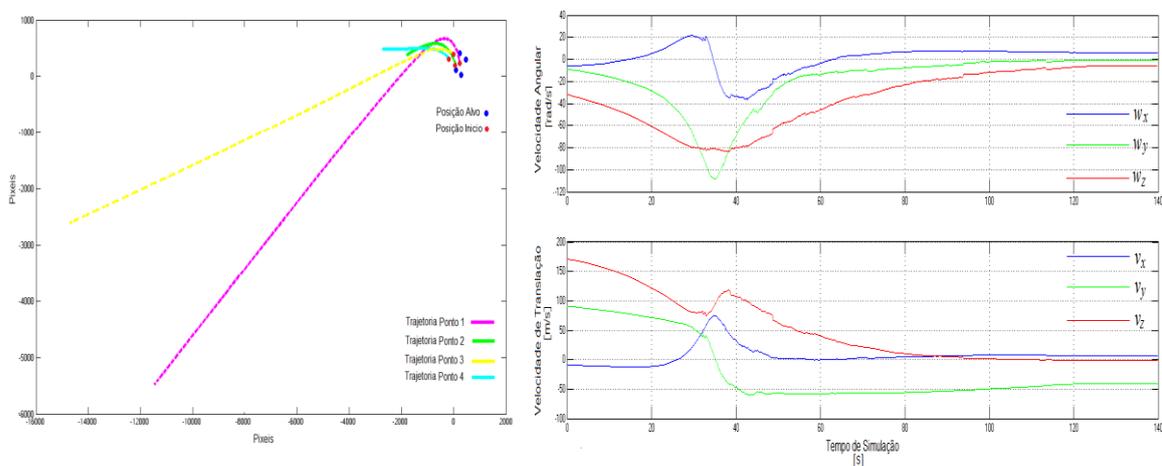


Figura 5.20. Evolução da Trajetória dos pontos e da Velocidade da câmera durante o controle PBVS em presença de erros de calibração do 20% (caso 1)

5.5.2- Erros em Medidas 3D

Outro tipo de erros que podem-se apresentar durante a modelagem do controle visual são os erros nas medições 3D envolvidas no sistema, como a estimativa da profundidade da

cena Z no cálculo da matriz de interação para o controle IBVS ou a descrição do modelo do objeto utilizado no controle PBVS.

5.5.2.1- Erro na estimativa da profundidade de cena Z

A medida da profundidade da cena, Z , é utilizada no cálculo da matriz de interação L_s no controle visual baseado em imagem. As imagens utilizadas durante a etapa de simulação forem adquiridas com um conhecimento aproximado desta medida, e, com o fim de observar o comportamento do sistema, este valor foi alterado em diferentes ordens de magnitude. Inicialmente utilizou-se uma profundidade de valor equivalente ao 10% do valor real e posteriormente o valor foi acrescentado até um 1000% do valor real.

Como se apresenta nas Figuras 5.21 a 5.23, a medida que o valor de Z se afasta de seu valor real a trajetória dos pontos, e por tanto o movimento da câmera, é mais errático, e a convergência da velocidade no valor zero está altamente comprometida assim como o tempo de convergência (a Figura 5.22 mostra um tempo de convergência um pouco maior que no caso observado na Figura 5.21). O caso específico apresentado na Figura 5.23 não atinge os pontos e a velocidade não converge.

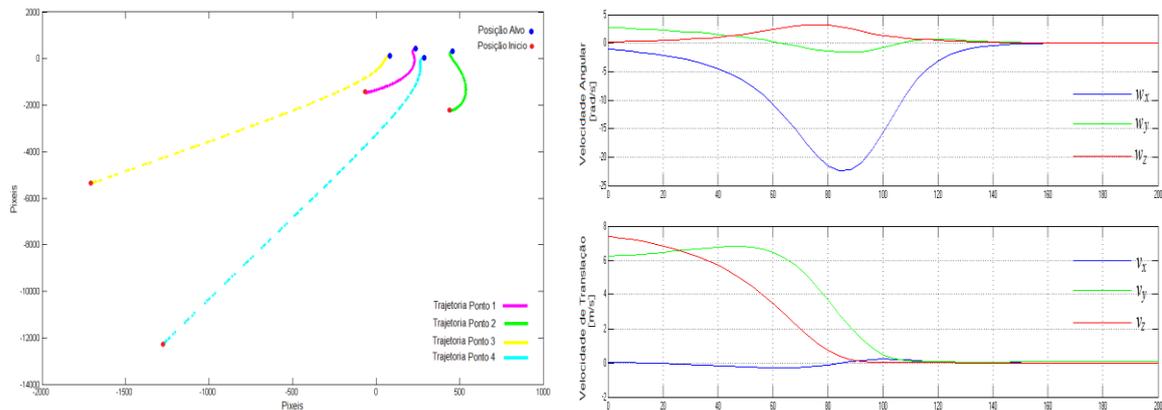


Figura 5.21. Evolução da Trajetória dos pontos e da Velocidade da câmera durante o controle IBVS com valor de profundidade Z real ($Z = 1m$)

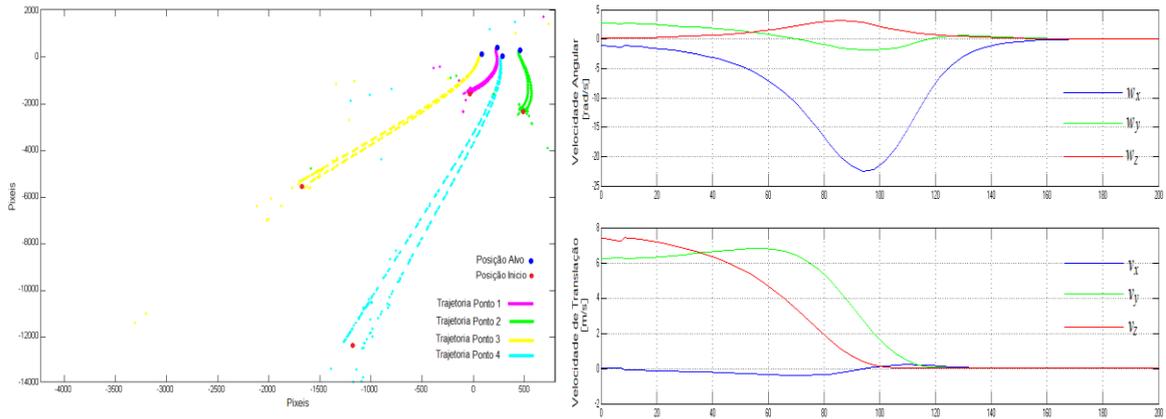


Figura 5.22. Evolução da Trajetória dos pontos e da Velocidade da câmera durante o controle IBVS com valor de profundidade Z modificado ($Z = 0.1m$)

