

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

**SISTEMA DE VISÃO ROBÓTICA PARA
RECONHECIMENTO E LOCALIZAÇÃO DE OBJETOS
SOB MANIPULAÇÃO POR ROBÔS INDUSTRIAIS EM
CÉLULAS DE MANUFATURA**

LAERCIO ARRAES JARDIM

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

**PUBLICAÇÃO: ENM.DM –
BRASÍLIA/DF: OUTUBRO – 2006**

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

**SISTEMA DE VISÃO ROBÓTICA PARA IDENTIFICAÇÃO
E LOCALIZAÇÃO DE OBJETOS SOB MANIPULAÇÃO
POR ROBÔS INDUSTRIAIS EM CÉLULAS DE
MANUFATURA.**

LAERCIO ARRAES JARDIM

**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. Dr. JOSÉ MAURÍCIO S. T. DA MOTTA - (ENM-UnB)
(Orientador)**

**Prof. Dra. CARLA MARIA C. C. KOIKE (CIC/UnB)
(Examinador Externo)**

**Prof. Dr. TEODIANO FREIRE BASTOS FILHO - (UFES)
(Examinador Externo)**

BRASÍLIA/DF, 19 DE OUTUBRO DE 2006

FICHA CATALOGRÁFICA

JARDIM, LAERCIO ARRAES

Sistema de Visão Robotica para Identificação e Localização de Objetos sob Manipulação por Robôs Industriais em Células de Manufatura.

[Distrito Federal] 2006.

xvii, 110p., 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2006). Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

1. Visão computacional

2. Projetivas invariantes

3. Identificação e localização de Objetos

4. Vértices

I. ENM/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

JARDIM, L. A. (2006). Sistema de Visão Robótica para Identificação e Localização de Objetos sob Manipulação por Robôs Industriais em Células de Manufatura. Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-04A/06, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 110p.

CESSÃO DE DIREITOS

AUTOR: Laercio Arraes Jardim.

TÍTULO: Sistema de Visão Robótica para Identificação e Localização de Objetos sob Manipulação por Robôs Industriais em Células de Manufatura.

GRAU: Mestre ANO: 2006

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.

Laercio Arraes Jardim

AGRADECIMENTOS

À Vida,

Essa dádiva que o Criador nos deu.

À Luz do Sol

Por brilhar todos os dias de minha vida.

Aos Mestres

Por transmitir seus conhecimentos. Em especial, o meu orientador, por seus conhecimentos transmitidos com paciência e dignidade.

À Todas...

... as pessoas que me apoiaram.

RESUMO

SISTEMA DE VISÃO ROBÓTICA PARA RECONHECIMENTO E LOCALIZAÇÃO DE OBJETOS SOB MANIPULAÇÃO POR ROBÔS INDUSTRIAIS EM CÉLULAS DE MANUFATURA

Autor: Laercio Arraes Jardim

Orientador: Prof. Dr. José Maurício S. T. da Motta

Programa de Pós-graduação em Sistemas Mecatrônicos

Brasília, Outubro de 2006.

Existem várias técnicas de aplicação de visão robótica para o reconhecimento (identificação e localização) de objetos industriais tipicamente encontrados em uma célula de manufatura, objetivando a pega e transporte de peças para agrupamento de forma inteiramente automática por manipuladores industriais.

Visão robótica automática para objetos industriais complicados é uma importante e ainda difícil tarefa, especialmente no estágio de extração de características desses objetos. Propõe-se um modelo de reconhecimento de objetos de formatos diversos, utilizando-se de extração de pontos característicos (vértices) para formação de bancos de invariantes, para posterior identificação e localização.

A fase de identificação dos objetos é desenvolvida em dois processamentos, *off-line* e *on-line*. O processo *off-line* tem como objetivo gerar um banco de dados, onde seriam armazenadas informações necessárias à identificação de objetos. Essas informações são constituídas de vetores de projetivas invariantes. Os vetores de projetivas invariantes são gerados através de uma lista de vértices extraídos dos objetos chamados modelos. A ordenação da lista dos vértices deve ser de tal maneira que sua seqüência (horária) apresente a geometria básica do modelo. O processo *on-line* encarrega-se de identificar objetos que tenham sido previamente armazenados no banco de dados.

A fase de localização foi desenvolvida com o objetivo de determinar posições e orientações de objetos em um sistema de referência conhecida. É possível relacionar, nesta

fase, os diversos planos que o objeto, devidamente identificado, se encontra, podendo ser o plano da imagem, da câmera ou da base do manipulador.

O reconhecimento de objetos, através de sua identificação por projetiva invariante e sua localização em coordenadas da câmera, permitirá o posicionamento do manipulador para a apreensão com controle de posição e orientação.

ABSTRACT

There are several techniques for applying robotic vision for identification and localization of industrial objects. These objects are typically found in manufacturing cells. The purpose is to catch and transport the pieces for automatically grouping.

Robotic vision for industrial objects is an important and still difficult task, especially feature extraction of the object image. For that, this work proposes a model for recognition of objects different shapes, most of time symmetric, using the extraction of feature points (corners) for creating a database of invariants, for identification and localization.

The step of object identification is developed in two different processes: on-line and off-line. The off-line step has the objective of creating a database, where the information needed to identify the objects is recorded by using vectors of invariant projective. These vectors are defined from a list of corners extracted from an object image. The order of the corners listed is sorted in some way that its sequence shows the basic geometry of the model. The on-line process has the objective of identifying the objects previously recorded.

The localization step was developed to assign positions and orientations of the object in a reference coordinate system. At this stage, it is possible to relate all the coordinate frames previously assigned to the object, like the image, camera or base manipulator frames

The recognition of objects, through the identification of invariant projective and its localization in camera coordinates, allows the positioning of a manipulator to catch and to control its position and orientation.

ÍNDICE

1.	INTRODUÇÃO.....	13
2.	NOÇÕES DE ÓTICA E PROCESSAMENTO DE IMAGENS	16
2.1	IMAGENS DIGITAIS – NOÇÕES DE ÓTICA.....	17
2.1.1	Imagens de Intensidade.....	17
2.1.2	Ótica Básica.....	18
2.1.3	Radiometria Básica	20
2.1.4	Formação Geométrica da Imagem.....	21
2.2	PARÂMETROS DE CÂMERAS.....	23
2.2.1	Parâmetros Extrínsecos	23
2.2.2	Parâmetros Intrínsecos	25
2.3	FILTRAGEM DE RUÍDOS	26
2.3.1	Filtro Linear.....	27
2.4	CARACTERÍSTICAS DAS IMAGENS	30
2.4.1	Contorno.....	31
2.4.2	Vértices	34
2.4.3	Centro de Massa	36
2.4.4	Segmentação de Imagem.....	38
3.	RECONHECIMENTO DE OBJETOS – IDENTIFICAÇÃO.....	40
3.1	Projetiva Invariante para Geração de Modelos.....	41
3.2	Configuração Geométrica do modelo	42
3.3	Criação do Banco de Invariantes	44
3.4	Identificação de Peças por Comparação	45
3.4.1	Processamento Off-Line.....	45
3.4.2	Processamento On-Line	50
4.	RECONHECIMENTO DE OBJETOS – LOCALIZAÇÃO	52
4.1	Conceitos Básicos.....	52
4.2	Localização de Objetos – Imagem de Intensidade.....	52

4.2.1	Modelo de Câmera de Perspectiva Completa	53
4.2.2	Modelo de Câmera de Perspectiva Fraca.....	55
4.2.3	Exemplo de Aplicação para Modelo de Câmera de Perspectiva Fraca	56
5.	CONTROLE DO ESPAÇO OPERACIONAL UTILIZANDO VISÃO.....	59
5.1	Aspectos Gerais.....	59
5.2	Subsistema de Controle do Manipulador	59
5.2.1	Espaço Operacional e Espaço Junta – Cinemática Direta e Inversa....	59
5.2.2	Matriz Jacobiano do Manipulador	61
5.3	Subsistema de Visão	62
5.3.1	Formação da Imagem.....	63
5.3.2	Jacobiano da imagem	64
5.3.3	Jacobiano Completo	65
5.4	Sistema de Controle Por Visão	67
5.5	Cinemática do Erro	69
6.	EXPERIMENTOS	71
6.1	Identificação dos Objetos.....	71
6.1.1	Processamento Off-line – Geração do Banco de Invariantes	71
6.1.2	Processamento On-Line – Identificação de Objetos.....	80
7.	CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS.....	107
	REFERÊNCIAS BIBLIOGRÁFICAS	109

LISTA DE TABELAS

TABELA 2.1 - MATRIZ 8x8 DE UMA IMAGEM SINTÉTICA.....	17
TABELA 3.1 – LISTA DE PONTOS (VÉRTICES) DO MODELO 1.....	47
TABELA 3.2 - LISTA DE PONTOS (VÉRTICES) DO MODELO 2.	47
TABELA 3.3 – VALORES DOS VETORES INVARIANTES DOS MODELOS 1 E 2.	49
TABELA 6.1 – VÉRTICES DO MODELO 1.	75
TABELA 6.2 – VÉRTICES DO MODELO 2.	76
TABELA 6.3 – VÉRTICES DO MODELO 3.	76
TABELA 6.4 – VÉRTICES DO MODELO 1.	76
TABELA 6.5 – VÉRTICES DO MODELO 2.	76
TABELA 6.6 – VÉRTICES DO MODELO 3.	77
TABELA 6.7 – VÉRTICES DO MODELO 1.	78
TABELA 6.8 – VÉRTICES DO MODELO 2.	78
TABELA 6.9 – VÉRTICES DO MODELO 3.	78
TABELA 6.10 – RESULTADO DOS CÁLCULOS DE PROJETIVA INVARIANTE PARA OS MODELOS 1, 2 E 3.....	80
TABELA 6.11 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 1.	82
TABELA 6.12 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 2.....	82
TABELA 6.13 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 3.....	82
TABELA 6.14 – VÉRTICES DO OBJETO (PIXEL).....	85
TABELA 6.15 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 1.....	86
TABELA 6.16 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 2.....	86
TABELA 6.17 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 3.....	87
TABELA 6.18 – VÉRTICES DO OBJETO (PIXEL).....	91
TABELA 6.19 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 1.....	91
TABELA 6.20 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 2.....	92
TABELA 6.21 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 3.....	92
TABELA 6.22 – VÉRTICES DO OBJETO (PIXEL).....	96
TABELA 6.23 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 1.....	96
TABELA 6.24 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 2.....	96
TABELA 6.25 – ERROS ENCONTRADOS EM K-COMBINAÇÕES DO OBJETO COM O MODELO 3.....	97
TABELA 6.26 – LISTA DOS VÉRTICES (PIXEL) SELECIONADOS DO OBJETO 1.....	101
TABELA 6.27 – LISTA DOS VÉRTICES (PIXEL) SELECIONADOS DO OBJETO 2.....	102
TABELA 6.28 – LISTA DOS VÉRTICES (PIXEL) SELECIONADOS DO OBJETO 3.....	103

LISTA DE FIGURAS

FIGURA 2.1 – (A) IMAGEM EM FORMA DE SUPERFÍCIE. (B) IMAGEM SINTÉTICA.....	17
FIGURA 2.2 – GEOMETRIA ÓTICA DAS LENTES FINAS.....	19
FIGURA 2.3 – FORMAÇÃO DE IMAGENS ATRAVÉS DAS LENTES FINAS.	20
FIGURA 2.4 – MODELO CÂMERA PERSPECTIVA.....	22
FIGURA 2.5 – EXEMPLO DE TRANSFORMAÇÃO ENTRE DOIS SISTEMAS – MUNDO (P_w) E CÂMERA (P_c).....	25
FIGURA 2.6 – (A) IMAGEM ORIGINAL, (B) IMAGEM ATENUADA COM MÁSCARA $A_{MEDI A1}$, (C) IMAGEM ATENUADA COM MÁSCARA $A_{MEDI A2}$	28
FIGURA 2.7 – (A) IMAGEM ORIGINAL. (B) IMAGEM COM ATENUAÇÃO GAUSSIANA $\sigma = 1.5$	30
FIGURA 2.8 – (A) IMAGEM ORIGINAL. (B) CONTORNO EXTRAÍDO POR <i>CANNY EDGE DETECTOR</i>	32
FIGURA 2.9 – (A) IMAGEM ORIGINAL. (B) EXTRAÇÃO DE CONTORNO POR ROBERT – $\tau = 34.88$. (C) EXTRAÇÃO DE CONTORNO POR SOBEL – $\tau = 78.4$	33
FIGURA 2.10 – (A) IMAGEM ORIGINAL. (B) IMAGEM BINARIZADA COM LIMIAR = 170. (C) VÉRTICES DE IMAGEM BINARIZADA.	36
FIGURA 2.11 – (A) IMAGEM BINARIZADA COM CENTRO DE MASSA. (B) IMAGEM BINARIZADA ROTACIONADA COM CENTRO DE MASSA.....	38
FIGURA 2.12 – (A) IMAGEM ORIGINAL (B) IMAGEM BINARIZADA. (C) IMAGEM SEGMENTADA SEM O PRIMEIRO OBJETO. (D) IMAGEM SEGMENTADA SEM DOIS OBJETOS.	39
FIGURA 3.1 – (A) IMAGEM DO MODELO 1 - APRESENTAÇÃO DA IMAGEM DO OBJETO MODELO 1 COM OS VÉRTICES;	43
FIGURA 3.2 – (A) IMAGEM DO MODELO 2. (A) - APRESENTAÇÃO DA IMAGEM DO OBJETO MODELO 2 COM OS VÉRTICES; (B) GEOMETRIA DO MODELO 2.....	43
FIGURA 3.3 – ESQUEMA DE ORDENAÇÃO DE VÉRTICES NO SENTIDO HORÁRIO.....	44
FIGURA 3.4 – FLUXOGRAMA DA TÉCNICA DE RECONHECIMENTO DE OBJETOS POR MEIO DO CÁLCULO DE INVARIANTES. PROCESSAMENTO OFF-LINE.	46
FIGURA 3.5 – (A) IMAGEM DO MODELO 1. (B) IMAGEM DO MODELO 2.	47
FIGURA 3.6 – GEOMETRIA DO MODELO 1 E LISTA DOS PONTOS NO SENTIDO HORÁRIO.....	48
FIGURA 3.7 – GEOMETRIA DO MODELO 2 E LISTA DOS PONTOS NO SENTIDO HORÁRIO.....	49
FIGURA 3.8 – FLUXOGRAMA DA TÉCNICA DE RECONHECIMENTO DE OBJETOS. PROCESSAMENTO ON-LINE.	50
FIGURA 4.1 – FASES PARA A LOCALIZAÇÃO DO OBJETO. (A) IMAGEM DO POSICIONAMENTO DO OBJETO SOBRE UMA GRADE. (B) HISTOGRAMA DA IMAGEM DO OBJETO. (C) IMAGEM BINARIZADA. (D) POSIÇÕES DO OBJETO E MARCAS - PIXELS.	57
FIGURA 5.1 – SISTEMAS DE COORDENADAS. (B) BASE DO MANIPULADOR, (G) GARRA E (C) CÂMERA.....	61
FIGURA 5.2 – SISTEMAS DE COORDENADAS. (B) BASE DO MANIPULADOR, (OBJ) OBJETO E (C) CÂMERA.	63
FIGURA 5.3 – RELAÇÃO ENTRE OS JACOBIANOS MANIPULADOR, IMAGEM E COMPLETO.....	65
FIGURA 5.4 – SISTEMAS DE CONTROLE POR POSIÇÃO (ALMEIDA, 2004).....	68
FIGURA 5.5 – SISTEMAS DE CONTROLE POR IMAGEM (ALMEIDA, 2004).	68
FIGURA 6.1 – MODELOS DE OBJETOS UTILIZADOS. (A) MODELO 1. (B) MODELO 2. (C) MODELO 3.....	72

FIGURA 6.2 – HISTOGRAMAS. (A) HISTOGRAMA MODELO 1. (B) HISTOGRAMA MODELO 2. (C) HISTOGRAMA MODELO 3.	73
FIGURA 6.3 – IMAGEM BINARIZADA. (A) IMAGEM BINARIZADA DO MODELO 1. (B) IMAGEM BINARIZADA DO MODELO 2. (C) IMAGEM BINARIZADA DO MODELO 3.	74
FIGURA 6.4 – APÓS A EXTRAÇÃO DOS VÉRTICES. (A) MODELO 1 – VÉRTICES TAU (τ) = 1,0527E+5. (B) MODELO 2 – VÉRTICES TAU (τ) = 1,0921E+5. (C) MODELO 3 – VÉRTICES TAU (τ) = 6.2502E+4.	75
FIGURA 6.5 – IMAGEM DOS OBJETOS COM OS VÉRTICES NECESSÁRIOS À GERAÇÃO DA GEOMETRIA DO MODELO. (A) MODELO 1. (B) MODELO 2. (C) MODELO 3.	77
FIGURA 6.6 – GEOMETRIA DOS OBJETOS. (A) MODELO 1. (B) MODELO 2. (C) MODELO 3.	79
FIGURA 6.7 – (A) IMAGEM ORIGINAL DO OBJETO. (B) IMAGEM BINARIZADA APÓS A EXTRAÇÃO DOS VÉRTICES.	81
FIGURA 6.8 – GEOMETRIA DO OBJETO E TABELA DOS VÉRTICES EXTRAÍDOS.	81
FIGURA 6.9 – RESULTADO DAS COMPARAÇÕES – IDENTIFICAÇÃO DO OBJETO.	83
FIGURA 6.10 – (A) IMAGEM ORIGINAL DO OBJETO. (B) IMAGEM BINARIZADA APÓS A EXTRAÇÃO DOS VÉRTICES.	84
FIGURA 6.11 – GEOMETRIA DO OBJETO E TABELA DOS VÉRTICES EXTRAÍDOS.	85
FIGURA 6.12 – RESULTADO DAS COMPARAÇÕES – IDENTIFICAÇÃO DO OBJETO.	89
FIGURA 6.13 – (A) IMAGEM ORIGINAL DO OBJETO. (B) IMAGEM BINARIZADA APÓS A EXTRAÇÃO DOS VÉRTICES.	90
FIGURA 6.14 – GEOMETRIA DO OBJETO E TABELA DOS VÉRTICES EXTRAÍDOS.	90
FIGURA 6.15 – RESULTADO DAS COMPARAÇÕES – IDENTIFICAÇÃO DO OBJETO.	94
FIGURA 6.16 – (A) IMAGEM ORIGINAL DO OBJETO – TRANSLADADA E ROTACIONADA. (B) IMAGEM BINARIZADA APÓS A EXTRAÇÃO DOS VÉRTICES.	95
FIGURA 6.17 – GEOMETRIA DO OBJETO E TABELA DOS VÉRTICES EXTRAÍDOS.	95
FIGURA 6.18 – RESULTADO DAS COMPARAÇÕES – IDENTIFICAÇÃO DO OBJETO.	98
FIGURA 6.19 – IMAGEM ORIGINAL.	99
FIGURA 6.20 – HISTOGRAMA DA IMAGEM ORIGINAL.	99
FIGURA 6.21 – IMAGEM BINARIZADA E VÉRTICES EXTRAÍDOS.	100
FIGURA 6.22 – IMAGEM DOS VÉRTICES SELECIONADOS DO OBJETO 1.	100
FIGURA 6.23 – GEOMETRIA DO OBJETO 1.	101
FIGURA 6.24 – IMAGEM DOS VÉRTICES SELECIONADOS DO OBJETO 2.	101
FIGURA 6.25 – GEOMETRIA DO OBJETO 2.	102
FIGURA 6.26 – IMAGEM DOS VÉRTICES SELECIONADOS DO OBJETO 3.	102
FIGURA 6.27 – GEOMETRIA DO OBJETO 3.	103
FIGURA 6.28 – RESULTADO DA COMPARAÇÃO DO OBJETO 1 COM OS MODELOS 1, 2 E 3.	104
FIGURA 6.29 – RESULTADO DA COMPARAÇÃO DO OBJETO 2 COM OS MODELOS 1, 2 E 3.	105
FIGURA 6.30 – RESULTADO DA COMPARAÇÃO DO OBJETO 3 COM OS MODELOS 1, 2 E 3.	106

1. INTRODUÇÃO

Com as palavras: “... *Tomemos u’a máquina autônoma que tenha energia e um motor para manifestá-la e ainda sutis dispositivos para dosar e dirigir suas manifestações, mas que não tenha um órgão que permita seu comando exterior. Ela pode variar e mesmo matizar sua ação, mas é incapaz de agir sem que mensagens exteriores lhe digam quando e como. Em outras palavras, ela não tem nenhum “programa” ...*” (Latil, 1959), pode-se observar que Pierre de Latil, na década de 50 do século XX, definia como seriam as máquinas inteligentes nos tempos mais modernos, e que dependeriam essencialmente de suas programações e de comportamentos pré-determinados.

O autor completa seu pensamento quando compara a estátua de Condillac com a tartaruga de Grey Walter (uma das primeiras máquinas descrita com inteligência artificial). “... *Dotemo-la, porém, de um sentido eletrônico, depois de um outro e, depois, de muitos... Que acontecerá, então?... Se um clérigo do XVIII século pôde, por extraordinário golpe de audácia intelectual, demonstrar que sua estátua seria capaz de ter “idéias” desde que fôsse dotada de sentidos, um cientista do XX século poderá bem demonstrar isso com um robô eletrônico...*”.

Nota-se que no final do parágrafo anterior aparecem as seguintes palavras: “**desde que**”, denotando a condição necessária e suficiente para a conclusiva “capaz de ter idéias”, do silogismo, cuja premissa é: “dotada de sentidos”, ou seja, o pensamento artificial deverá ser dotado de conhecimentos prévios, necessários para tomadas de decisões.

Atualmente, a comunidade de visão computacional é consciente de que, em diversas pesquisas desenvolvidas, necessita-se de processamento de tarefas bastante complexas para interpretação de imagens, elevando assim o custo e reduzindo a velocidade computacional, comprometendo a precisão e a flexibilidade dos sistemas.

Reconhecimento de objetos é parte essencial de quase todos os sistemas de robótica de alto nível. Reconhecer um objeto significa identificá-lo e localizá-lo. Ainda não há uma técnica geral que identifique e localize qualquer tipo de objeto, independente de suas propriedades intrínsecas como cor, forma, medidas e outras.

Técnicas para reconhecimento de objetos podem ser classificadas em duas classes gerais: uma que se utiliza de bases de dados dos modelos de objetos (nessa classe, encontra-se a representação geométrica dos mesmos que depende da imagem e de como são armazenados seus dados), e outra, que usa as aparências de objetos, utilizando-se das suas características globais e locais (Bandlow, 1998).

Três técnicas serão apresentadas para identificação de objetos: a) árvores de interpretação – baseiam-se na identificação de todas as partes de um objeto em uma imagem que combine com um dado modelo, previamente armazenado; b) Invariantes – indexam geometrias de modelos previamente armazenados para identificar objetos em uma imagem; e c) métodos que se baseiam em aparências – utilizam aparência ao invés das características do objeto na imagem.

Em geral, a Geometria Projetiva fornece uma excelente ferramenta para identificação de objetos, através do emprego de geometria invariante de pontos característicos dos mesmos. Esses pontos característicos podem ser vértices, linhas ou curvas.

Essa pesquisa tem como foco principal o reconhecimento (identificação e localização) de objetos em imagens, através de comparação desses objetos com modelos previamente armazenados. Os objetos modelos têm suas características geométricas conhecidas e estão armazenadas em um banco de invariantes.

O problema da localização de um objeto consiste em conhecer a posição e a orientação desse objeto em uma determinada imagem. Para tanto, é necessário conhecer o modelo do sensor (câmera) e o modelo geométrico do objeto. Existem dois métodos de localização de objetos em imagens que se baseiam em apenas uma imagem. O primeiro é a localização de objetos no espaço 3-D através do método iterativo de Newton, que resolve um sistema de equações não lineares, sendo necessário o conhecimento prévio de alguns pontos característicos dessa imagem e os parâmetros intrínsecos e extrínsecos da câmera. O segundo é o método de modelo de câmera de perspectiva fraca que se utiliza do conhecimento prévio do objeto identificado na imagem, não sendo necessário conhecer os parâmetros intrínsecos e extrínsecos da câmera (Trucco *et al*, 1998). O segundo método é também foco deste trabalho.

Neste trabalho, no capítulo 2, serão apresentados os principais conceitos de ótica e processamento de imagens. A extração de características da imagem (vértices) é o foco principal desse capítulo. No capítulo 3, serão apresentados os modelos mais conhecidos para o reconhecimento de objetos – **identificação**, e também descreverá os procedimentos necessários à identificação de objetos por invariantes, abrangendo as fases off-line (geração do banco de invariantes dos modelos) e on-line (identificação de objetos por comparação). O capítulo 4 é destinado às técnicas de **localização** de objetos por dois modelos: câmera perspectiva completa e câmera perspectiva fraca. No capítulo 5 será apresentado o controle de um manipulador face a um sistema de visão. Nesse capítulo, dois subsistemas caracterizam o sistema de controle do espaço de trabalho: subsistema de controle do manipulador e subsistema de visão. O capítulo 6 apresenta os resultados de experimentos sobre reconhecimento de objetos. No capítulo 7 as conclusões e sugestões de trabalhos futuros.

Espera-se neste trabalho, mostrar a aplicação de sistemas de visão computacional, no sentido de reconhecimento de objetos, com a finalidade da pega e classificação desses objetos, por um manipulador industrial.

2. NOÇÕES DE ÓTICA E PROCESSAMENTO DE IMAGENS

Os elementos básicos de um sistema de visão computacional são: a) aquisição; b) armazenamento; c) processamento; d) comunicação; e e) exibição de imagens. O processamento de imagens é o elemento em que a maioria de suas funções pode ser desenvolvida em software, ou seja, envolvem procedimentos que são expressos em forma de algoritmos. Essas funções consistem em avaliar as propriedades da imagem e suas transformações.

Com o processamento de imagens pode-se: a) melhorar a qualidade de apresentação dessas imagens; b) compactar imagens para melhorar a transmissão; c) restaurar imagens para eliminar efeitos de degradação; e d) extrair características (contorno, vértices e outras) dessas imagens, com diversas finalidades, por exemplo, o reconhecimento de objetos.

As rotinas utilizadas no processamento de imagens deste trabalho foram desenvolvidas em MATLAB (Versão 6.0), um software matemático cujo núcleo está no processamento de matrizes e vetores. Além de operações algébricas comuns com matrizes, o MATLAB fornece operações com vetores que permitem a manipulação rápida de um conjunto de dados, de muitas maneiras diferentes. Enfim, a combinação existente entre a estrutura de dados vetoriais, os recursos de programação e uma interface gráfica que permite a utilização em desenvolvimento de aplicativos, torna o MATLAB uma ferramenta poderosa para a resolução de problemas em diversas áreas, principalmente em visão computacional, considerando que uma imagem é um vetor bidimensional (matriz).

Utilizando-se da ferramenta MATLAB, é possível visualizar, através da **Figura 2.1**, como uma imagem pode ser apresentada. A Tabela 2.1 apresenta a matriz 8x8 de uma imagem sintética, a **Figura 2.1a** apresenta esta mesma matriz na forma de superfície e a **Figura 2.1b** mostra a imagem sintética, representada como intensidade de brilho.

Tabela 2.1 - Matriz 8x8 de uma imagem sintética.

1	2	3	4	1	1	2	1
2	2	3	2	1	2	2	1
3	1	38	39	37	36	3	1
4	1	45	44	41	42	2	1
1	2	43	44	41	42	2	1
2	1	39	41	42	40	2	1
1	2	1	2	2	3	1	1
1	2	1	3	1	1	4	2

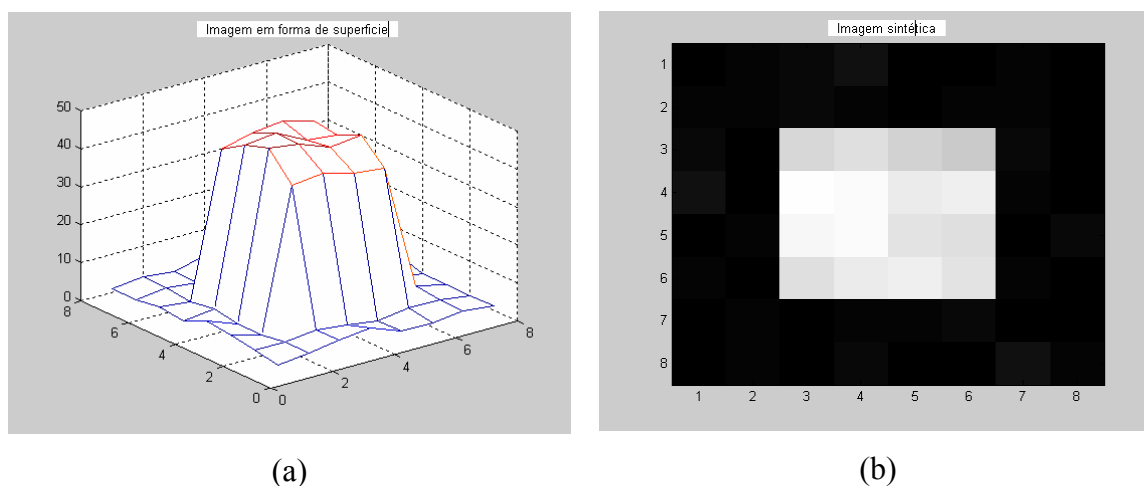


Figura 2.1 – (a) Imagem em forma de superfície. (b) Imagem sintética.

2.1 IMAGENS DIGITAIS – NOÇÕES DE ÓTICA

O principal componente de visão computacional é a *imagem digital*. Pode-se afirmar que existem dois tipos básicos de imagens digitais: a) imagem de intensidade – avalia-se a quantidade de luz capturada pelo dispositivo da câmera; e b) image de profundidade – estima-se diretamente a estrutura 3-D da cena (mundo real). As imagens digitais, tanto de intensidade quanto de profundidade, são matrizes 2-D.

2.1.1 Imagens de Intensidade

Uma fotografia é uma imagem de intensidade. Uma imagem de intensidade contém parâmetros físicos que estão diretamente relacionados com a imagem vista. São parâmetros físicos as características óticas das lentes, a fotometria e a geometria da imagem projetada. A

distância focal, o campo de visão e a abertura angular são as principais características óticas das lentes. Os parâmetros fotométricos estão relacionados à intensidade e direção da luminosidade; em outras palavras, referem-se às propriedades de reflexão da superfície vista em relação à quantidade de luz que atinge o fotorreceptor. Tipo de projeção, posição da câmera e distorções introduzidas no processamento da imagem são alguns parâmetros geométricos que interferem na posição da imagem no qual um ponto 3-D da cena é projetado.

2.1.2 Ótica Básica

A formação de uma imagem depende da quantidade de raios de luz que penetram a câmera pela abertura angular e chega ao plano de projeção da imagem. Um aparelho fotossensível na câmera registra essa intensidade de luz.

Foco da imagem

Um ponto de uma superfície recebe raios de luz de várias direções e conseqüentemente reflete esses raios, também em várias direções, e alguns desses raios podem entrar na câmera. Uma imagem é mais nítida quando esses raios convergem para um ponto no plano de projeção da imagem na câmera.

Pode-se conseguir maior nitidez em uma imagem reduzindo o tamanho do diafragma da câmera para um ponto, chamado de *pinhole*, ou introduzindo um sistema ótico composto de lentes e diafragmas que convirgam seus raios luminosos para um único ponto.

O tempo de exposição é também um fator importante na qualidade da imagem. Esse tempo é inversamente proporcional ao quadrado do diâmetro do diafragma. Assim, quanto menor o diâmetro do diafragma maior o tempo necessário para a exposição. Por outro lado, necessita-se de um mínimo de luz para registrar uma imagem visível, então uma redução do diafragma nem sempre é a alternativa mais eficaz para se obter melhor nitidez da imagem. Um sistema ótico permite uma abertura do diafragma maior num tempo menor de exposição, melhorando a capacidade de captar luminosidade.

Equação fundamental das lentes finas

A **Figura 2.2** mostra os principais elementos na construção das lentes finas. Onde F_L e F_R são chamados de foco esquerdo e direito respectivamente, e estão situados no eixo focal. Quando a lente tem curvaturas iguais nas duas superfícies, os focos têm distâncias iguais e, neste caso, é chamado de comprimento focal f .

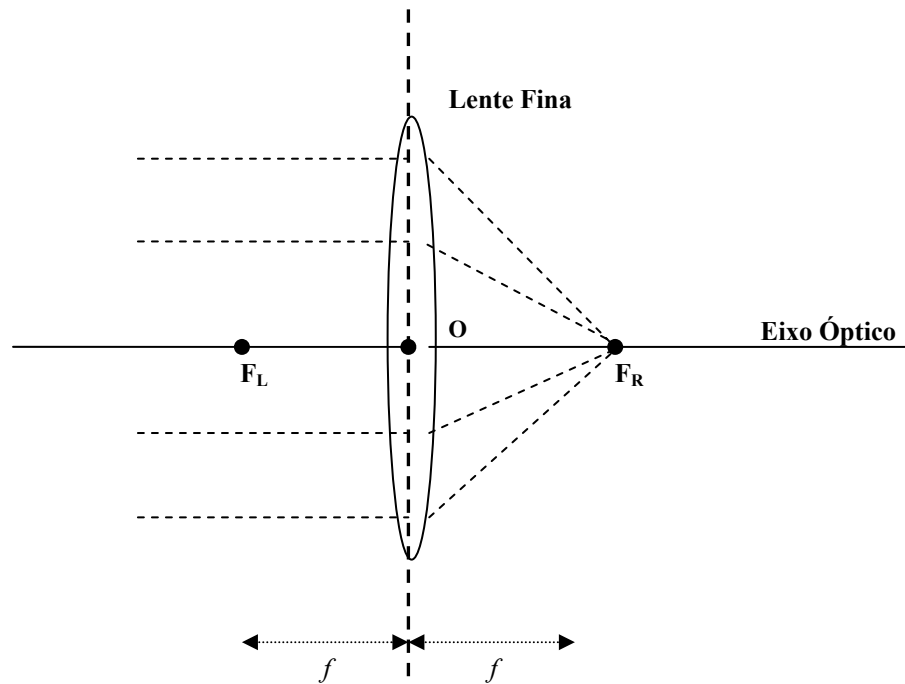


Figura 2.2 – Geometria ótica das lentes finas.

Duas propriedades básicas das lentes finas definem uma equação fundamental. A primeira, por construção das lentes, permite que qualquer raio de luz que entra na lente e é paralelo ao eixo ótico de um lado, passe pelo ponto focal do outro lado. A segunda, também por sua construção, permite que qualquer raio que entra na lente, passando pelo ponto focal de um lado, emerge paralelo ao eixo ótico do outro lado, como mostra a **Figura 2.3**.