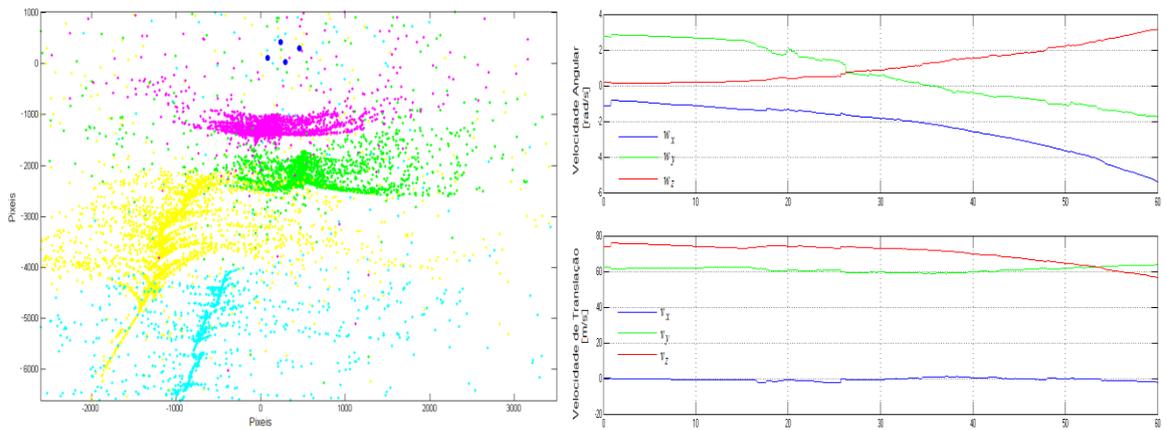


Figura 5.23. Evolução da Trajetória dos pontos e da Velocidade da câmera durante o controle IBVS com valor de profundidade Z modificado ($Z = 10m$)

5.5.2.2- Erro no Modelo do Objeto

O modelo do objeto utilizado no controle visual baseado em posição serve para determinar a postura da câmera, neste caso particular mediante o casamento do modelo com as imagens adquiridas em cada posição. Os erros do modelo forem introduzidos nas coordenadas (dadas em centímetros) dos pontos que o definem (Figura 4.9), como se apresenta na Tabela 5.1.

Tabela 5.1. Modificação das coordenadas do modelo do objeto

Ponto	Coordenada modificada x^*	Coordenada modificada y^*
1	$x^* = x + 10\% x$	$y^* = y + 10\% y$
2	$x^* = x + 15\% x$	$y^* = y + 10\% y$
3	$x^* = x - 10\% x$	$y^* = y - 5\% y$
4	$x^* = x - 10\% x$	$y^* = y - 10\% y$

Como esperado, as mudanças nos parâmetros do modelo influenciam altamente no comportamento do controle. Nas Figuras 5.24 e 5.25 observa-se que não foi possível atingir com a tarefa de controle em nenhum dos casos, a velocidade da câmera não converge ao valor esperado, isto devido à alta dependência do modelo do objeto no cálculo da postura da câmera para a aproximação apresentada neste trabalho. Desta forma, poder-se-ia afirmar que na medida em que se apresentem erros na modelagem da cena, o controle baseado em posição não satisfaz a propriedade de robustez que deveria caracterizar os sistemas de controle visual.

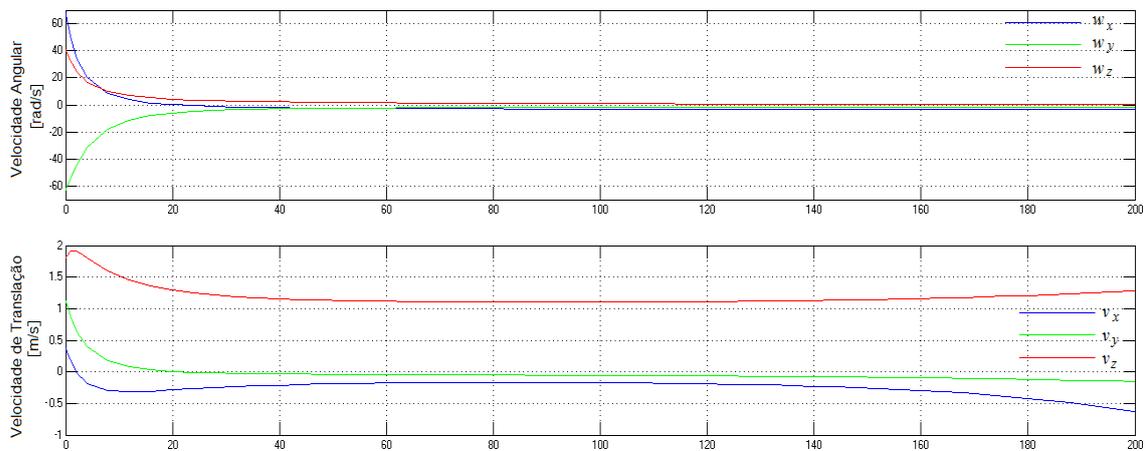


Figura 5.24. Velocidade da câmera durante o controle PBVS com coordenadas do modelo modificado como indicado na Tabela 5.1. (caso 1)

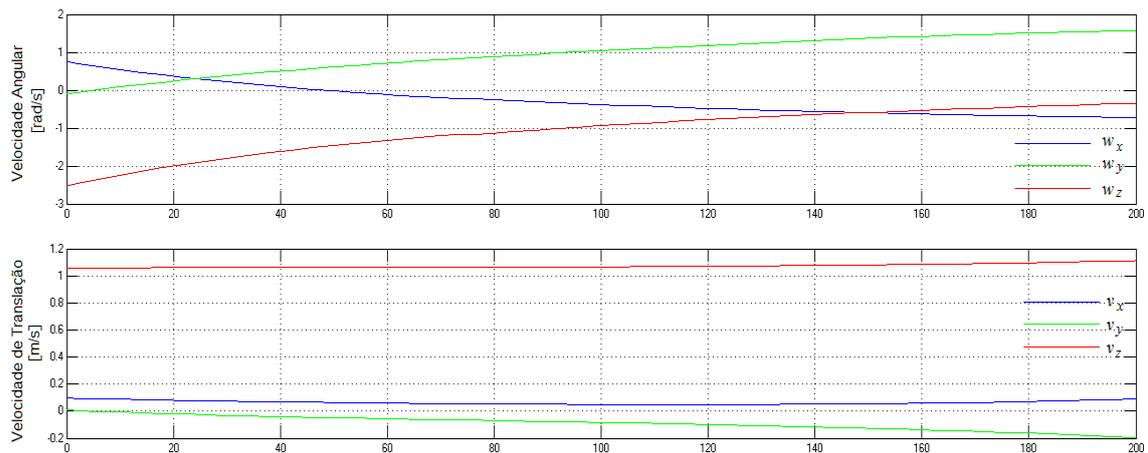


Figura 5.25. Velocidade da câmera durante o controle PBVS com coordenadas do modelo modificado como indicado na Tabela 5.1. (caso 2)

5.6- COMPARATIVA DO CONTROLE VISUAL BASEADO EM IMAGEM E O CONTROLE VISUAL BASEADO EM POSIÇÃO

Os dois tipos de controle estudados neste trabalho, através de sua simulação, apresentam características bastante favoráveis no controle de posição de um manipulador robótico a partir de imagens, mas, sob diferentes circunstâncias podem não conseguir cumprir com sua tarefa de forma robusta.

O controle visual baseado em imagem, por exemplo, apresenta robustez frente a erros de calibração e erros de medição 3D, mas o controle baseado em posição tem um melhor comportamento quando estes erros são mínimos. Contudo, os dois tipos de controle estudados nesta pesquisa são perfeitamente aplicáveis em ambientes reais tendo em conta estas características e tentando adaptá-los ao problema particular. Nas Tabelas 5.2 e 5.3 se apresenta um resumo dos resultados obtidos na etapa de simulação do controle visual com o fim de ilustrar as melhores condições de entorno para cada uma das técnicas.

Tabela 5.2. Comparação entre o IBVS e PBVS, posição inicial próxima ao alvo

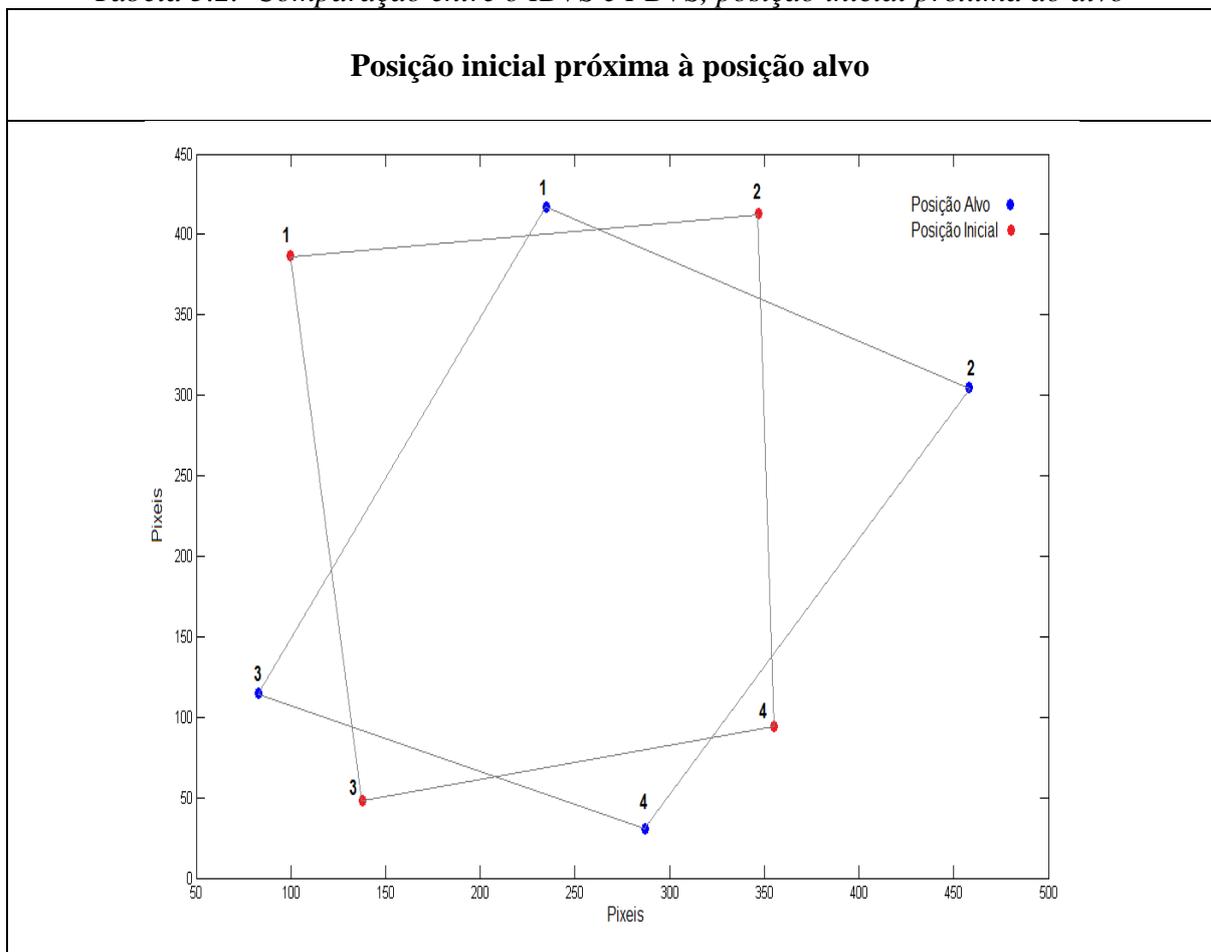


Tabela 5.2. Comparação entre o IBVS e PBVS, posição inicial próxima ao alvo
(continuação)

<p align="center">Câmera no Controle Servo Visual Baseado em Imagem</p>	<p align="center">Controle Servo Visual Baseado em Posição</p>
<p>Sem Erros</p>	
<p>Com erros de calibração</p>	
<p>Com erros de medição 3D</p>	

Tabela 5.2. Comparação entre o IBVS e PBVS, posição inicial próxima ao alvo
(continuação)