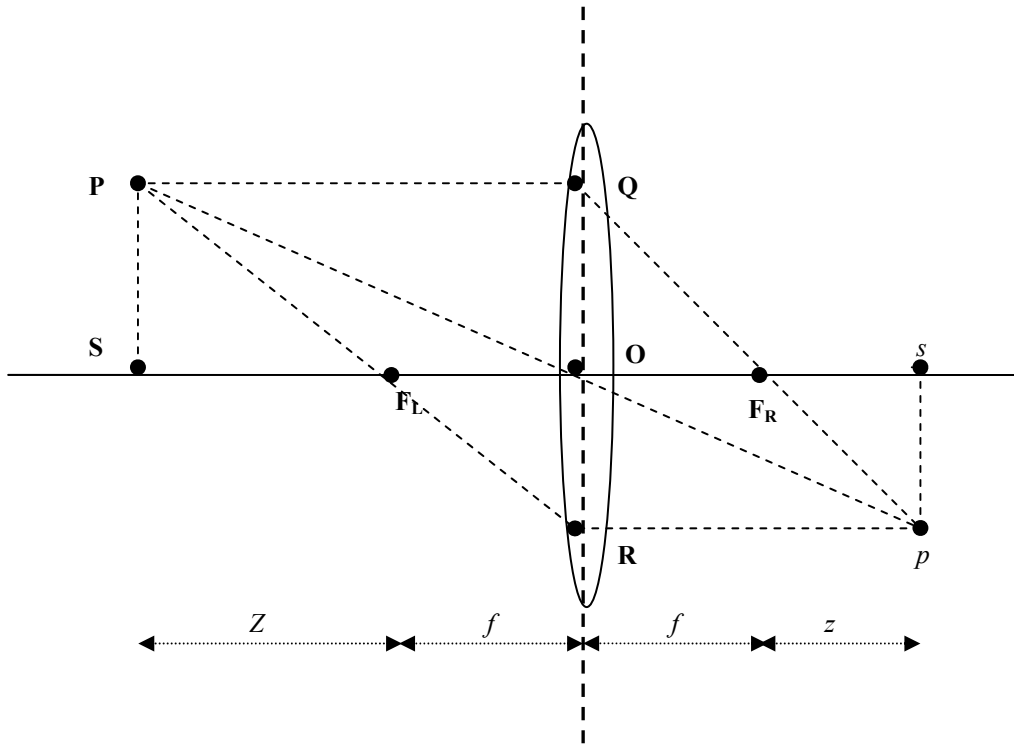


Figura 2.3 – Formação de imagens através das lentes finas.

Sejam os triângulos $\widehat{P\hat{S}F_L}$, $\widehat{R\hat{O}F_L}$, $\widehat{p\hat{s}F_R}$ e $\widehat{Q\hat{O}F_R}$. Pela teoria dos triângulos semelhantes pode-se demonstrar uma forma da equação fundamental das lentes, equação (2.1).

$$Z \cdot z = f^2 \quad (2.1)$$

Substituindo $\widehat{Z} = Z + f$ e $\widehat{z} = z + f$, obtém-se a equação fundamental das lentes, equação (2.2).

$$\frac{1}{\widehat{Z}} + \frac{1}{\widehat{z}} = \frac{1}{f} \quad (2.2)$$

2.1.3 Radiometria Básica

Radiometria é: “designação comum às técnicas de medidas de grandezas associadas à energia radiante” (Novo Dicionário Aurélio 15ª Edição), em outras palavras, é o estudo das

relações entre as quantidades de energia luminosa emitida de fontes de luz, refletida de superfícies e registrada por sensores.

Irradiança da imagem – Energia luminosa incidente, por unidade de área, e em cada ponto p do plano da imagem, medida em watt.

Radiança de imagem – Energia luminosa, por unidade de área, idealmente emitida por cada ponto P de uma superfície no espaço 3-D em uma dada direção d ao ponto p , medida em lumens.

Superfície lambertiana – Uma superfície é dita lambertiana se seus pontos refletem igualmente a luz incidente em todas as direções, ou seja, não absorve energia luminosa.

2.1.4 Formação Geométrica da Imagem

Para entender a formação geométrica de imagens é necessário conhecer o que significa uma transformação de perspectiva, também chamada de transformação de imagem. Essa transformação projeta pontos no espaço 3-D sobre um plano, e mostra uma aproximação de como uma imagem é formada ao se olhar para um mundo tridimensional.

- **Câmera Perspectiva**

O mais comum modelo geométrico para câmeras de intensidade é a perspectiva, também chamado de *modelo pinhole*. Esse modelo consiste de um plano π (plano da imagem), e um ponto \mathbf{O} em espaço 3-D (centro ou projeção do foco). A distância entre π e \mathbf{O} é chamado de comprimento focal f . A linha que passa por \mathbf{O} e é perpendicular a π é chamado de eixo ótico. O ponto \mathbf{o} onde o eixo ótico intercepta o plano da imagem é o centro de projeção da imagem, como mostra a **Figura 2.4**. O ponto \mathbf{p} é a imagem do ponto \mathbf{P} determinado pela intersecção do segmento de linha $\overline{\mathbf{OP}}$ com o plano da imagem π .

Considere o sistema de referência de coordenadas 3-D no qual \mathbf{O} é a origem e o plano π é ortogonal ao eixo Z, e sendo $\mathbf{P} = [X, Y, Z]^T$ e $\mathbf{p} = [x, y, z]^T$, é possível escrever as equações (2.3) da projeção perspectiva para as coordenadas da câmera.

$$x = f \frac{X}{Z} \quad \text{e} \quad y = f \frac{Y}{Z} \quad (2.3)$$

Note que se $z = f$, então o ponto \mathbf{p} pode ser escrito por $[x,y]^T$ ao invés de $[x,y,z]^T$.

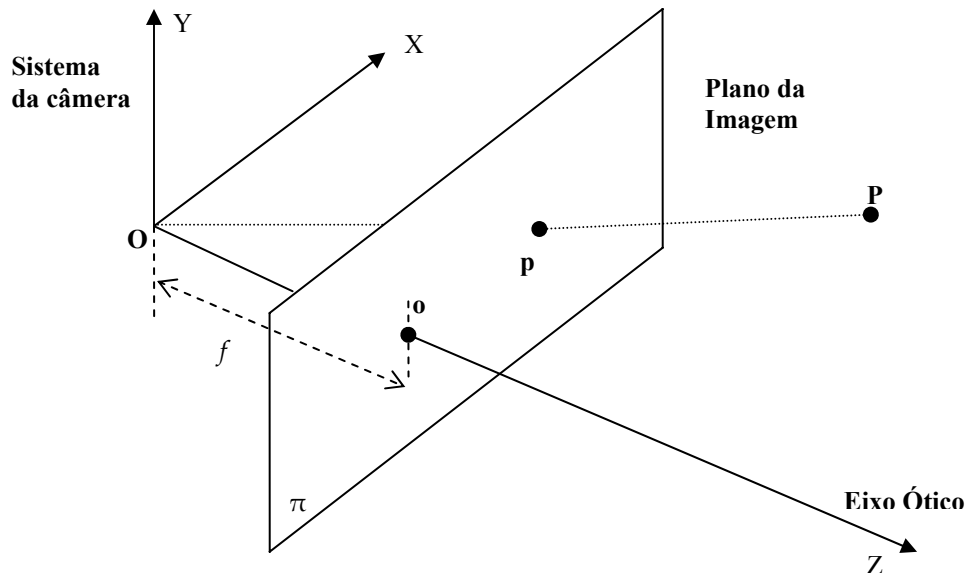


Figura 2.4 – Modelo câmera perspectiva.

- Câmera Perspectiva Fraca

As Equações 2.3 não são lineares devido ao fator $1/Z$. Transformá-las em equações lineares é o que se chama de modelo câmera perspectiva fraca. Para tanto, suponha que se a distância relativa ao longo do eixo óptico (Δz), de quaisquer dois pontos do objeto no espaço 3-D (profundidade) seja muito menor que a distância média \bar{Z} desses pontos ao plano 2-D da imagem (na câmera), então é possível reescrevê-las para obter as equações (2.4) lineares.

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

Se a variação de $\Delta z < \bar{Z}/20$ (Trucco *et al*, 1998), pode-se considerar uma aproximação linear das Equações 2.3, também chamada de projeção ortográfica. Neste caso, significa que quaisquer pontos são projetados ao longo de raios paralelos ao eixo óptico, tomando $\frac{f}{Z} \rightarrow 1$.

2.2 PARÂMETROS DE CÂMERAS

Para conhecer a posição de objetos no espaço, necessita-se de equações que relacionem as coordenadas dos pontos no sistema 3-D com as coordenadas dos pontos correspondentes na imagem da câmera. As Equações 2.3 definem bem essa situação e assumem que:

- Pode-se relacionar o sistema de referência da câmera com algum outro sistema conhecido, por exemplo, sistema de coordenadas externa.
- As coordenadas dos pontos da imagem no sistema de referência da câmera podem ser obtidas através das coordenadas em pixels.

Algumas características das câmeras são chamadas em visão computacional como *parâmetros extrínsecos e intrínsecos*.

Parâmetros extrínsecos – São parâmetros que definem a localização e orientação do sistema de referência da câmera em relação a um sistema de referência conhecido.

Parâmetros intrínsecos – São parâmetros que relacionam as coordenadas em pixels de uma imagem com as coordenadas correspondentes no sistema de referência da câmera.

2.2.1 Parâmetros Extrínsecos

O sistema de referência da câmera tem sido mencionado com o objetivo de escrever as equações fundamentais da projeção perspectiva, equações (2.3), numa forma mais simples. Normalmente o sistema de referência da câmera é desconhecido, então o problema é conhecer a localização e a orientação do sistema da câmera em relação a um sistema conhecido, utilizando-se apenas dados da imagem.

Trucco *et al* (1989), define parâmetros extrínsecos como sendo *qualquer conjunto de parâmetros geométricos que identifiquem univocamente as transformações entre o sistema de referência da câmera desconhecido e um sistema conhecido, chamado de sistema de coordenadas externo*.

Para essas transformações utilizam-se normalmente um vetor de translação, \mathbf{T} , e uma matriz de rotação, \mathbf{R} . O vetor 3-D de translação descreve a posição relativa das origens dos dois sistemas. A matriz de rotação é uma matriz 3x3 ortogonal, ou seja, $\mathbf{R}^T \cdot \mathbf{R} = \mathbf{R} \cdot \mathbf{R}^T = \mathbf{I}$, que permite o alinhamento entre os eixos dos dois sistemas. A **Figura 2.5** mostra a relação entre as coordenadas de um ponto no sistema 3-D qualquer ($\mathbf{P}_W = [25 \ 35 \ 19]^T$), e as coordenadas desse ponto no sistema de referência da câmera ($\mathbf{P}_C = [15 \ -17 \ 26]^T$), também determinado pela equação (2.5).

$$\mathbf{P}_C = \mathbf{R} \cdot (\mathbf{P}_W - \mathbf{T}) \quad (2.5)$$

, onde $\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$, é o produto de três matrizes de rotação ao longo dos eixos X, Y e Z.

Essas matrizes de rotação são definidas como:

$$\mathbf{R}_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}, \quad \mathbf{R}_y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}$$

$$\text{e } \mathbf{R}_z(\gamma) = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

No exemplo apresentado na Figura 2.5, a rotação se dá no eixo Y em 180°, gerando a matriz

de rotação e o vetor de translação igual a $\mathbf{R} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$ e $\mathbf{T} = [40 \ 52 \ 45]$,

respectivamente.

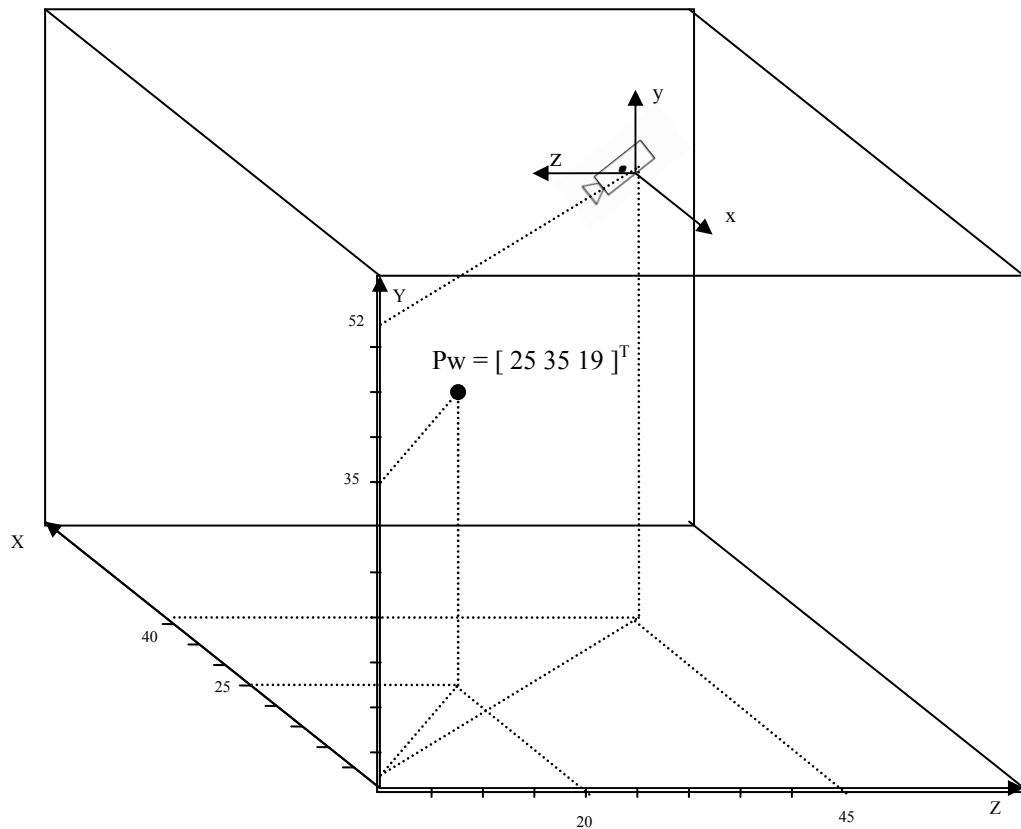


Figura 2.5 – Exemplo de transformação entre dois sistemas – Mundo (P_w) e Câmera (P_c).

2.2.2 Parâmetros Intrínsecos

São os parâmetros que definem as características óticas, geométricas e digitais da câmera.

Para o modelo pinhole, os parâmetros intrínsecos são:

- A projeção perspectiva da qual o único parâmetro é a distância focal f ;
- A transformação entre os sistemas da câmera e o sistema da imagem (pixel); e
- As distorções geométricas introduzidas pelo sistema ótico.

Para encontrar o conjunto de parâmetros intrínsecos, devem-se relacionar as coordenadas (X_{im}, Y_{im}) de um ponto da imagem em unidade de pixel com as coordenadas (X, Y) do mesmo ponto no sistema de referência da câmera, equação (2.6), considerando que a imagem não tenha distorção geométrica, e que o sensor CCD seja um “grid” retangular de elementos fotossensíveis.

$$X = -(X_{im} - O_x) \cdot S_x \quad \text{e} \quad Y = -(Y_{im} - O_y) \cdot S_y \quad (2.6)$$

, onde (O_x, O_y) são as coordenadas do centro da imagem em pixel e (S_x, S_y) o tamanho dos pixels.

São parâmetros intrínsecos a distância focal f , as coordenadas do centro da imagem em pixel e o tamanho efetivo dos pixels nas direções horizontais e verticais. O coeficiente de distorção radial, também é um parâmetro intrínseco.

Resumindo, a equação (2.6) é a equação matricial da projeção perspectiva e pode ser reescrita na forma da equação (2.7).

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = M_{\text{int}} \cdot M_{\text{ext}} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \text{ onde } M_{\text{int}} = \begin{pmatrix} -f/S_x & 0 & O_x \\ 0 & -f/S_y & O_y \\ 0 & 0 & 1 \end{pmatrix} \text{ e } M_{\text{ext}} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & -\mathbf{R}_1^T \cdot \mathbf{T} \\ r_{21} & r_{22} & r_{23} & -\mathbf{R}_2^T \cdot \mathbf{T} \\ r_{31} & r_{32} & r_{33} & -\mathbf{R}_3^T \cdot \mathbf{T} \end{pmatrix} \quad (2.7)$$

, onde \mathbf{R}_i ($i = 1, 2, 3$) é o vetor 3-D formado pela i -ésima linha da matriz \mathbf{R} .

2.3 FILTRAGEM DE RUÍDOS

Ruído é qualquer detalhe na imagem, dado ou resultado intermediário, que não seja interessante aos propósitos do cálculo principal. Imagine duas imagens de uma mesma cena, adquiridas por uma mesma câmera e com a mesma intensidade de luz, em que os valores de pixels não sejam iguais. Esses valores diferentes podem introduzir erros nos resultados de cálculos. Neste caso é muito importante dimensionar a magnitude desse erro, também chamado de ruído.

Em algoritmos de processamento de imagens, como por exemplo, na detecção de linhas ou contornos, considera-se ruído a flutuação de pixel espúrios, normalmente introduzidos pelo sistema de aquisição de imagem. É possível reduzir a magnitude do ruído, através de técnicas chamadas de filtragem, sem alterar significativamente a qualidade da imagem.

2.3.1 Filtro Linear

Filtragem linear é uma técnica que tem como finalidade atenuar ou eliminar o ruído de uma imagem, e consiste, no domínio da frequência, em efetuar uma convolução entre os valores dos pixels da imagem e uma matriz (A) chamada de máscara, conforme mostra a equação (2.8), onde uma imagem I de dimensões NxM, tem sua versão filtrada I_A através do uso da convolução discreta com a máscara A de dimensões $m \times m$, sendo m um valor muito menor que N e M (Cai, 1990).

$$I_A(i, j) = I * A = \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} A(h, k) I(i-h, j-k) \quad (2.8)$$

*indica convolução e $\frac{m}{2}$ é o valor inteiro da divisão.

Filtros lineares baseiam-se em que a função de transferência e o impulso são Transformadas de Fourier inversas uma da outra. Os denominados filtros *passa-baixas* atenuam ou eliminam os componentes de alta frequência no domínio de Fourier enquanto deixam as baixas frequências inalteradas. Os componentes de alta frequência caracterizam bordas ou outros detalhes finos de uma imagem, de forma que o efeito resultante da filtragem *passa-baixas* é o borramento da imagem. Do mesmo modo, filtros *passa-altas* atenuam ou eliminam os componentes de baixa frequência, e como esses componentes são responsáveis pelas características que variam lentamente em uma imagem (contraste total e a intensidade média), o resultado dessa filtragem é a redução destas características, ou seja, um afinamento das bordas e outros detalhes finos. Por último, os filtros *passa-banda*, cuja característica é remover regiões selecionadas entre frequências altas e baixas, são mais utilizados em restauração de imagens, tendo pouca aplicação na eliminação ou atenuação de ruídos para realce das imagens (Gonzalez *et al*, 2000).

Atenuação pela média

Nos filtros *passa-baixas* todos os coeficientes de suas máscaras devem ser positivos. Uma máscara para essa filtragem teria seus componentes iguais a 1 (um). Como a convolução resulta em valores diferentes do previsto em escala de cinza (0 a 255), faz-se necessário que se obtenha um valor médio, e neste caso é necessário dividir o resultado pela quantidade de elementos da máscara. Essas máscaras podem ser:

$$\mathbf{A}_{\text{media 1}} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ ou } \mathbf{A}_{\text{media 2}} = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

A **Figura 2.6a** apresenta uma imagem em sua forma original. A **Figura 2.6b** apresenta a mesma imagem após filtragem com uma máscara $\mathbf{A}_{\text{media 1}}$. A **Figura 2.6c** apresenta a mesma imagem após filtragem com máscara $\mathbf{A}_{\text{media 2}}$.

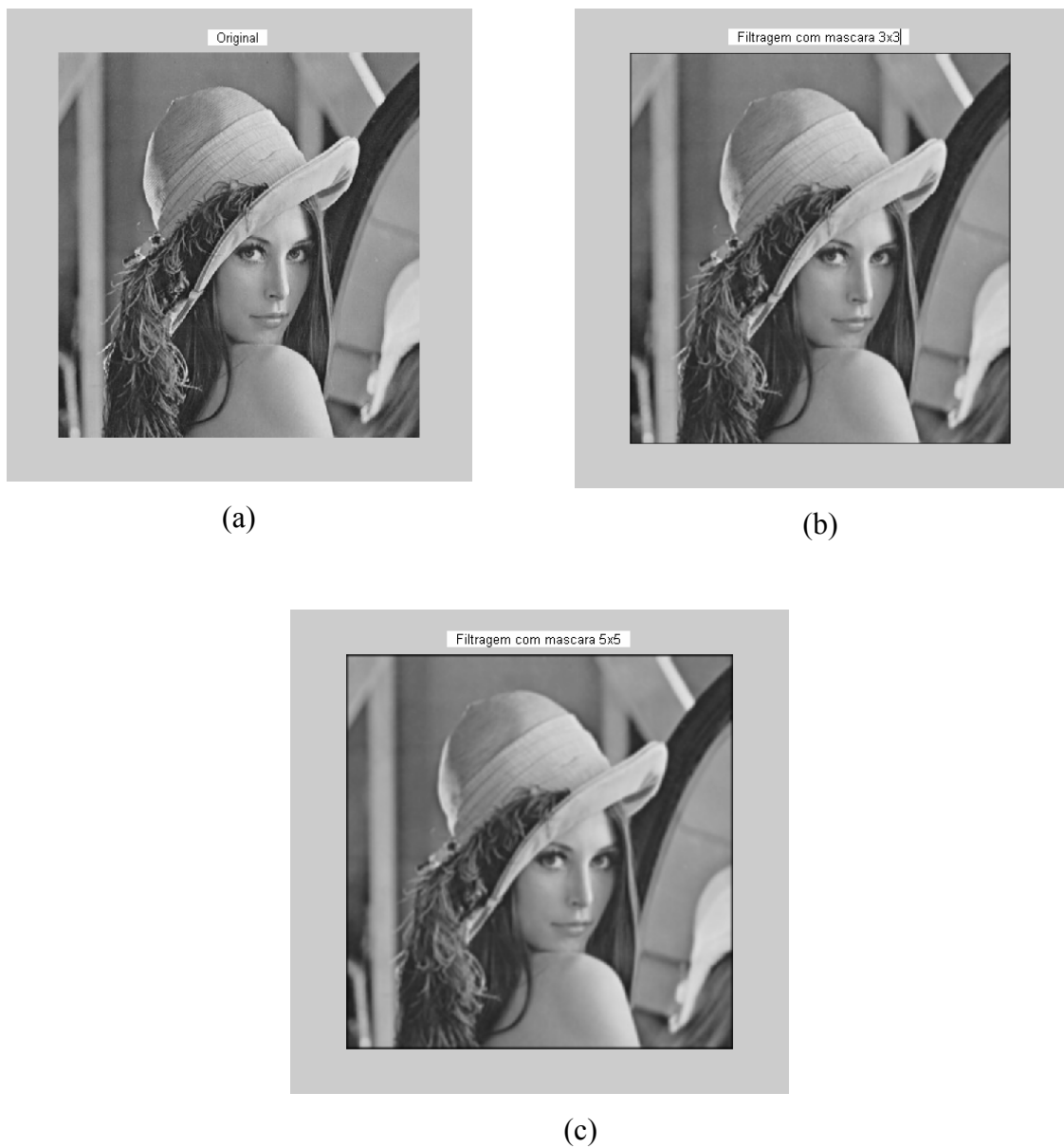


Figura 2.6 – (a) Imagem original, (b) Imagem Atenuada com máscara $\mathbf{A}_{\text{media 1}}$, (c) Imagem atenuada com máscara $\mathbf{A}_{\text{media 2}}$.

Atenuação gaussiana

Atenuação gaussiana é um caso particular da atenuação pela média, onde a máscara é um 2-D gaussiano. Neste caso a convolução da imagem com a máscara gaussiana 2-D é igual à convolução de todas as linhas da imagem com a máscara gaussiana 1-D somada à convolução de todas as colunas da imagem com a máscara 1-D, desde que ambas tenham o mesmo desvio padrão. A equação (2.9) mostra a separabilidade das máscaras gaussianas.

$$\begin{aligned} I_G(i, j) &= I * G = \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} G(h, k) I(i-h, j-k) \\ &= \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} e^{-\frac{h^2+k^2}{2\sigma^2}} I(i-h, j-k) = \\ &= \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} e^{-\frac{h^2}{2\sigma^2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} e^{-\frac{k^2}{2\sigma^2}} I(i-h, j-k) \end{aligned} \quad (2.9)$$

Para a construção da máscara gaussiana deve-se conhecer o desvio padrão σ para obter o tamanho da máscara. Segundo Trucco *et al* (1998), a relação usada em geral é $\sigma_w = w/5$, onde w define a largura da máscara. Para a amostragem ser gaussiana, no intervalo de $[-\pi, \pi]$, σ_w deve ser um valor maior ou igual a 0.8, logo a menor largura da máscara deve ser 5.

A **Figura 2.7a** apresenta a imagem em sua forma original. A **Figura 2.7b** mostra a mesma imagem com atenuação gaussiana, com desvio padrão $\sigma = 1,5$.

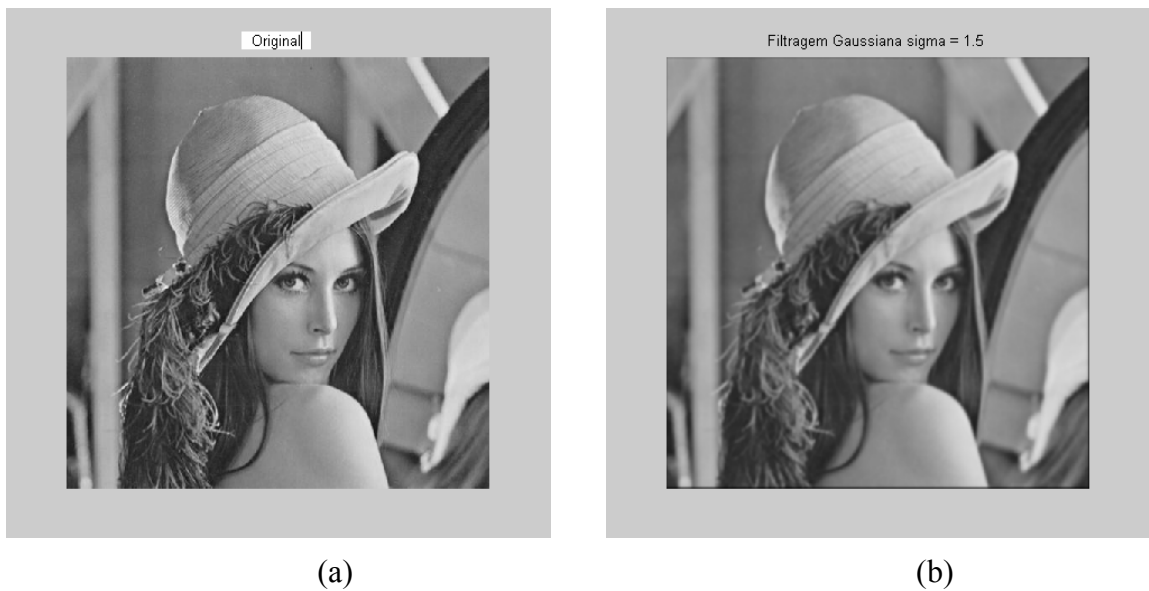


Figura 2.7 – (a) Imagem original. (b) Imagem com atenuação gaussiana $\sigma = 1.5$.

2.4 CARACTERÍSTICAS DAS IMAGENS

As características mais importantes para análise de imagem de peças industriais são as fronteiras e as medidas geométricas derivadas dessas fronteiras. Essas características são agrupadas em três tipos: *Global*, *Local* e *Característica relacional* (Chin *et al*, 1986). São exemplos de características globais, o perímetro, o centro de massa, distâncias entre dois pontos da fronteira e o centro de massa, curvatura, área e momento de inércia. É exemplo de características locais o segmento de linha, segmento de arco com curvatura constante e vértices que definem fronteiras dos objetos. Exemplos de características relacionais são as medidas de distância e a sua orientação relativa que interligam subestruturas e região de um objeto.

Antes de iniciar o processo de extração de características deve-se obter uma imagem binarizada de uma imagem original em escala de cinza, e para isso é necessário escolher um limiar apropriado, ou um sensor que produza imagem binarizada. Uma imagem binarizada reduz a complexidade dos dados que serão trabalhados.

Vários algoritmos utilizados na extração de características em imagens binarizadas são para obtenção de contornos ou fronteiras. O problema é que esses algoritmos detectam fronteiras de objetos planos e são normalmente falhos na obtenção de fronteiras em superfícies. Apesar

dos sistemas serem bastante sofisticados na extração de características, seus métodos são aplicados em modelos 2-D.

São características de imagens os contornos, vértices, linhas, cônicas e outros. Neste trabalho será dada ênfase aos vértices.

2.4.1 Contorno

Várias interpretações de imagens são baseadas em contornos, ou melhor, onde a imagem possui contraste. O contraste para visão humana significa a capacidade de perceber o perímetro de um objeto, desde que esse objeto tenha diferentes intensidades de luz à sua volta. Essencialmente, a fronteira de um objeto é uma mudança de valor nos níveis da imagem de intensidade. Um contorno é detectado pela derivada de primeira ordem, localizando assim regiões onde ocorre uma diferença de intensidade ou um máximo local.

Em visão computacional, a detecção de uma fronteira é um processo de três passos:

Atenuar ruídos – Suprimir o máximo de ruídos sem comprometimento do contorno;

Realçar contorno – Projetar um filtro que tenha maior saída nos pontos do contorno que em outras regiões. Assim o contorno é o máximo local; e

Localizar o contorno – Estabelecer, por um limiar, quais os máximos que significam contorno. Os contornos devem estar bem afilados.

“Canny Edge Detector”

O operador *Canny Edge Detector* é uma técnica para detecção de fronteira em visão computacional. Seus principais objetivos são:

Detecção ótima – Imagem de intensidade sem dados espúrios ou ruídos;

Boa localização – Detectar arestas o mais próximo possível da fronteira real; e

Resposta única – O operador retorna apenas um ponto para cada ponto do contorno verdadeiro.

Canny [1986], mostra em seu trabalho os passos e o desenvolvimento completo do algoritmo do operador *canny edge detector* para a extração de contornos.

A **Figura 2.8a** apresenta a figura original. A **Figura 2.8b** apresenta o contorno da imagem utilizando-se do algoritmo de *Canny Edge Detector*.



Figura 2.8 – (a) Imagem original. (b) Contorno extraído por *Canny Edge detector*.

Outros Operadores – Robert e Sobel

Ambos os operadores *Robert* e *Sobel* são mais simples de executar que o operador *canny edge detector*, pois suas formulações matemáticas são menos complexas. Em ambos os casos seus algoritmos seguem 4 passos bem definidos, alterando apenas a máscara aplicada na filtragem:

- 1- Aplicar um filtro adequado para atenuar o ruído;
- 2- Filtrar a imagem sem ruídos com as máscaras – A_{R1} e A_{R2} (Robert) ou – A_{S1} e A_{S2} (Sobel), obtendo duas imagens I_1 e I_2 , respectivamente;

$$A_{R1} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad A_{R2} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{e} \quad A_{S1} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad A_{S2} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- 3- Calcular a magnitude do gradiente, equação (2.10), para cada pixel (i,j) ;

$$G(\mathbf{i}, \mathbf{j}) = \sqrt{I_1^2(\mathbf{i}, \mathbf{j}) + I_2^2(\mathbf{i}, \mathbf{j})} \quad (2.10)$$

4- Selecionar como contorno todos os pixels com valor de gradiente maior que um limiar – (τ).

A **Figura 2.9a** apresenta a imagem original. A **Figura 2.9b** apresenta o contorno da imagem utilizando o operador Robert com limiar $\tau = 34.88$. A **Figura 2.9c** apresenta o contorno da imagem utilizando o operador Sobel com limiar $\tau = 78.4$.

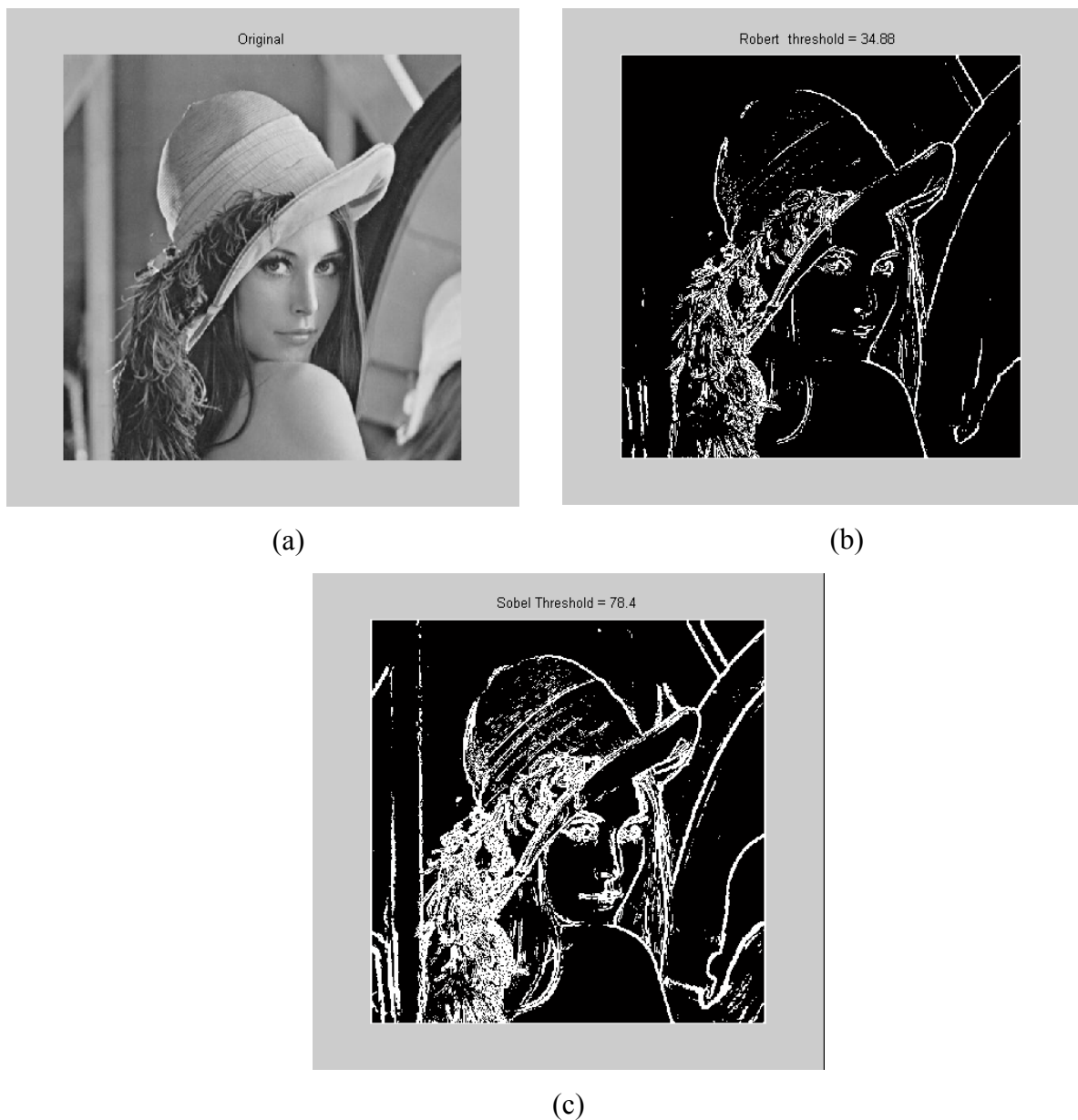


Figura 2.9 – (a) Imagem original. (b) Extração de contorno por Robert – $\tau = 34.88$. (c) Extração de contorno por Sobel – $\tau = 78.4$.