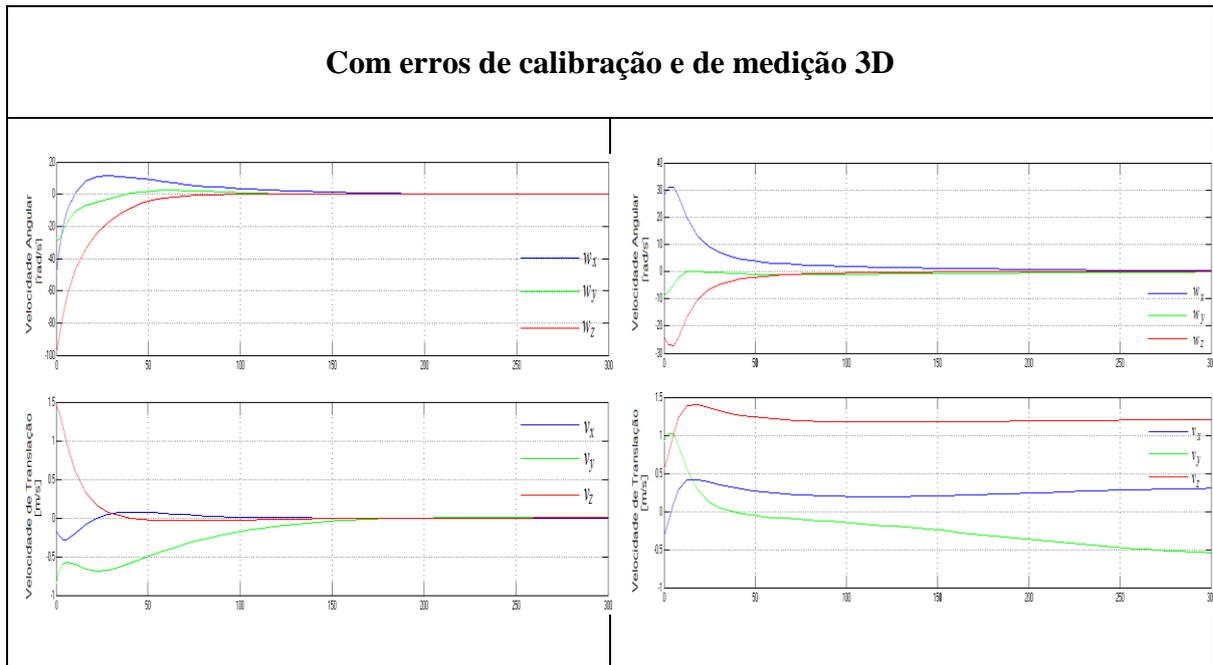


Tabela 5.3. Comparação entre o IBVS e PBVS, posição inicial afastada ao alvo

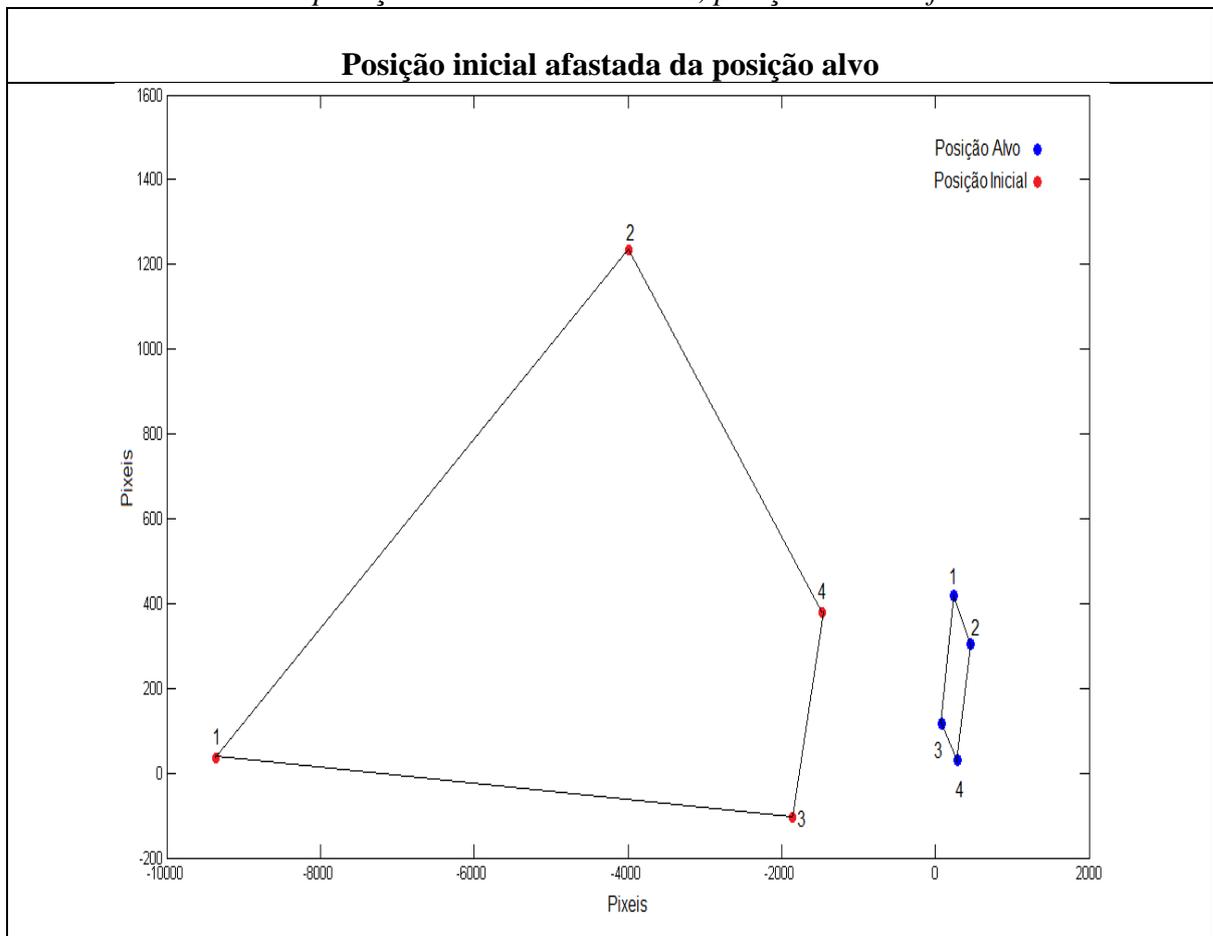


Tabela 5.3. Comparação entre o IBVS e PBVS, posição inicial afastada ao alvo
(Continuação)

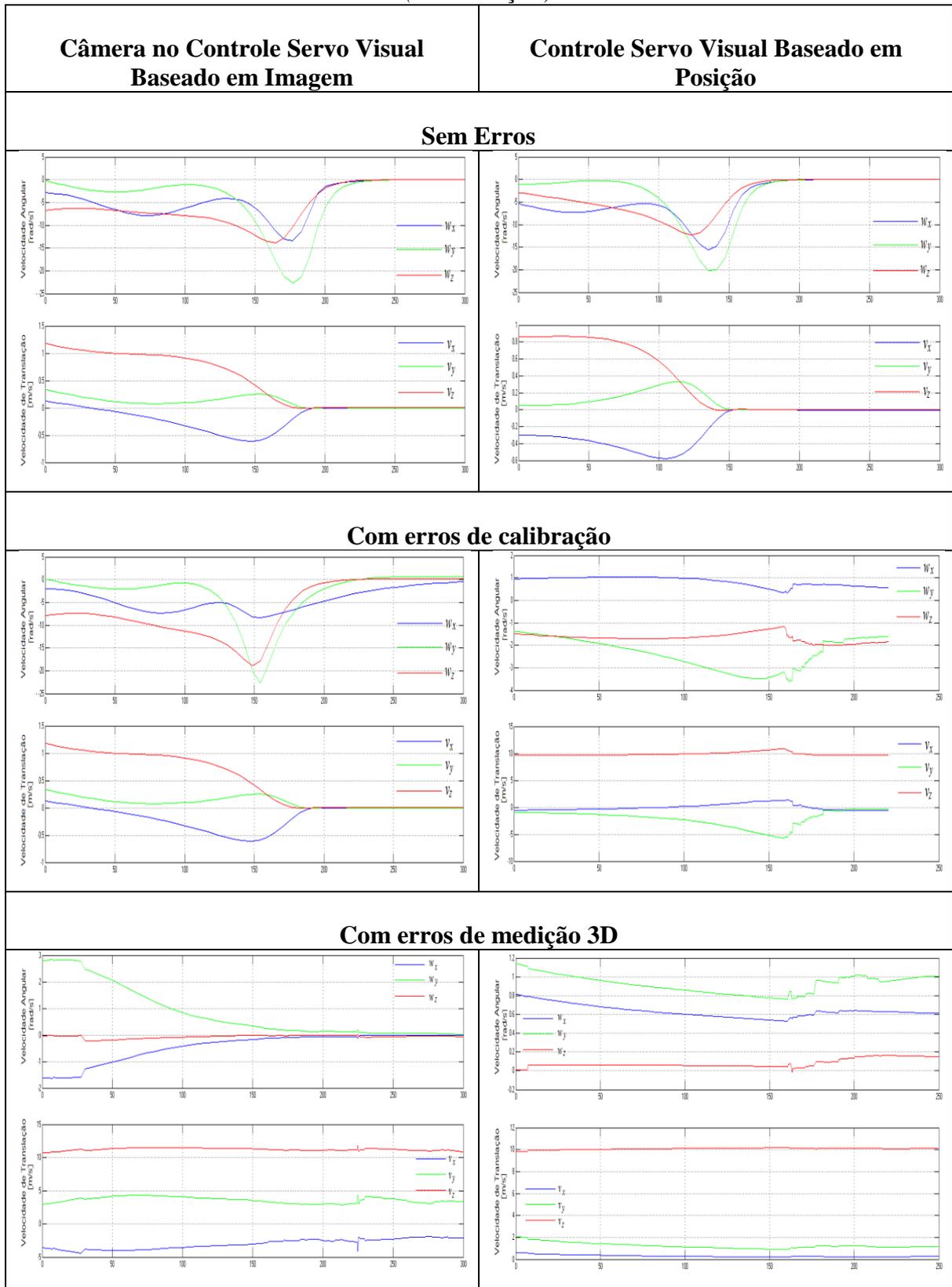
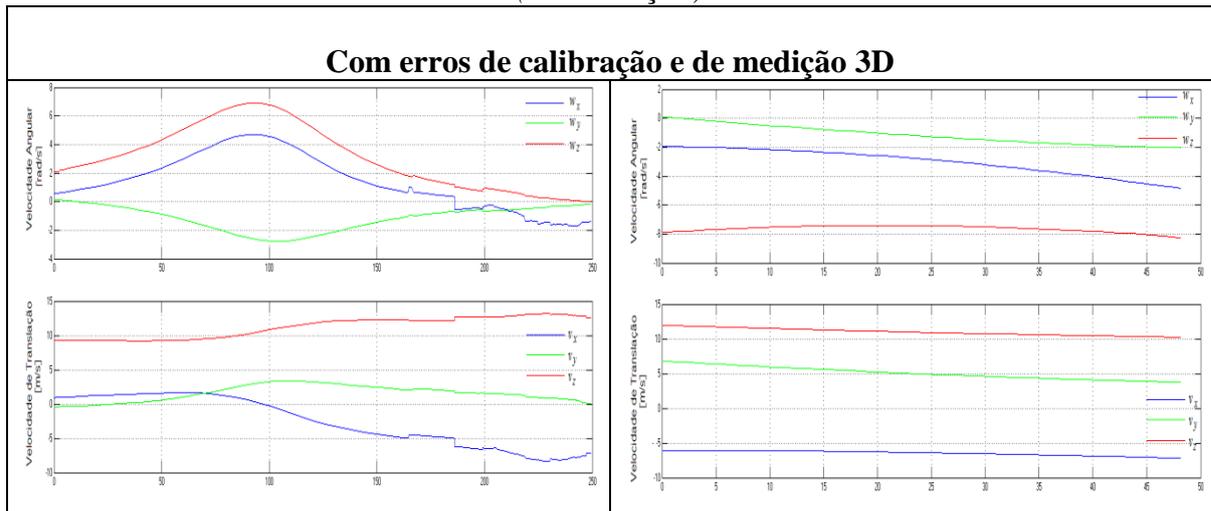


Tabela 5.3. Comparação entre o IBVS e PBVS, posição inicial afastada ao alvo
(Continuação)



De acordo com os casos observados nas Tabelas 5.2 e 5.3, não é possível afirmar de uma forma definitiva que alguma das técnicas seja melhor que a outra, porém sob certas condições especiais, o comportamento de cada uma delas é bem diferenciado. Por exemplo, o controle servo visual baseado em imagem tem um comportamento robusto aos erros de calibração e de medições 3D quando a posição inicial do elemento terminal do robô se encontra próximo à posição alvo, como apresentam as figuras da Tabela 5.2. No entanto, o controle baseado em posição tem dificuldades em atingir o objetivo de posicionamento frente a este tipo de erros. Outro comportamento se observa quando o sistema encontra-se livre de erros, neste caso o controle servo visual baseado em posição apresenta os melhores resultados. Já no caso da posição inicial estar muito afastada da posição alvo o comportamento varia um pouco no comportamento do controle baseado em imagem, já que na presença de erros de medição 3D o sistema não consegue atingir o objetivo, porém na presença de erros de calibração esta técnica apresenta robustez. O controle baseado em posição não atinge a posição desejada em nenhum dos casos onde há presença de erros, mas no caso livre de erros seu comportamento é muito satisfatório em termos de tempo e trajetória.

6 - CONCLUSÕES E PROPOSTAS DE TRABALHOS FUTUROS

Neste trabalho foi apresentado o desenvolvimento de duas metodologias de controle servo visual para manipuladores robóticos com o fim de estabelecer uma comparação de desempenho entre elas. O sistema conta com diferentes algoritmos, tanto de processamento de imagens como de controle cinemático do manipulador, que foram testados inicialmente de forma individual, para garantir seu bom desempenho. Finalmente foram elaboradas as etapas de integração dos algoritmos mediante uma simulação, o que permitiu a comparação das duas técnicas de controle visual de uma forma direta.

Uma das etapas que definiu a metodologia do trabalho foi a escolha do tipo de objeto e primitivas que seriam analisados. O fato de terem sido escolhidos objetos planos e pontos como primitivas para caracterizá-los, definiu o tipo de algoritmos de detecção implementados e, apesar de ter uma limitação dimensional devido à natureza plana dos objetos, existem muitas vantagens em termos de tempo de análise de cenas e facilidade dos algoritmos. Além disso, em geral muitos dos elementos encontrados no entorno contam com superfícies planas às quais poderiam aplicar-se os algoritmos aqui descritos.

A partir da definição das primitivas foi elaborada a etapa de rastreamento do objeto mediante o casamento robusto de pontos nas cenas, que apresentou um bom desempenho nas imagens testadas, obtendo em quase todas elas um casamento de pontos de 100% de efetividade. O sucesso da metodologia implementada está na imposição da restrição pela matriz de homografia, cujo cálculo empregando o método RANSAC, garantiu um tempo do processo suficientemente baixo e uma alta efetividade na resposta.

Uma vez que se garantiu o casamento robusto entre duas cenas, foi possível estendê-lo ao modelo do objeto e suas imagens, com o qual conseguiu-se aplicar o algoritmo de recuperação da postura da câmera, necessário em varias etapas do desenvolvimento do trabalho. Já com o casamento de características e com a recuperação da postura da câmera a partir de sua imagem foi possível definir as duas metodologias de controle servo visual estudadas nesta pesquisa, o controle servo visual baseado em imagem (IBVS) e o controle servo visual baseado em posição (PBVS).