2.4.2 Vértices

Vértice é definido por Deriche (1993) como sendo pontos pertencentes ao contorno da imagem onde ocorre um máximo local de curvatura, ou a interseção de 2 ou mais arestas do contorno.

Existem várias técnicas para extração de vértices. Vicente et al. (2002) classificam em dois grupos: o primeiro consiste na operação sobre a imagem do objeto com a extração prévia do contorno; e o segundo grupo consiste em operação diretamente nos valores dos pixels em escala de cinza. As técnicas mais clássicas são de Beaudet, Kitchen, Noble e Trucco.

Técnica de extração de vértices por Beaudet

Em Beaudet (1978), foi proposto o operador invariante rotacional, chamado DET, equação (2.11). Este operador é obtido usando a segunda ordem da expansão de Taylor na imagem de intensidade $I(i, j)$.

$$DET = I_{xx} \cdot I_{yy} - I_{xy}^2 \quad (2.11)$$

Os vértices são detectados tomando um valor limite entre os valores extremos desse operador. DET é estimado como o determinante da matriz Hessiana H , que está relacionado com o produto das curvaturas principais $k_{min} \cdot k_{max}$, equação (2.12), chamada de curvatura gaussiana:

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} \quad \text{e} \quad k_{min} \cdot k_{max} = \frac{DET}{1 + I_x^2 + I_y^2} \quad (2.12)$$

Então, para o pixel $I(i,j)$ se $k_{min} \cdot k_{max} = 0$, o pixel é parabólico; se $k_{min} \cdot k_{max} > 0$, o pixel é elíptico; e se $k_{min} \cdot k_{max} < 0$, o pixel é hiperbólico. Como o denominador da equação (2.12) é sempre positivo, o sinal da curvatura gaussiana tem o mesmo sinal de DET. Localizam-se os vértices onde DET tem valor positivo ou negativo em ambos os lados do contorno.

Técnica de extração de vértices por Kitchen

Em Kitchen (1982), foi proposto que para se determinarem os vértices, basta localizar onde ocorre a mudança de direção do gradiente ao longo do contorno, multiplicando esse ponto pela magnitude do gradiente local. Os vértices são identificados pelos valores máximos de K , equação (2.13).

$$K = \frac{I_{xx} \cdot I_y^2 + I_{yy} \cdot I_x^2 - 2 \cdot I_{xy} \cdot I_x \cdot I_y}{I_x^2 + I_y^2} \quad (2.13)$$

Técnica de extração de vértices por Noble

Noble (1982) utiliza-se da teoria de geometria diferencial para determinação de vértices.

Técnica de extração de vértices por Trucco

Em Trucco et al, (1998), faz-se uso da técnica do gradiente espacial da imagem $[E_x, E_y]^T$. A extração de vértices é feita utilizando-se do menor autovalor de uma matriz C , equação (2.14), que considera um ponto genérico p na imagem e é calculado na sua vizinhança Q $2N+1 \times 2N+1$ píxeis (por exemplo, 5×5), definida por Trucco *et al*, (1998):

$$C = \begin{bmatrix} \sum E_x^2 & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^2 \end{bmatrix}, \text{ onde } E_x = \frac{\partial E}{\partial x} \text{ e } E_y = \frac{\partial E}{\partial y} \quad (2.14)$$

Pode-se adotar um valor de limiar τ , determinado como uma porcentagem dos maiores valores dentre os menores autovalores de C . Isso garante que em duas direções de um ponto têm-se valores de gradiente suficientemente altos para caracterizar um vértice.

A **Figura 2.10a** apresenta a imagem original. A **Figura 2.10b** apresenta a imagem binarizada com limiar igual a 170. A **Figura 2.10c** apresenta os vértices da imagem binarizada, utilizando a técnica apresentada por Trucco.

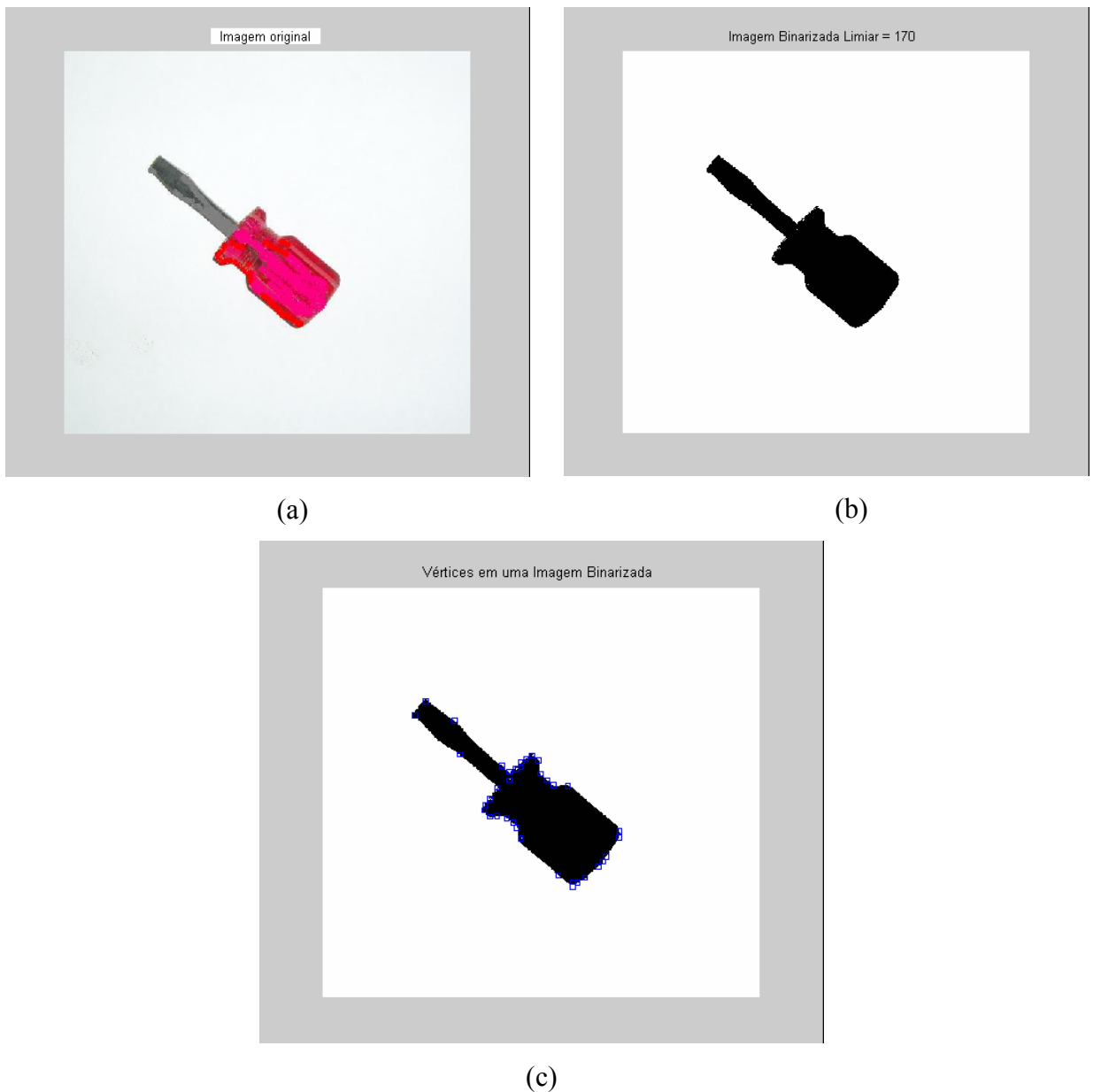


Figura 2.10 – (a) Imagem original. (b) Imagem binarizada com limiar = 170. (c) Vértices de imagem binarizada.

2.4.3 Centro de Massa

Uma boa característica de imagens é aquela que pode ser localizada sem ambigüidades em diferentes vistas da mesma cena.

Considerando imagens binárias, a função característica $f(i,j)$ assume valores 1 (um) para pontos do objeto ou 0 (zero) para os demais (background).

Assim, a área de um objeto pode ser obtida em sua forma discreta pela equação (2.15):

$$A = \sum_I \sum f(i, j) \quad (2.15)$$

O centro de área é o centro geométrico da figura com a mesma forma e com massa constante por unidade de área. O centro de massa é o ponto onde toda a massa do objeto pode ser concentrada, sem que com isso se altere o momento da primeira ordem relativamente a qualquer um dos eixos.

Numa imagem de intensidade 2-D, o momento de primeira ordem em torno de i e j é dado pela equação (2.16).

$$\begin{aligned} \bar{i} \cdot \sum_I \sum f(i, j) &= \sum_I \sum i \cdot f(i, j) \\ \bar{j} \cdot \sum_I \sum f(i, j) &= \sum_I \sum j \cdot f(i, j) \end{aligned} \quad (2.16)$$

, onde (\bar{i}, \bar{j}) é a posição do centro de área. As equações 2.16 podem ser reescritas na forma das equações (2.17).

$$\begin{aligned} \bar{i} &= \frac{\sum_I \sum i \cdot f(i, j)}{A} \\ \bar{j} &= \frac{\sum_I \sum j \cdot f(i, j)}{A} \end{aligned} \quad (2.17)$$

A **Figura 2.10** apresenta a imagem de um objeto e seu centro de massa em duas posições diferentes. A **Figura 2.10a** apresenta a imagem e seu centro de massa. A **Figura 2.10b** apresenta a mesma imagem anterior, porém com uma rotação de 180° , e seu centro de massa.

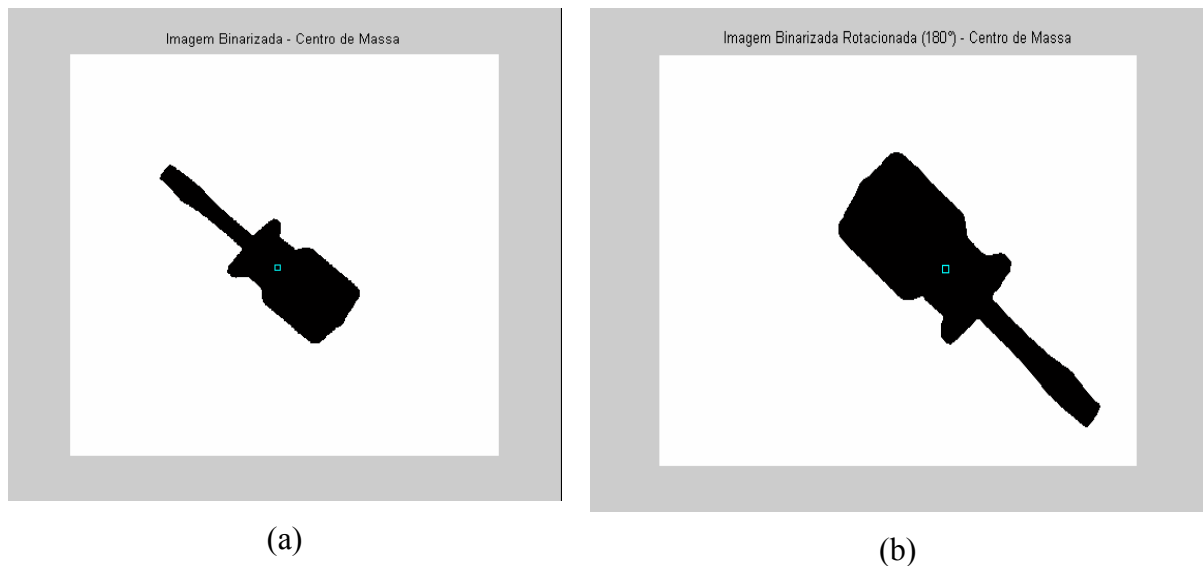
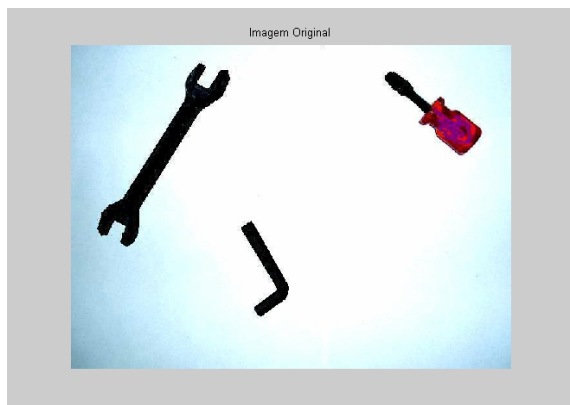


Figura 2.11 – (a) Imagem binarizada com Centro de Massa. (b) Imagem Binarizada Rotacionada com Centro de Massa.

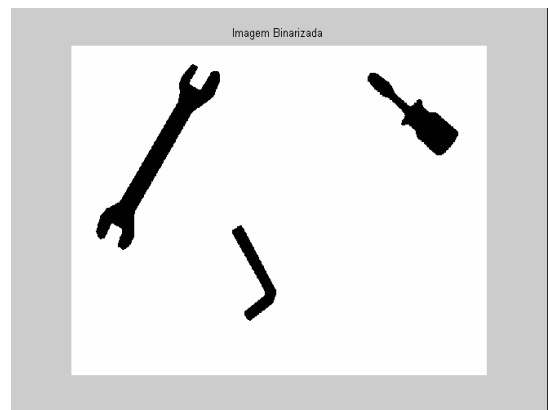
2.4.4 Segmentação de Imagem

O processo de segmentação de imagens consiste em separar a imagem nos seus diversos componentes, e assim separar os diversos objetos que nela existem. Esta separação consiste em agrupar todos os pixels que pertencem a um dado objeto ou região, através de um critério de semelhança, por exemplo, sua intensidade.

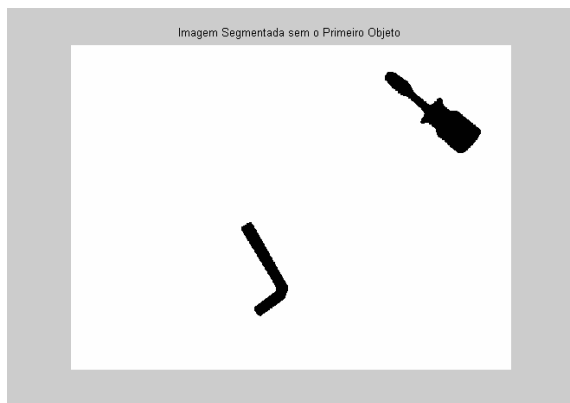
A **Figura 2.10** apresenta uma imagem contendo vários objetos. A **Figura 2.10a** apresenta a imagem original. A **Figura 2.10b** apresenta a imagem original binarizada. A **Figura 2.10c** apresenta a imagem original sem o primeiro objeto. A **Figura 2.10d** apresenta a imagem original sem o primeiro e o segundo objetos.



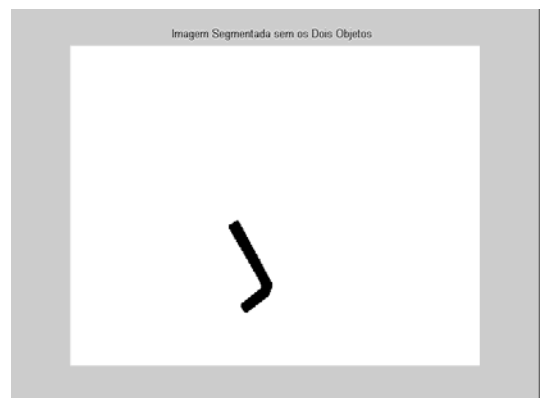
(a)



(b)



(c)



(d)

Figura 2.12 – (a) Imagem original (b) Imagem Binarizada. (c) Imagem Segmentada sem o primeiro objeto. (d) Imagem segmentada sem dois objetos.

3. RECONHECIMENTO DE OBJETOS – IDENTIFICAÇÃO

Reconhecimento de objetos significa que seus descritores ou simplesmente modelos estejam disponíveis, ou melhor, reconhecer objetos significa que se conhece previamente sua geometria. Não há como reconhecer aquilo que é desconhecido, ou seja, que não tenha sido previamente armazenado em um banco de dados. Em visão computacional esta classe de reconhecimento chama-se *reconhecimento baseado em modelos*, onde a idéia central está na comparação entre uma imagem plana 2-D e uma base de dados armazenada de modelos.