A principal diferença entre o IBVS e o PBVS é a determinação do espaço no qual se define a variável de controle (a velocidade da câmera). Cada um dos casos apresenta uma forma diferente de calcular esta variável. No controle baseado em imagem, o cálculo da velocidade da câmera foi determinado pela matriz de interação da imagem, que relaciona de forma direta a variação entre os pontos em duas imagens de uma mesma cena e a velocidade que experimenta a câmera para obtê-las. No controle baseado em posição, o cálculo da velocidade da câmera foi obtido a partir da diferença da postura da câmera em duas posições diferentes.

Foi aplicada, nos dois casos de estudo, uma lei de controle proporcional, cujo objetivo foi o decréscimo do erro em forma exponencial; erro que definiu-se como uma função da velocidade da câmera, assim, o objetivo final do controle foi encontrar um ponto onde a velocidade da câmera fosse zero. Nesta pesquisa, implementou-se uma configuração de câmera monocular fixada no elemento terminal de um manipulador robótico, assim a velocidade da câmera é a mesma velocidade deste elemento terminal e, de acordo à cinemática do manipulador, é possível determinar a configuração do robô de em função desta velocidade. Desta forma, com o cálculo da velocidade da câmera necessária para atingir uma posição alvo e a cinemática do manipulador, foi implementado um sistema simulado de controle servo-visual.

Os resultados obtidos desta simulação permitiram observar o comportamento de cada uma das técnicas sob diferentes características do entorno, como na presença de erros nos parâmetros da matriz de calibração e nas medidas sobre a cena. De acordo com estes resultados, os dois tipos de controle cumprem, geralmente, com a tarefa de posicionamento de um manipulador robótico a partir de imagens, mas, seu comportamento depende dos tipos de erros que se apresentam durante a modelagem. De acordo com isto, não poder-se-ia afirmar que uma das técnicas é melhor que a outra, só que em diferentes circunstâncias é preferível utilizar uma sobre a outra. Por exemplo, no caso livre de erros tanto de medição como de calibração, o comportamento do PBVS é melhor que o IBVS já que o tempo de convergência é menor e a trajetória seguida pela câmera é mais parecida a uma trajetória linear. Já no caso de apresentar tanto erros de calibração como de medição, o controle IBVS sempre apresenta um melhor comportamento que o controle PBVS. Em muitos casos, o controle baseado em imagem encontrou a convergência e atingiu o objetivo, entretanto o controle baseado em posição foi mais afetado e não encontrou a convergência.

Finalmente, o fato de ter implementado estas duas aproximações de controle servo visual num ambiente simulado permitiu observar suas principais características, seus requisitos, e as melhores condições para sua possível implementação num ambiente real, possibilitando, se for necessário, uma escolha preliminar entre estes dois modelos, de acordo com as características particulares de cada caso.

A principal vantagem do controle servo visual é o fornecimento de flexibilidade no sistema robótico, já que não se limita apenas a aplicações repetitivas e a ambientes controlados, mas também pode ser utilizado em ambientes dinâmicos, contando com a possibilidade de reagir diante de possíveis mudanças no ambiente. Como demonstrado nos resultados deste trabalho, foi possível implementar o controle servo visual em manipuladores robóticos de forma simulada com condições livres de erros e com a inclusão de erros de medição e de calibração, o que leva a concluir que também é possível sua implementação em uma situação real, como por exemplo, em uma tarefa de posicionamento em ambientes externos dinâmicos ou em uma plataforma subaquática, onde é comum a presença de erros e condições do espaço de trabalho variáveis durante a realização da tarefa.

6.1- TRABALHOS FUTUROS

O trabalho até aqui apresentado poder-se-ia considerar como o passo inicial numa linha de pesquisa enorme no campo da robótica, o que permite uma grande quantidade de trabalhos complementares, em função dos diferentes objetivos traçados no início da pesquisa. Os trabalhos que se propõem são:

- Estudo de uma abordagem híbrida entre o controle visual baseado em imagem e baseado em posição, que poderia aproveitar as virtudes de cada um deles para aumentar a robustez do controle;
- Ampliação da aplicação para objetos em 3D. Neste caso, o controle visual pode-se ampliar para qualquer tipo de objeto, inicialmente com objetos compostos de superfícies planas e posteriormente ampliá-lo para superfícies não planas;
- Automação completa do processo de determinação do modelo dos objetos e seu casamento nas imagens, utilizando, por exemplo, modelos tipo CAD;
- Implementar um sistema de controle servo visual num ambiente real.

REFERÊNCIAS BIBLIOGRÁFICAS

- Andreasson, H., Treptow, A., Duckett, T. (2007). Self-localization in non-stationary environments using omni-directional vision. *Robotics and Autonomous Systems* , 55, pp. 541-551.
- Barrientos, A., Peñin, L. F., Balaguer, C., Santoja, R. A. (2007). *Fundamentos de Robótica* (2nd ed.). Madrid, ESP: McGraw-Hill.
- Bouguet, J.-Y. (2009). *Camera Calibration Toolbox for Matlab*. Acesso em 05 de 2010, disponível em www.vision.caltech.edu/bouguetj/calib_doc/index.html
- Chaumette, F. (1990). *La relation vision-commande: théorie et application à des tâches robotiques*. L'Université de Rennes I, Institut de Formation Supérieure en Informatique et Communication (IFSIC). Rennes: These.
- Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. *Conference of Control Vision, Lecture Notes in Control and Informations Systems*. 237, pp. 66-78. Springer-Verlag.
- Chaumette, F., Hutchinson, S. (2006). Visual Servo Control Part I: Basic Approaches. *Robotics Automation Magazine* , 13 (4), pp. 82-90.
- Chesi, G., Malis, E., Cipolla, R. (2000). Automatic Segmentation and Matching of Planar Contours for Visual Servoing. *Proceedings of the 2000 IEEE International Conference on Robotics Automation*, pp. 2753-2758. San Francisco, EEUU.
- Chesi, G., Malis, E., Cipolla, R. (1999). Collineation Estimation from two Unmatched Views of an Unknown Planar Contour for Visual Servoing. *10th British Machine Vision Conference*. Nottingham, UK.
- Comport, A. I. (2005). *Towards a Computer Imagination: Robust Real-time 3D Tracking of Rigid and Articulated Objects for Augmented Reality and Robotics*, L'Université de Rennes I, Institut de Formation Supérieure en Informatique et Communication (IFSIC). Rennes: These.
- Comport, A., Pressigout, M., Marchand, É., Chaumette, F. (2003). A visual servoing law that is robust to image outliers. *Proceedings of the 2003 IEEE/RSJ, Conference on Intelligent Robots and Systems*. Las Vegas: IEEE.
- Corke, P. I. (1993). Visual control of robot manipulators: a review. In: *Visual Servoing* (Vol. 7, pp. 1-31). Singapore: World Sci.
- Craig, J. J. (1989). *Introduction to Robotics: Mechanics and Control* (2nd ed.). Addison-Wesley Publishing Company Inc.