Reconhecimento de objetos envolve duas operações básicas: *Identificação* e *Localização*. Este capítulo é destinado às técnicas de *Identificação* de objetos. Identificação determina a natureza e localização determina a posição 3-D do objeto na imagem. As técnicas mais clássicas para identificação de objetos utilizando-se de modelos são: *Árvore de interpretação*, *Identificação baseado em aparências* e *Invariantes* (Trucco *et al*, 1998).

- Árvores de interpretação

Essa técnica engloba a classe de algoritmos que se baseiam na identificação de características. São algoritmos que identificam partes de um objeto em uma imagem, ou seja, identifica todos os subconjuntos da imagem que tenham forma de um modelo conhecido. Enfim, árvore de interpretação consiste em identificar as formas de um objeto o mais consistente possível, sem ser necessário efetuar todas as combinações entre estas formas, pois isso seria impraticável no âmbito computacional.

- Identificação baseado em aparências

Essa técnica engloba a classe de algoritmos que para identificar objetos, utilizam-se de imagens desses objetos e não de suas características. São algoritmos que identificam se em partes de uma imagem encontram-se partes de um determinado objeto. A idéia central dessa técnica consiste em simplesmente identificar um objeto por sua aparência. Para tanto deve armazenar um conjunto de imagens, em diversos ângulos, que representem as aparências do objeto modelo. Assim um objeto de uma determinada imagem poderá ser identificado por comparação.

- Invariantes

Essa técnica visa uma classe de algoritmos que identifica objetos, utilizando-se de características geométricas invariantes de modelos previamente armazenados em um banco de dados. São algoritmos que identificam objetos ou partes deles, em uma imagem. A idéia central dessa classe de algoritmos consiste em armazenar as características geométricas invariantes dos objetos, para sua posterior identificação em uma determinada imagem.

O foco deste trabalho está na classe de algoritmos de identificação de objetos por invariantes, e os pontos característicos utilizados na construção da geometria dos objetos serão seus vértices.

3.1 Projetiva Invariante para Geração de Modelos

Invariantes são propriedades funcionais de configurações geométricas que não se alteram sob certas classes de transformações, por exemplo, o comprimento de um segmento de linha não varia em função de uma rotação ou translação. Nas transformações usuais, existem invariantes bem conhecidas como comprimentos e áreas – transformações Euclidianas, (Vicente *et al*, 2002) ou razão de comprimentos e ângulos – transformações por similaridades (Hartley *et al*, 2000). Em transformações projetivas ocorrem também invariantes, onde a mais conhecida é a razão das razões de comprimentos (cross-ratio).

Uma grande vantagem na utilização de invariantes está na facilidade de aquisição de modelos para geração da base de dados, bastando para isso adquirir uma única imagem do objeto real e gerar um vetor de invariantes que será armazenado como índice do objeto.

Assumindo que todos os objetos são planares, afirma-se que a transformação de imagens é uma transformação entre planos. Assim, pode-se considerar diretamente o mapeamento entre pontos correspondentes do plano do objeto e o plano da imagem. O mapeamento é construído por uma transformação projetiva, daí o nome: *projetiva invariante*.

Seja $\mathbf{p} = [x,y,1]^T$ as coordenadas homogêneas de um ponto na imagem e $\mathbf{P} = [X,Y,1]^T$ as coordenadas homogêneas dos pontos correspondentes no plano real. A transformação projetiva que mapeia \mathbf{P} em \mathbf{p} pode ser escrita pela equação (3.1).

$$k \cdot p = T \cdot P \quad (3.1)$$

, onde T (matriz projetiva 3x3) e k são definidos como um fator de escala.

Nessa seção serão apresentadas invariantes projetivas construídas em função de pontos característicos da imagem (vértices). Essas invariantes são calculadas utilizando 5 vértices da imagem ($p_i \quad i= 1,2,3,4,5$) em coordenadas homogêneas (não colineares). Vários trabalhos (Trucco *et al.*, 1998; Song *et al.*, 2000 e Vicente *et al.*, 2002) mostram duas projetivas invariantes independentes I_1 e I_2 definidas pelas equações (3.2):

$$I_1 = \frac{S_{431} \cdot S_{521}}{S_{421} \cdot S_{531}} \quad I_2 = \frac{S_{421} \cdot S_{532}}{S_{432} \cdot S_{521}} \quad (3.2)$$

, onde S_{ijk} é o determinante da matrix 3x3 composta pelos pontos $|p_i \quad p_j \quad p_k|$.

3.2 Configuração Geométrica do modelo

Para a construção dos modelos de objetos aplica-se o algoritmo para determinação de vértices, descrito no item 2.4.2 (Técnica de extração de vértices por Trucco), sobre a imagem do objeto, e seleciona-se um conjunto desses vértices cujo determinante associado seja diferente de zero, o que evita valores de I_1 e I_2 indesejáveis (zero ou indefinido). Para tanto, deve-se evitar que 3 de cada grupo de 5 pontos sejam colineares. O conjunto selecionado deve ter a forma geométrica do objeto modelo.

Por exemplo, dois modelos de objetos são apresentados nas **Figura 3.1** e **Figura 3.2**. A escolha se deu pela complexidade geométrica da peça, mais próximos de objetos industriais comuns.

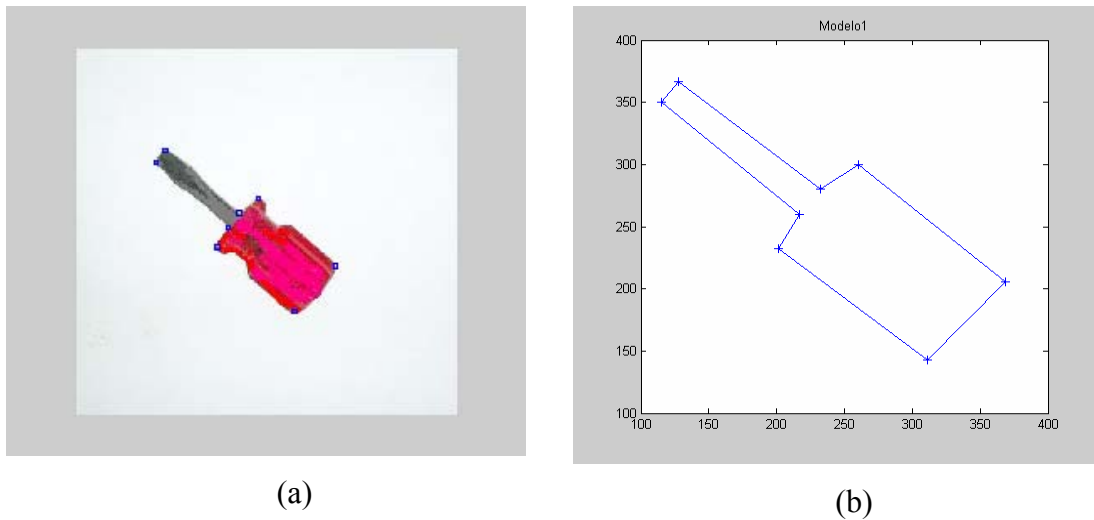


Figura 3.1 – (a) Imagem do modelo 1 - Apresentação da imagem do objeto modelo 1 com os vértices; (b) Geometria do modelo 1.

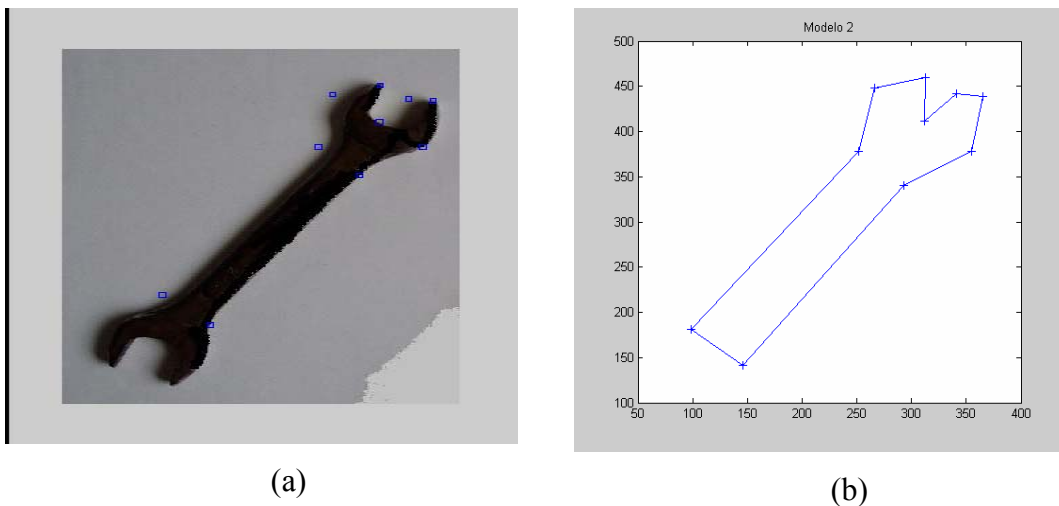


Figura 3.2 – (a) Imagem do modelo 2. (a) - Apresentação da imagem do objeto modelo 2 com os vértices; (b) Geometria do modelo 2.

A seqüência dos pontos extraídos do conjunto de vértices, obtidos na imagem original, deve seguir uma ordenação de maneira que forme um polígono fechado e numa seqüência de pontos no sentido horário (Song, 2000), conforme esquema mostrado na **Figura 3.3**. Na **Figura 3.1b** e **Figura 3.2b** os vértices estão ordenados no sentido horário.

O objetivo de se obter uma seqüência de pontos (vértices) no sentido horário é reduzir o número de combinações possíveis para a identificação do objeto. Para se obter a seqüência de vértices de um objeto em uma imagem, sugere-se usar uma imagem binarizada. Inicia-se o algoritmo em um ponto p pertencente ao contorno da imagem, e desloca-se a janela de

busca no sentido horário, ao longo de todo o contorno do objeto, criando uma lista de vértices, à medida que os mesmos são encontrados.

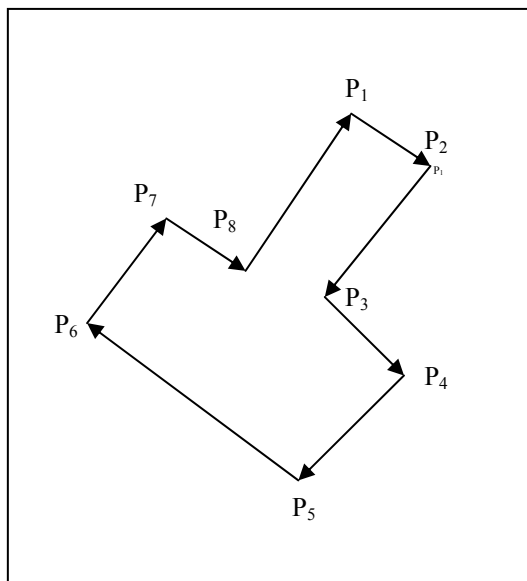


Figura 3.3 – Esquema de ordenação de vértices no sentido horário.

3.3 Criação do Banco de Invariantes

Assumindo que o modelo é composto de M vértices (Song, 2000), definem-se dois vetores de invariantes g_1 e g_2 , equação (3.3), que identificam o modelo:

$$g_i = \begin{bmatrix} I_i(1,2,3,4,5) \\ I_i(2,3,4,5,6) \\ I_i(3,4,5,6,7) \\ I_i(M-1,M,1,2,3) \\ I_i(M,1,2,3,4) \end{bmatrix} \quad i = 1,2 \quad (3.3)$$

, onde I_i são invariantes dos vértices calculados utilizando a equação 3.2.

Utilizando-se do conjunto de pontos da Figura 3.3, onde $M=8$, os vetores g_i são definidos pela equação (3.4):

$$g_i = \begin{bmatrix} I_i(P_1, P_2, P_3, P_4, P_5) \\ I_i(P_2, P_3, P_4, P_5, P_6) \\ I_i(P_3, P_4, P_5, P_6, P_7) \\ I_i(P_4, P_5, P_6, P_7, P_8) \\ I_i(P_5, P_6, P_7, P_8, P_1) \\ I_i(P_6, P_7, P_8, P_1, P_2) \\ I_i(P_7, P_8, P_1, P_2, P_3) \\ I_i(P_8, P_1, P_2, P_3, P_4) \end{bmatrix} \quad i = 1, 2 \quad (3.4)$$

3.4 Identificação de Peças por Comparação

A técnica de identificação de objetos por invariantes desenvolvida neste Projeto considera que um modelo é constituído de um conjunto de pontos característicos (vértices), e através desse conjunto pode-se calcular o vetor de invariantes (equação 3.3). Essa técnica inicia com um processamento off-line para a geração de um banco de objetos modelos associados a vetores invariantes. Na fase de identificação, onde o processamento é on-line, o vetor de invariantes do objeto a ser identificado também é gerado por um conjunto de pontos característicos. Esse vetor de invariantes será utilizado na comparação com o banco de modelos, a fim de identificar o objeto na imagem.

O objeto modelo será chamado simplesmente de *modelo* e o objeto a ser identificado será chamado simplesmente de *objeto*

3.4.1 Processamento Off-Line

A **Figura 3.4** mostra o diagrama de blocos da técnica de identificação, na fase do processamento off-line.

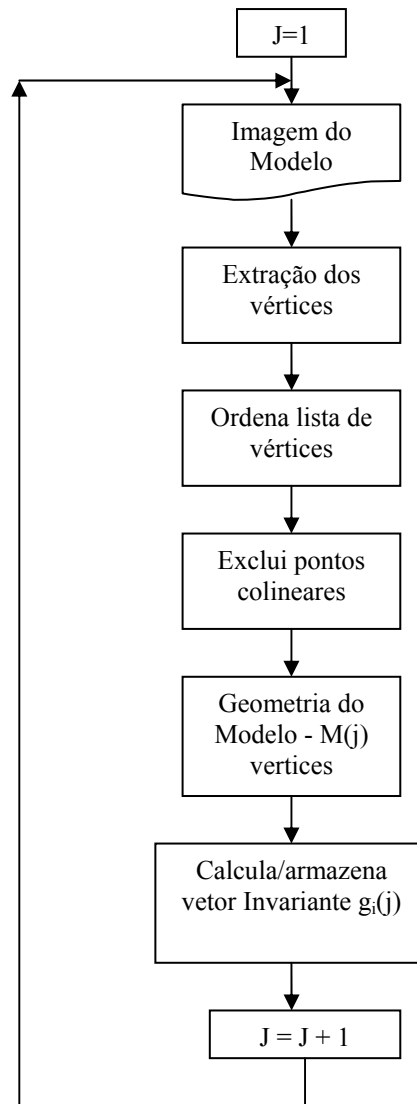


Figura 3.4 – Fluxograma da técnica de reconhecimento de objetos por meio do cálculo de invariantes.
Processamento Off-Line.

O processo é iniciado adquirindo uma imagem de cada objeto modelo (J). Em seguida executa-se a extração dos vértices, citado no item 2.4.2, utilizando a técnica de extração por Trucco. O resultado é uma lista de pontos em ordem crescente do número das linhas. A **Figura 3.5** apresenta o resultado da extração de vértices, sendo (a) e (b) os modelos 1 e 2, respectivamente. A **Tabela 3.1** apresenta a lista dos pontos ordenados pelos números das linhas para modelos 1 e a **Tabela 3.2** apresenta a lista dos pontos ordenados pelos números das linhas para modelos 2. Essa lista será re-ordenada com o objetivo de se obter um polígono fechado (sentido horário).



(a)



(b)

Figura 3.5 – (a) Imagem do modelo 1. (b) Imagem do modelo 2.

Tabela 3.1 – Lista de pontos (vértices) do modelo 1.

i	J
142	128
159	115
209	260
229	232
249	217
277	201
303	369
366	311

Tabela 3.2 - Lista de pontos (vértices) do modelo 2.

i	J
49	313
61	266
67	341
70	365
98	312
131	252
131	355
169	293
328	99
368	146

Em cada seqüência de 5 pontos, deve-se ter o cuidado de não conter 3 pontos colineares. Conhecer a geometria do objeto modelo facilita a construção do polígono fechado que represente sua melhor configuração, sendo que em algumas vezes é importante eliminar vértices desnecessários.

A **Figura 3.6** e a **Figura 3.7** mostram a geometria dos modelos 1 e 2, respectivamente, após a ordenação dos pontos no sentido horário. A inversão da figura ocorre porque, para a câmara (pixels), a ordem crescente do eixo y é de cima para baixo.

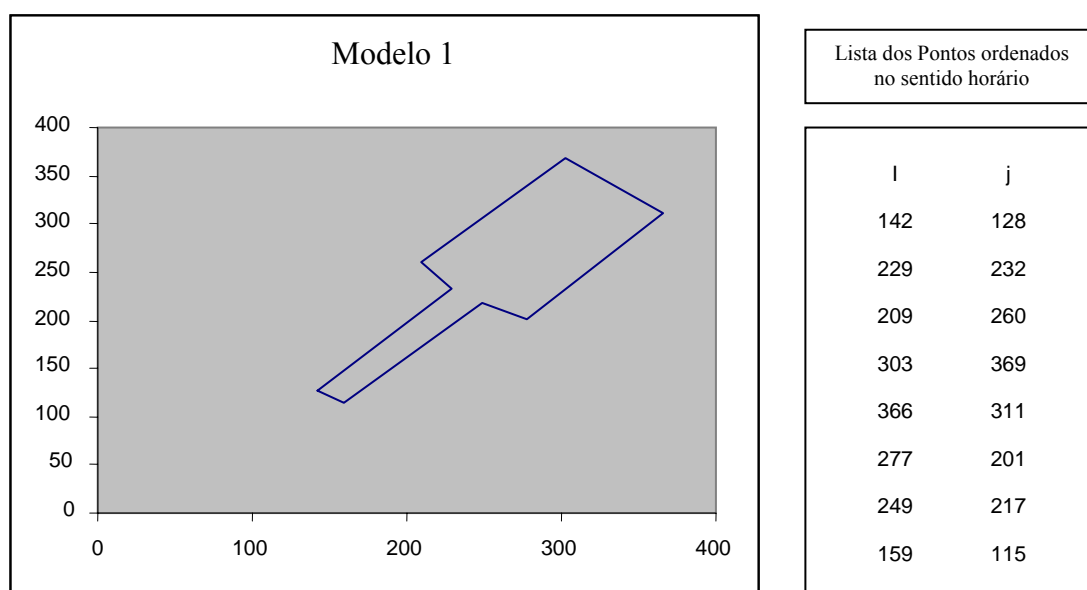


Figura 3.6 – Geometria do modelo 1 e lista dos pontos no sentido horário.

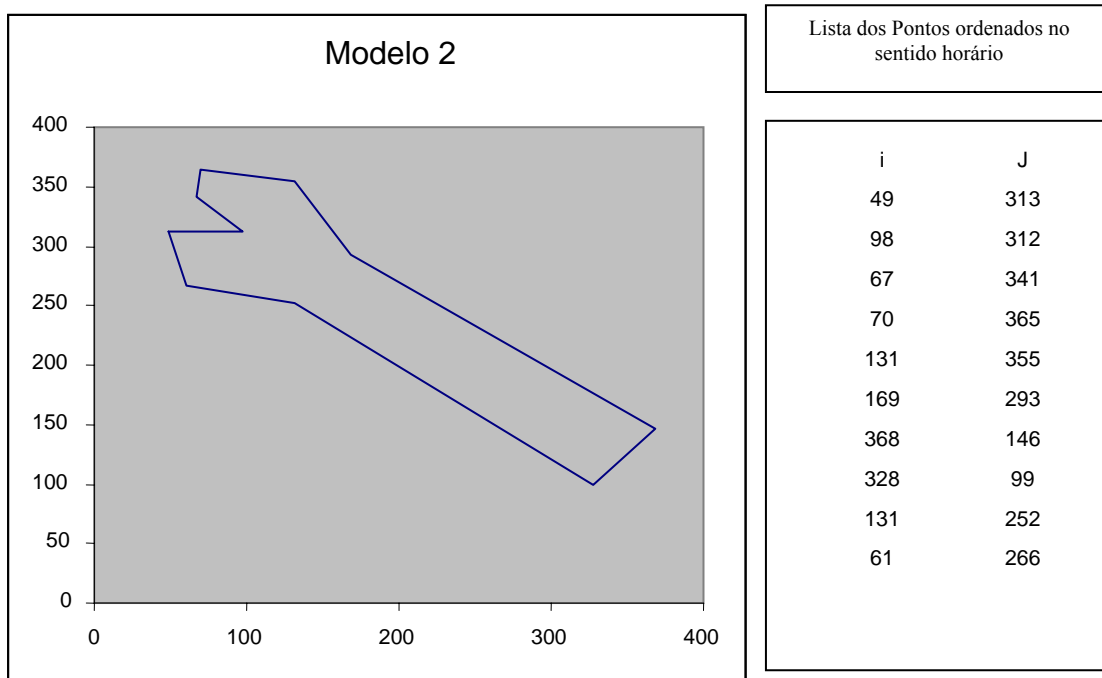


Figura 3.7 – Geometria do modelo 2 e lista dos pontos no sentido horário.

Finalmente, calcula-se o vetor de invariante (equação 3.3), para gerar a tabela dos vetores que representam os modelos (**Tabela 3.3**).

Tabela 3.3 – Valores dos vetores invariantes dos modelos 1 e 2.

Vetor Invariante Modelo 1 (g)		Vetor Invariante Modelo 2 (g)	
I ₁	I ₂	I ₁	I ₂
-0.5151	-0.6445	-0.1882	3.3081
0.1948	7.8687	0.5867	2.7027
0.9390	1.6208	0.8121	1.0127
0.5579	0.6170	0.0381	0.2800
15.3477	0.6240	-1.2576	-0.5731
-0.0092	-49.0756	0.8507	0.8088
0.9727	0.4582	-0.8020	0.0590
9.1682	0.4706	-7.3112	-0.0935
		1.2222	0.3616
		1.0483	1.3560

3.4.2 Processamento On-Line

A **Figura 3.8** mostra o diagrama de blocos da técnica de identificação, na fase do processamento on-line.

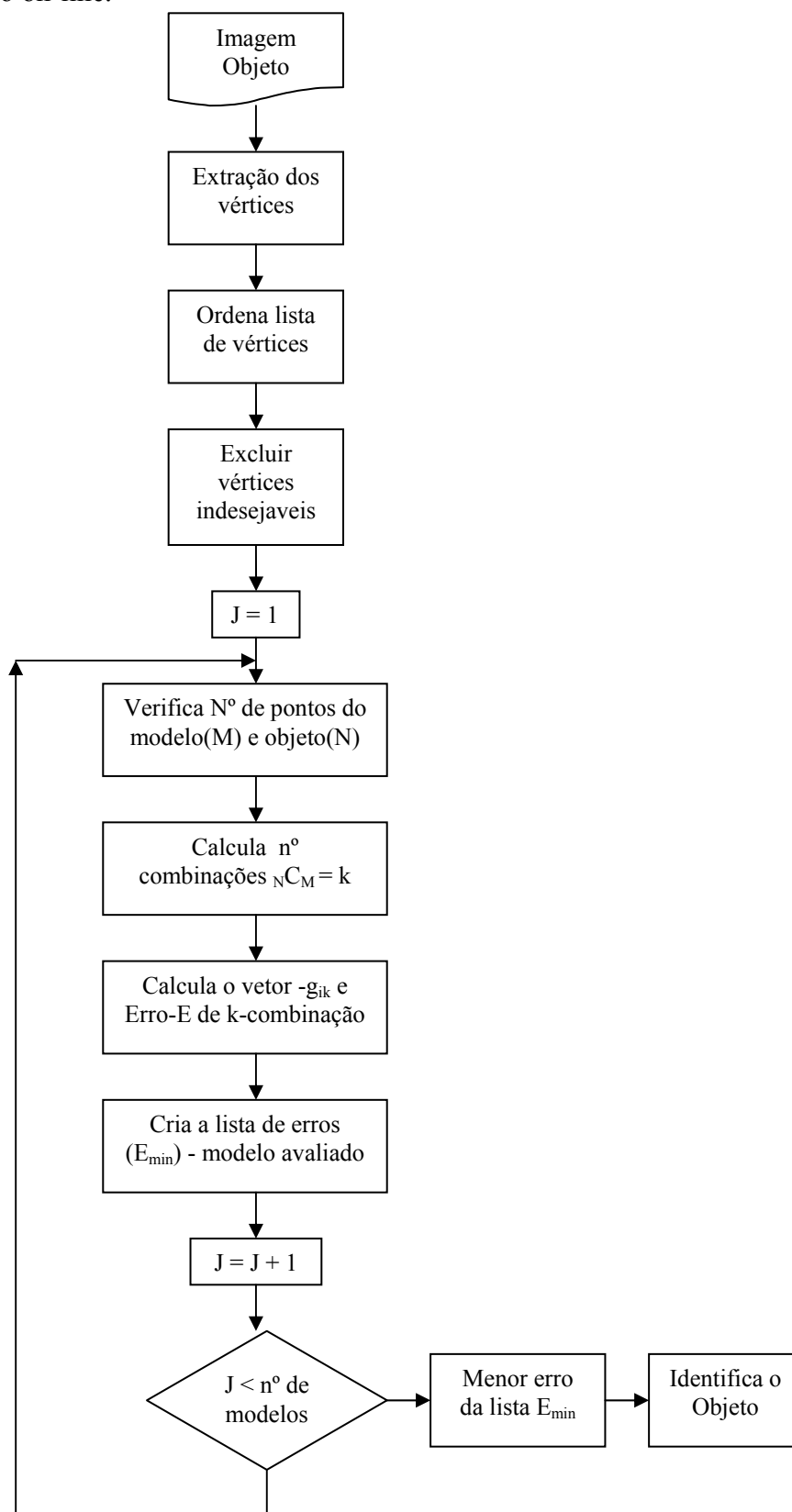


Figura 3.8 – Fluxograma da técnica de reconhecimento de objetos. Processamento On-Line.

Inicia-se o processamento on-line com uma imagem em que se deseja identificar algum objeto, e é semelhante ao processamento off-line, inclusive o processo de exclusão dos pontos indesejáveis. São pontos indesejáveis os pontos colineares. Outra forma de excluir pontos indesejáveis é através das distâncias máxima e mínima entre os pontos que compõem a configuração geométrica dos modelos.

Em seguida, deve-se verificar se o número de vértices que gerou o polígono fechado do objeto é maior que o número de vértices dos modelos. Conhecendo o número de vértices do modelo e do objeto que se deseja identificar, é possível calcular o número de combinações possíveis (k). Para cada combinação calcula-se o vetor de invariantes, equação (3.3), e o erro associado em relação ao modelo, equação (3.5).

$$E = \sum_{k=1}^M \left(\left| \frac{g_{1k} - \hat{g}_{1k}}{g_{1k}} \right| + \left| \frac{g_{2k} - \hat{g}_{2k}}{g_{2k}} \right| \right) \quad (3.5)$$

, onde g_{ik} é o k -ésimo elemento do vetor invariante g_i do modelo, e \hat{g}_{ik} é o invariante correspondente a g_{ik} do objeto.

Gera-se, assim, um vetor de erros (tamanho k), constituído de erros para cada combinação. O menor valor é a combinação que melhor identifica o objeto.

Para cada modelo é possível gerar uma configuração do objeto associado ao erro mínimo, gerando uma lista de erros mínimos. O menor dos erros mínimos identifica finalmente o objeto.

4. RECONHECIMENTO DE OBJETOS – LOCALIZAÇÃO

4.1 Conceitos Básicos

Como já explanado anteriormente, o reconhecimento de um objeto envolve duas operações básicas: *Identificação e localização*. Este capítulo é destinado às técnicas de localização de objetos. O problema de localização é conhecido em visão computacional como *localização de objetos baseados em modelos* ou *estimar sua de posição*.

A localização de objetos significa determinar a posição 3-D e a sua orientação. Para isso é necessário que se tenha uma imagem, um sensor e a geometria do objeto que se deseja localizar. No espaço 3-D a posição está vinculada à translação – **T**, e a orientação à rotação – **R**.

A localização de objetos desenvolvida neste trabalho, baseia-se na utilização de apenas uma imagem, ou seja, não é necessário calibrar o sensor. Portanto, é importante que seus parâmetros intrínsecos sejam conhecidos. Também é necessário que os modelos sejam centrados em suas características geométricas. A condição mais importante é que o objeto já tenha sido identificado. Como os objetos estão vinculados às suas características geométricas, neste caso devem-se excluir as técnicas de identificação de objetos baseada em aparências e outras técnicas que não façam uso das características geométricas do objeto.

Podem-se utilizar imagens de intensidade ou imagens de profundidade. No caso de imagens de intensidade, os modelos estão apresentados em espaço 3-D e as imagens dos objetos em 2-D, necessitando, assim, de uma transformação adequada para a sua devida identificação e localização. Para as imagens de profundidade, tanto os modelos quanto os objetos sob análise estão definidos em espaço 3-D, e nesse caso, podem ser comparados diretamente. Neste trabalho, aplicam-se apenas os casos de utilização de imagens de intensidade.

4.2 Localização de Objetos – Imagem de Intensidade

A localização de objetos a partir de uma imagem de intensidade, dado um modelo, consiste em que todos os pontos e vetores extraídos da imagem estão definidos em função do sistema de referência da câmera e que os parâmetros intrínsecos são todos conhecidos.

O problema consiste em solucionar a equação (4.1).

$$[X_i, Y_i, Z_i]^T = \bar{R} \cdot P_i^m + \bar{T} \quad (4.1)$$

, onde $P_i = [X_i, Y_i, Z_i]^T$ são n pontos ($n \geq 3$), expressos no sistema de referência da câmera, e $P_i^m = [X_i^m, Y_i^m, Z_i^m]^T$ são os n pontos correspondentes do objeto, expressos no sistema de referência do objeto. Como $p_i = [\bar{x}_i, \bar{y}_i]^T$ são as projeções de P_i no sistema da câmera (pixels), então, torna-se necessário conhecer o vetor de translação \bar{T} e a matriz de rotação \bar{R} , para efetuar as transformações rígidas.

Enfim, solucionar a equação (4.1) consiste em determinar os parâmetros extrínsecos da câmera, que pode está conectada ao atuador de um robot.

Dois métodos são conhecidos: modelo da câmera de perspectiva completa e modelo de câmera de perspectiva fraca.

4.2.1 Modelo de Câmera de Perspectiva Completa

Este modelo aplica o método iterativo de Newton (Trucco, *et al* 1998). O método consiste em relacionar pontos do modelo e do objeto, ambos em coordenadas da câmera – 3-D, tomadas através das projeções perspectivas, conforme a equação (4.2)

$$p_i = [x_i, y_i]^T = \left[f \cdot \frac{X_i}{Z_i}, f \cdot \frac{Y_i}{Z_i} \right]^T \quad (4.2)$$

Aplicando a equação (4.1) na equação (4.2), obtém-se a equação (4.3)

$$\begin{aligned} x_i &= f \cdot \frac{r_{11} \cdot X_i^m + r_{12} \cdot Y_i^m + r_{13} \cdot Z_i^m + T_1}{r_{31} \cdot X_i^m + r_{32} \cdot Y_i^m + r_{33} \cdot Z_i^m + T_3} \\ y_i &= f \cdot \frac{r_{21} \cdot X_i^m + r_{22} \cdot Y_i^m + r_{23} \cdot Z_i^m + T_2}{r_{31} \cdot X_i^m + r_{32} \cdot Y_i^m + r_{33} \cdot Z_i^m + T_3} \end{aligned} \quad (4.3)$$

Como \bar{R} e \bar{T} são desconhecidos, basta aplicar os pontos conhecidos (no mínimo 3) para obter assim seis equações com seis variáveis. As variáveis são $R_x(\alpha)$, $R_y(\beta)$ e $R_z(\gamma)$ – Rotações nos eixos $x \rightarrow \alpha$, $y \rightarrow \beta$ e $z \rightarrow \gamma$, respectivamente; e T_x , T_y e T_z – Translações nos eixos x , y e z , respectivamente.

Para determinar as variáveis que solucionam o sistema de equações não lineares, aplica-se o Método Iterativo Newton (Trucco *et al*, 1998), que consiste em:

1. Dados $n \geq 3$ pontos do modelo 3-D e seus correspondentes pontos no sistema de referencia da câmera 3-D – Estima-se R_0 e T_0 – Valores iniciais para Rotação e Translação, respectivamente;
2. Utilizando $R_0 = \bar{R}$ e $T_0 = \bar{T}$ calcule P_i , que deve ser a localização prevista para P_i^m ;
3. Projete P_i no plano da imagem (π) – pixel, utilizando a equação (4.2);
4. Avalie o residual δ_{xi} e δ_{yi} através das equações (4.4);

$$\delta_{xi} = x_i(R_0, T_0) - \bar{x}_i \quad \text{e} \quad \delta_{yi} = y_i(R_0, T_0) - \bar{y}_i \quad (4.4)$$

5. Resolva o sistema linear – equação (4.5);

$$\sum_{j=1}^3 \left[\frac{\partial x_i}{\partial T_j} \Delta T_j + \frac{\partial x_i}{\partial R_j} \Delta R_j \right] = \delta x_i \quad (4.5)$$

$$\sum_{j=1}^3 \left[\frac{\partial y_i}{\partial T_j} \Delta T_j + \frac{\partial y_i}{\partial R_j} \Delta R_j \right] = \delta y_i$$

Para determinar ΔT_j e ΔR_j , $j = 1, 2, 3$

6. Altere os valores estimados para a rotação R e a translação T ;
7. Se δ_{xi} e δ_{yi} são valores muito pequenos, os novos valores de R e T são a solução, caso contrário voltar ao passo 2 com os novos valores de R e T .

4.2.2 Modelo de Câmera de Perspectiva Fraca

O método aplicado a modelos de câmera de perspectiva fraca é matematicamente mais simples que o método de câmera de perspectiva completa, porém é mais restritivo. Neste caso não se utilizam os parâmetros intrínsecos e extrínsecos da câmera. Como não é possível definir quão longe está o ponto (objeto), este modelo não fornece informações em distância.

O problema de localização de objetos, em visão monocular, torna-se mais complexo quando os parâmetros intrínsecos e extrínsecos são desconhecidos. Uma técnica proposta de localização de objetos utiliza-se do conceito de transformações perspectivas que preservam razões cruzadas (cross-ratio) (Horaud *et al*, 1989 e Michel, 1852).