- Dementhon, D. F., Davis, L. S. (1995). Model-based object postura in 25 lines of code. *International Journal of Computer Vision* , 15 (1), pp. 123-141.
- Espiau, B. (1994). Effect of camera calibration errors on visual servoing in robotics. In: *Experimental Robotics III, series: Lecture Notes in Control and Information Sciences* (Vol. 200, pp. 182-192). Heidelberg: Springer Berlin.
- Espiau, B., Chaumette, F., Rives, P. (1992). A New Approach to Visual Servoing in robotics. *IEEE Transactions on Robotics and Automation* , 3 (8), pp. 313-326.
- Farid, H., Simoncelli, E. P. (2004). Differentiation of discrete multi-dimensional signals. *IEEE Transactions on Image Processing*, 13 (4), pp 496-508 .
- Faugeras, O., Luong, Q.-T. (2001). *The Geometry of Multiple Images*. London, England: The MIT Press.
- Feddema, J. T., Lee, C. S., Mitchel, O. R. (1989). Automatic selection of image features for visual servoing of a robot manipulator. *IEEE Proceedings on Robotics and Automation*, 2, pp. 14-19.
- Fischler, M. A., Bolles, R. C. (1981). Random Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing, Communications of ACM* , 24 (6), pp. 381-395.
- Gonzalez, R. C., Woods, R. E. (2000). *Processamento De Imagens Digitais*. Addison-Wesley Publishing Company, Inc.
- Hager, G. D. (1997). A modular system for robust positioning using feedback from stereo vision. *IEEE Transactions on Robotics and Automation* , 13 (4), pp. 582-595.
- Hager, G. D. (1995). Calibration-free visual control using projective invariance. In: *Proceedings of the IEEE Fifth International Conference on Computer Vision*. pp. 1009-1015.
- Harris, C., Stephens, M. (1988). A combined corner and edge detector. In: *Proceedings of the Fourth Alvey Vision Conference*. pp. 147-151.
- Hartley, R., Zisserman, A. (2003). *Multiple View Geometry in computer vision* (2nd ed.). Cambridge, United Kingdom: Cambridge University Press.
- Hemayed, E. E. (2003). A survey of camera self-calibration. *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance AVSS'03*. IEEE.
- Hill, J., Park, W. T. (1979). Real time control of a robot with a mobile camera. 9th *International Symposium on Industrial Robots*, pp. 233-246. Washington.
- Hutchinson, G., Hager, G., Corke, P. (1996). A Tutorial on Visual Servo Control. *IEEE Transaction on Robotics and Automation* , 5 (12), pp. 651-670.

- Iwatsuki, M., Okiyama, N. (2005). A new formulation of visual servoing based on cylindrical coordinate system. *IEEE Transactions on Robotics* , 21 (2), pp. 266 - 273.
- Johansson, B., Cipolla, R. (2002). A system for automatic postura-estimation from a single image in a city scene. *In International Conference on Signal Processing, Pattern Recognition and Applications*.
- Kim, D., Rizzi, A. A., Hager, G. D., Zeditschek, D. E. (1995). A “robust” convergent visual servoing system. *In Proceedings of the IEEE/RSJ Intelligent Robots and Systems 95. “Human Robot Interaction and Cooperative Robots”*, 1, pp. 348-353.
- Kragic, D. (2001). *Visual Servoing for Manipulation: Robustness and Integration Issues*. Royal Institute of Technology, Numerical Analysis and Computer Science, Computational Vision and Active Perception Laboratory (CVAP), Stockholm: these
- Kragic, D., Christensen, H. I. (2002). *Survey on visual servoing for manipulation*. Royal Institute of Technology, Computational Vision and Active Perception Laboratory. Stockholm: Royal Institute of Technology.
- Liu, S., Xie, W.-F., Su, C.-Y. (2009). Image-Based Visual Servoing using improved image moments. *In IEEE International Conference on Information and Automation. ICIA'09*. pp. 577-582.
- Ma, Yi, Jana Kosecká, Stefano Soatto, Shankar Sastry. *An Invitation to 3-D Vision, From Images to Models*. 2001.
- Malis, E. (1998). *Contributions à la modélisation et à la commande en asservissement visuel*, Université de Rennes I, Traitement du Signal et Télécommunications, Rennes: These.
- Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. *Proceedings of the 2004 IEEE* .
- Malis, E., Chaumette, F., Boudet, S. (1999). 2-1/2-D Visual Servoing. *IEEE Transactions on Robotics and Automation* , Vol. 15 (2), pp. 238-250.
- Marchand, É., Chaumette, F. (2002). Virtual visual servoing: a framework for real-time augmented reality. *EUROGRAPHICS 2002*, 21, pp. 289-298.
- Marey, M., Chaumette, F. (2008). Analysis of classical and new visual servoing control laws. *IEEE International Conference on Robotics and Automation*. Pasadena, CA.
- Martinet, P., Daucher, N., Gallice, J., Dhome, M. (1997). Robot control using monocular postura estimation. *Workshop New Trends Image-Based Robot Servoing IROS'97*.

- Martinet, P., Gallice, J., Khadraoui, D. (1996). Vision based control law using 3D visual features. *Proceedings of the World Automation Congress. WAC'96, Robotics and Manufacturing Systems*, 3, pp. 497-502. Montpellier.
- McCarthy, J., Earnest, L. D., Reddy, D. R., Vines, P. J. (1968). A Computer with Hands, eyes and ears. *Fall Joint Computer Conference*, 33. San Francisco.
- Mendoça, P. R., e Cipolla, R. (1999). A simple technique for self-calibration. *IEEE Computer Society Conference on Computer Vision and Pattern recognition*. 1. IEEE.
- Paul, R. P., Shimano, B., Mayer, G. E. (1981). Differential kinematic control equations for simple manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11 (6), pp. 456-460.
- Roth, G., e Whitehead, A. (2000). Using Projective Vision to Find Camera Positions in an Image Sequence. *Proceedings of the Vision Interface conference*, (pp. 87-94). Montréal, Québec, Canada.
- Sanderson, A., e Weiss, L. (1980). Image-based visual servo control using relational graph error signals. *Proc. IEEE*, (pp. 1074-1077).
- Shirai, Y., e Inoue, H. (1973). Guiding a Robot by Visual Feedback in Assembling Tasks. *Pattern Recognition*, 5, 99-108.
- Shiroma, P. M. (2004). *Controle por Visão de Veículos Robóticos*. Dissertação de Mestrado, Universidade Estadual de Campinas, Instituto de Computação.
- Snith, C. E., Papanikolopoulos, N. P. (1995). Using vision-based control techniques for grasping objects. In *IEEE International Conference on Systems, Man and Cybernetics 1995. Intelligent Systems for the 21st Century*. 5, pp. 4434 - 4439. Vancouver .
- Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Journal of Robotics and Automation*, 3 (4), pp. 323-344.
- Vincent, É., Laganière, R. (2001). An empirical study of some feature matching strategies. *Machine Graphics Vision*, 10 (3), pp. 237-259.
- Wilson, W. J., Williams, C. C., Bell, G. S. (1996). Relative end-effector control using cartesian position based visual servoing. *Transactions on Robotics and Automation*, 12 (5), pp. 684-696.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (11), pp. 1330-1334.

APÊNDICES

APÊNDICE A: DETERMINAÇÃO DE H (MATRIZ DE HOMOGRAFIA) A PARTIR DE CORRESPONDÊNCIAS

Para determinar a relação entre pontos correspondentes entre duas imagens de um plano é necessário reescrever a equação (2.4) da forma:

$$\mathbf{x}' \times H\mathbf{x} = 0$$

, onde os pontos $\mathbf{x}' \leftrightarrow \mathbf{x}$ são pontos correspondentes (bem seja entre um plano e sua imagem ou entre duas imagens do mesmo plano) em coordenadas homogêneas.

$$\mathbf{x}' = [x'_1, x'_2, x'_3]^T \quad \text{e} \quad \mathbf{x} = [x_1, x_2, x_3]^T$$

Escrevendo:

$$H = \begin{bmatrix} \mathbf{h}^{1T} \\ \mathbf{h}^{2T} \\ \mathbf{h}^{3T} \end{bmatrix}$$

, onde cada \mathbf{h}^{iT} representa uma fila de H

$$H\mathbf{x} = \begin{bmatrix} \mathbf{h}^{1T}\mathbf{x} \\ \mathbf{h}^{2T}\mathbf{x} \\ \mathbf{h}^{3T}\mathbf{x} \end{bmatrix}$$

,e

$$\mathbf{x}' \times H\mathbf{x} = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} \times \begin{bmatrix} \mathbf{h}^{1T}\mathbf{x} \\ \mathbf{h}^{2T}\mathbf{x} \\ \mathbf{h}^{3T}\mathbf{x} \end{bmatrix} = \mathbf{0}$$

$$\mathbf{x}' \times H\mathbf{x} = \begin{bmatrix} (x'_2 \mathbf{h}^{3T}\mathbf{x}) - (x'_3 \mathbf{h}^{2T}\mathbf{x}) \\ (x'_3 \mathbf{h}^{1T}\mathbf{x}) - (x'_1 \mathbf{h}^{3T}\mathbf{x}) \\ (x'_1 \mathbf{h}^{2T}\mathbf{x}) - (x'_2 \mathbf{h}^{1T}\mathbf{x}) \end{bmatrix} = \mathbf{0}$$

dado que $\mathbf{h}^{iT}\mathbf{x} = \mathbf{x}^T\mathbf{h}^i$

$$\begin{bmatrix} (x'_2 \mathbf{x}^T \mathbf{h}^3) - (x'_3 \mathbf{x}^T \mathbf{h}^2) \\ (x'_3 \mathbf{x}^T \mathbf{h}^1) - (x'_1 \mathbf{x}^T \mathbf{h}^3) \\ (x'_1 \mathbf{x}^T \mathbf{h}^2) - (x'_2 \mathbf{x}^T \mathbf{h}^1) \end{bmatrix} = \mathbf{0}$$

ou, em forma matricial

$$\begin{bmatrix} \mathbf{0}^T & -(x'_3 \mathbf{x}^T) & (x'_2 \mathbf{x}^T) \\ (x'_3 \mathbf{x}^T) & \mathbf{0}^T & -(x'_1 \mathbf{x}^T) \\ -(x'_2 \mathbf{x}^T) & (x'_1 \mathbf{x}^T) & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0$$

este sistema linear tem duas filas linearmente independentes, e pode-se reescrever

$$\begin{bmatrix} \mathbf{0}^T & -(x'_3 \mathbf{x}^T) & (x'_2 \mathbf{x}^T) \\ (x'_3 \mathbf{x}^T) & \mathbf{0}^T & -(x'_1 \mathbf{x}^T) \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0$$

$$\mathbf{A} \mathbf{h} = \mathbf{0}$$

Desta forma, a cada correspondência de pontos $\mathbf{x}' \leftrightarrow \mathbf{x}$ obtêm-se duas equações linearmente independentes, com quatro correspondências pode-se obter uma solução \mathbf{h} do sistema.

Devido aos erros introduzidos nas medições das coordenadas dos pontos não é possível encontrar uma solução exata, por esta razão utiliza-se uma solução aproximada de \mathbf{h} tal que:

$$\min_h \|\mathbf{A} \mathbf{h}\|, \text{ sujeito a } \|\mathbf{h}\| = 1$$

A solução deste sistema é o eigenvetor de $\mathbf{A}^T \mathbf{A}$ correspondente ao menor valor singular de \mathbf{A} que pode ser obtido a partir da decomposição em valores singulares dado pela SVD.

APÊNDICE B : DECOMPOSIÇÃO EM VALORES SINGULARES SVD

A decomposição por valores singulares (SVD) por suas siglas em inglês singular (value decomposition) é uma das decomposições de matrizes mais uteis, particularmente para cálculos numéricos.

Se A é uma matriz real de $m \times n$, então existem matrizes ortogonais:

$$U = [u_1, u_2, \dots, u_m] \in \mathbb{R}^{m \times m} \quad \text{e} \quad V = [v_1, v_2, \dots, v_n] \in \mathbb{R}^{n \times n}$$

tais que a matriz A pode-se fatorar como:

$$A = U\Sigma V^T$$

Onde

$$\Sigma = \begin{cases} \begin{bmatrix} \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n) \\ \mathbf{0}_{(m-n) \times n} \end{bmatrix} & \text{se } m \geq n \\ \begin{bmatrix} \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n) & \mathbf{0}_{m \times (n-m)} \end{bmatrix} & \text{se } n \geq m \end{cases}$$

Em qualquer caso, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$, $p = \min\{m, n\}$ são números reais não negativos chamados *valores singulares de A*. Os vetores u_i e v_i que formam as colunas das matrizes U e V são os *vetores singulares* de AA^T e de $A^T A$ respectivamente, e cumprem que:

$$Av_i = \sigma_i u_i \quad \text{para } i = 1, 2, \dots, p$$

$$Au_i = 0 \quad \text{para } i = p + 1, \dots, n$$

APÊNDICE C: PARÂMETROS DE DENAVIT-HATEMBERG

Uma das convenções mais comumente utilizadas para o análise das relações entre as juntas de um manipulador robótico e a posição e orientação do seu elemento terminal, é denominada a convenção de Denavit-Hatemberg ou convenção D-H, na qual cada transformação homogênea ${}^{i-1}\mathbf{A}_i$ representa-se como um produto de quatro transformações básicas:

$${}^{i-1}\mathbf{A}_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (D.1)$$

$$= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (D.2)$$

Onde os parâmetros $\theta_i, a_i, d_i, \alpha_i$ estão associados ao elo i e à junta i . estes parâmetros são geralmente denominados: ângulo da junta, offset da junta, comprimento do elo e rotação da junta respectivamente. Três destes parâmetros são constantes para cada junta e um de eles varia dependendo do tipo de junta, numa junta de revolução a variável é θ_i e para juntas prismáticas a variável é d_i . A vantagem deste tipo de parametrização é a possibilidade de determinar a transformação ${}^{i-1}\mathbf{A}_i$ com tão só estes quatro parâmetros e com uma escolha adequada dos referenciais associados às diferentes juntas do manipulador. Uma descrição detalhada do algoritmo para a determinação adequada dos referenciais associados a cada junta é apresentado em Barrientos et. al. (2007).