O método tem referencia no trabalho do matemático francês Michel Chasle (1852), que estudou transformações perspectivas que preservam razões cruzadas (cross-ratio) de quatro (4) pontos colineares, e por extensão definiu uma relação entre coordenadas 2-D (x,y) de um ponto no plano do objeto e sua projeção perspectiva (x',y') no plano da imagem. Horaud, *et al* (1989) mostra que essa relação pode ser escrita pela equação (4.6).

$$\begin{aligned}x' &= \lambda_1 \frac{ax + by + l}{px + qy + l} \\y' &= \lambda_2 \frac{cx + dy + l}{px + qy + l}\end{aligned}\tag{4.6}$$

, onde $a, b, c, d, p, q, \lambda_1$ e λ_2 são constantes.

Sejam as equações (4.7)

$$x' = \frac{u}{w}, y' = \frac{v}{w}, a' = \lambda_1 \cdot a, b' = \lambda_1 \cdot b, c' = \lambda_2 \cdot c, d' = \lambda_2 \cdot d\tag{4.7}$$

Aplicando as equações (4.7) em (4.6), pode-se reescrevê-las, gerando assim a equação (4.8).

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} a' & b' & \lambda_1 \\ c' & d' & \lambda_2 \\ p & q & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ l \end{bmatrix}\tag{4.8}$$

Na prática, deseja-se determinar (x', y') no plano do objeto, a partir das coordenadas da câmera (pixel). Enfim, devem-se determinar os coeficientes $a', b', c', d', p, q, \lambda_1$ e λ_2 da matriz.

As equações de Horaud *et al.*, (1989), representam um sistema de 2 equações com 8 incógnitas, então, necessitam-se de no mínimo 4 pontos não colineares marcados no plano da imagem (pixel – (x', y')) e seus correspondentes no plano físico do objeto (coordenadas homogêneas – $(x, y, 1)$), para resolver o sistema. Esse sistema matricial resulta na equação $A.x = b$ e pode ser escrita pela equação (4.9).

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x'_4 & -y_4x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y'_4 & -y_4y'_4 \end{bmatrix} \begin{bmatrix} a' \\ b' \\ \lambda_1 \\ c' \\ d' \\ \lambda_2 \\ p \\ q \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{bmatrix} \quad (4.9)$$

Pode-se melhorar a robustez do método, admitindo-se mais alguns pontos, com o objetivo de mensurar os erros encontrados entre os valores previstos e os valores reais.

O sistema de equações (4.9) é resolvido, utilizando-se do método dos mínimos quadrados do sistema $A^T . A . x = A^T . b$.

Finalmente, deseja-se localizar um ponto M de coordenadas desconhecidas $(x_m, y_m, 1)$, no plano do objeto a partir de coordenadas da imagem (pixel).

4.2.3 Exemplo de Aplicação para Modelo de Câmera de Perspectiva Fraca

A **Figura 4.1** apresenta as fases do resultado de localização de objetos. Na **Figura 4.1a** o objeto está sobre uma grade com marcas que definem suas posições no plano. A **Figura 4.1b** mostra o histograma da imagem (**Figura 4.1a**), utilizado na identificação do limiar (220)

necessário à binarização da imagem. A **Figura 4.1c** apresenta a imagem binarizada, utilizada para extração dos vértices. Finalmente a **Figura 4.1d** identifica as posições da imagem no sistema da câmera (pixel).

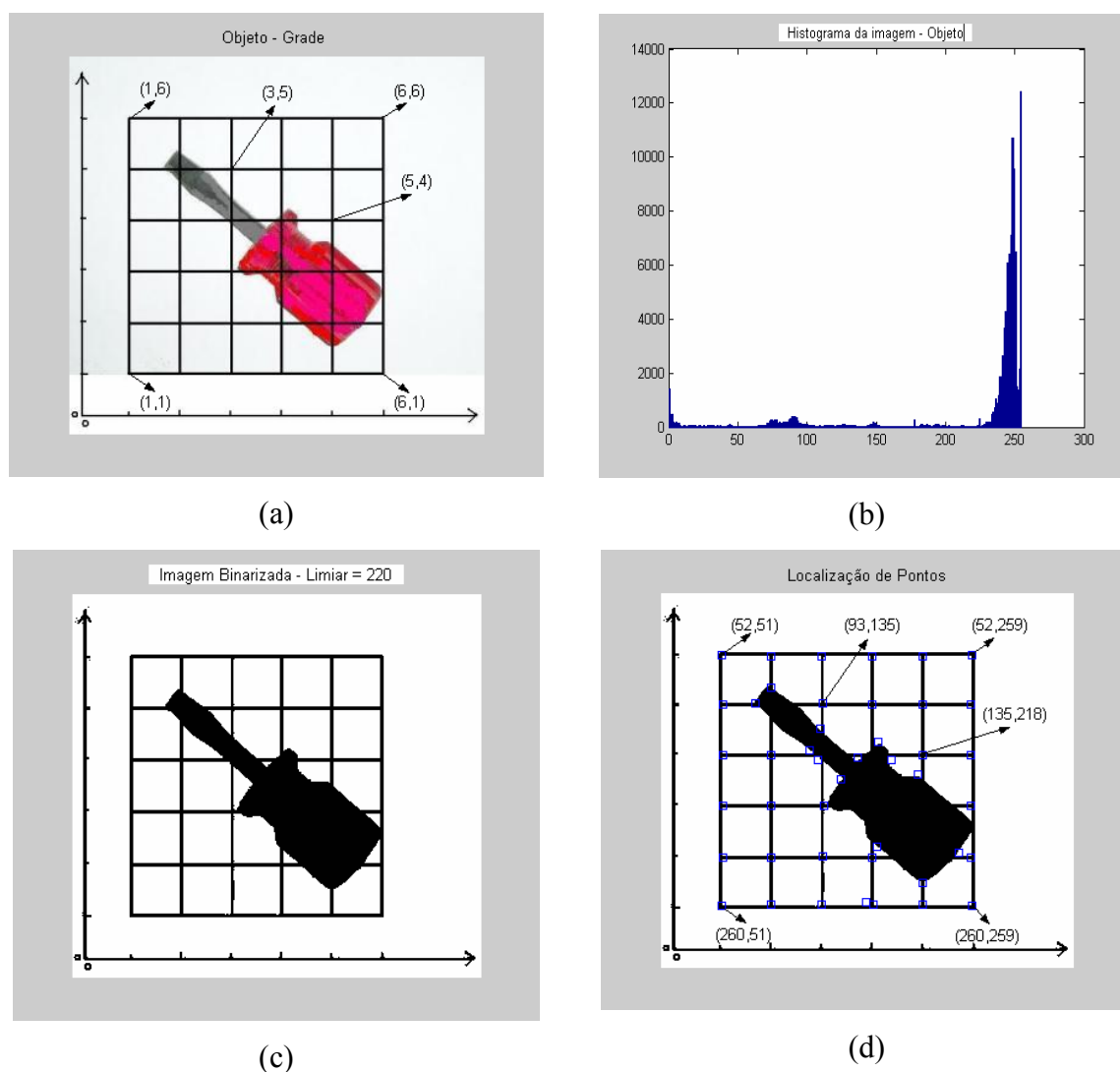


Figura 4.1 – Fases para a localização do objeto. (a) Imagem do posicionamento do objeto sobre uma grade. (b) Histograma da imagem do objeto. (c) Imagem binarizada. (d) Posições do objeto e marcas - pixels.

Utilizando-se os pontos (1,1), (1,6), (6,1) e (6,6) no plano de referência do objeto (**Figura 4.1a**) e seus respectivos pontos (260,51), (52,51), (260,259) e (52,259) no plano de referência da imagem em pixel (**Figura 4.1c**), na equação (4.9), é possível determinar os coeficientes a' , b' , c' , d' , p , q , λ_1 e λ_2 . Enfim, através desses coeficientes e da equação (4.8) é possível determinar um ponto do plano de referência da imagem fazendo uso de um ponto no plano de referência do objeto e vice-versa.

Assim, utilizando-se os pontos, no plano de referência da imagem, $A = (93,135)$ e $B = (135,218)$ é possível calcular os respectivos pontos $A' = (3,0192 ; 5,0144)$ e $B' = (5,0144 ; 4,0048)$, no plano de referência do objeto, com uma taxa de erro muito pequena, sem conhecer os parâmetros intrínsecos e extrínsecos da câmera.

5. CONTROLE DO ESPAÇO OPERACIONAL UTILIZANDO VISÃO

5.1 Aspectos Gerais

Em termos gerais, o controle do espaço operacional resulta da integração de dois subsistemas:

- Subsistema de controle do manipulador
- Subsistema de visão

O subsistema de controle do manipulador tem um comportamento dinâmico e complexo. No presente estudo admite-se que o controle dinâmico dos movimentos do manipulador fique a cargo do controlador, cujos movimentos não perpassam pelos requisitos de evolução temporal restritivos, admitindo assim que a operação decorre em modo estacionário. Então, proceder-se-á apenas ao estudo da cinemática e do jacobiano do manipulador.

O subsistema de visão deverá ter capacidade para fazer o acompanhamento dinâmico do manipulador, e ser capaz de computar, em tempo real, os parâmetros que caracterizam seu movimento. Neste trabalho, o reconhecimento de objetos (identificação e localização), principalmente identificação, é realizado por técnicas de geometria invariante.

5.2 Subsistema de Controle do Manipulador

5.2.1 Espaço Operacional e Espaço Junta – Cinemática Direta e Inversa

Pode-se definir *espaço operacional* como sendo o local onde a tarefa está sendo executada, ou seja, a posição da garra. O *espaço junta* está relacionado com as variáveis de junta.

Um manipulador consiste basicamente de uma série de corpos rígidos unidos entre si por articulações (juntas). As juntas podem ser prismáticas ou rotativas. O ligamento da base é usualmente fixo em relação ao ambiente externo e o último elemento é também chamado de efetuador, garra ou ferramenta, doravante será designado como garra.

Representa-se a posição e a orientação da garra no sistema de coordenadas $O_n-x_ny_nz_n$, descrito em relação ao sistema $O_0-x_0y_0z_0$, fixo na base (primeiro elo). Define-se para cada elo um sistema de coordenadas $O_i-x_iy_iz_i$. É possível determinar a posição e orientação do sistema i em relação ao sistema anterior $i-1$ pelo uso de matrizes homogêneas relacionando as transformações entre esses sistemas. Assim, a posição e a orientação da garra em relação à base são obtidas por uma composição de transformações homogêneas consecutivas, partindo-se do sistema da base para o último sistema (garra).

A notação de *Denavit-Hartenberg* é utilizada para descrever de forma sistêmica a posição e a orientação entre dois elos de um manipulador, através de coordenadas homogêneas. Logo, pode-se obter a posição e a orientação da garra em relação ao sistema da base em função dos deslocamentos de todas as articulações. O deslocamento de cada articulação é dado por d_i para articulação prismática e θ_i para articulação rotativa, aqui, simplesmente denotada por q_i . Assim, a matriz homogênea $A_{i-1}^i(q_i)$ descreve a posição e a orientação em função de q_i . A equação (5.1) descreve a posição e a orientação da garra em relação à base (Spong, 1989).

$$A_0^n(q_1, q_2, \dots, q_n) = A_0^1(q_1) \cdot A_1^2(q_2) \cdot \dots \cdot A_{n-1}^n(q_n) \quad (5.1)$$

A equação (5.1) pode ser escrita na forma da equação (5.2)

$$A_0^n(q_1, q_2, \dots, q_n) = \begin{bmatrix} R_0^n(q_1, q_2, \dots, q_n) & X_0^n(q_1, q_2, \dots, q_n) \\ 0 & I \end{bmatrix} \quad (5.2)$$

, onde $R_0^n(q_1, q_2, \dots, q_n)$ é a matriz de rotação que representa a orientação da garra em relação ao sistema da base e $X_0^n(q_1, q_2, \dots, q_n)$ é o vetor que fornece a posição da garra em relação ao sistema da base.

Neste caso, *espaço operacional* está relacionado à posição $X_0^n(q_1, q_2, \dots, q_n)$ da garra. O *espaço junta* está relacionado com as variáveis $R_0^n(q_1, q_2, \dots, q_n)$ da junta.

As equações da cinemática direta (Murray, *et al* 1994) permitem calcular a posição da garra em função das variáveis de junta. A cinemática inversa consiste em determinar as variáveis

da junta para uma determinada posição da garra, que nem sempre existe, ou é uma solução única.

A **Figura 5.1** apresenta os diversos referenciais do espaço de trabalho; (B) referencial da base do manipulador, (G) referencial da garra e (C) referencial da câmera.

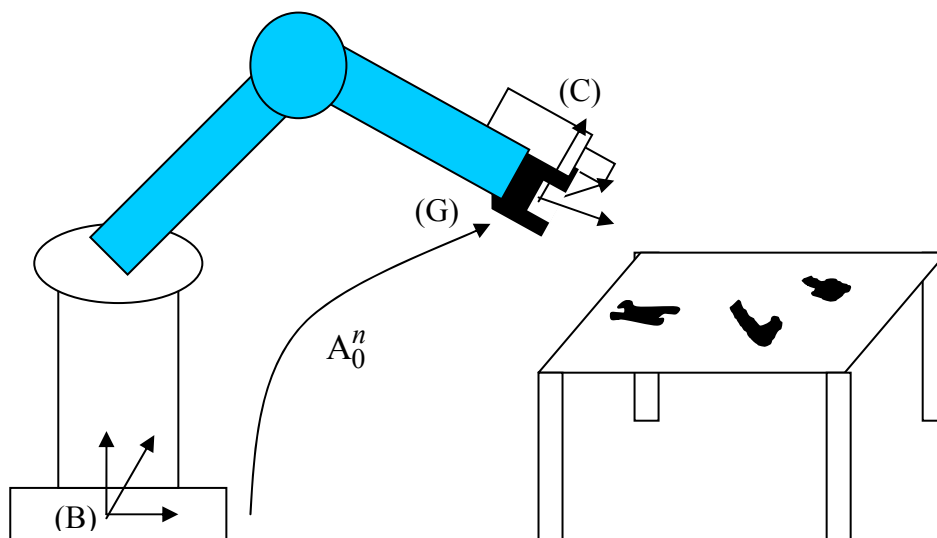


Figura 5.1 – Sistemas de coordenadas. (B) Base do manipulador, (G) garra e (C) Câmera.

5.2.2 Matriz Jacobiano do Manipulador

Nos sistemas de controle de movimento é necessário conhecer como a variação infinitesimal dos elementos de um manipulador afeta o movimento infinitesimal da garra, sendo que a vantagem de estabelecer pequenas variações é obter uma relação linear.

Podem-se relacionar as velocidades linear e angular da garra em função das velocidades das articulações em uma mesma equação (Crane, 1998), resultando na equação (5.3):

$$V_n = \begin{bmatrix} v_n \\ \omega_n \end{bmatrix} = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix} \cdot \dot{q} = J_M \cdot \dot{q} \quad (5.3)$$

A matriz J_M é definida como sendo a Matriz do Jacobiano da garra. Esta matriz relaciona as velocidades linear e angular do efetuador, expressas no sistema de coordenadas da base, com as velocidades das articulações, para uma dada configuração do manipulador.

Resumindo, cada coluna da matriz do jacobiano relaciona-se com cada articulação do manipulador e pode ser escrita pela equação (5.4) ou (5.5), dependendo do tipo da articulação.

$$J_{M i} = \begin{bmatrix} Z_{i-1} \times r_{i-1,n} \\ Z_{i-1} \end{bmatrix}, \text{ se a articulação } i \text{ for de revolução} \quad (5.4)$$

$$J_{M i} = \begin{bmatrix} Z_{i-1} \\ 0 \end{bmatrix}, \text{ se a articulação } i \text{ for de translação} \quad (5.5)$$

Sendo $P_0 = [0 \ 0 \ 0 \ 1]^T$ e $Z_0 = [0 \ 0 \ 1]^T$, $r_{i-1,n}$ é o vetor que une a origem do sistema de coordenadas da articulação i (ponto 0_{i-1}) à origem do sistema de coordenadas da garra (ponto 0_n) e é obtido pela terceira coluna da matriz de translação, e Z_{i-1} é dado pela terceira coluna da matriz de rotação.

A dimensão da Matriz do Jacobiano é $m \times n$, onde m é o número de linhas e corresponde ao número de graus de liberdade do campo de trabalho do manipulador e n é o número de colunas e corresponde ao número de articulações. O valor de m será no máximo igual a 6 se o manipulador trabalhar no espaço (3 graus de liberdade de posicionamento e 3 para orientação), ou será igual a 3 se trabalhar no plano (2 graus de liberdade de posicionamento e 1 para orientação).

5.3 Subsistema de Visão

O sistema de visão deve ser capaz de extrair informações da imagem necessárias à identificação e localização de objetos.

Para controlar um manipulador, utilizando-se de informações proveniente de um sistema de visão, é necessário conhecer os aspectos geométricos do processo de formação da imagem. A imagem produzida por uma câmera é uma projeção 2-D de uma cena. Conhecendo a relação geométrica entre os pontos extraídos da imagem e o do objeto real, é possível obter a informação 3-D.

5.3.1 Formação da Imagem

Para a formação geométrica da imagem pode-se utilizar o modelo de perspectiva ou o modelo de perspectiva fraca, já apresentados em capítulo anterior.

É possível conhecer as coordenadas de um ponto P no sistema de referência do mundo (o sistema de coordenadas da base do manipulador), se forem conhecidas as coordenadas no sistema de referência da câmera e vice-versa. A **Figura 5.2** apresenta os diferentes sistemas de referência; (B) sistema de referência da base do manipulador, (Obj) sistema de referência do objeto e (C) sistema de referência da câmera.

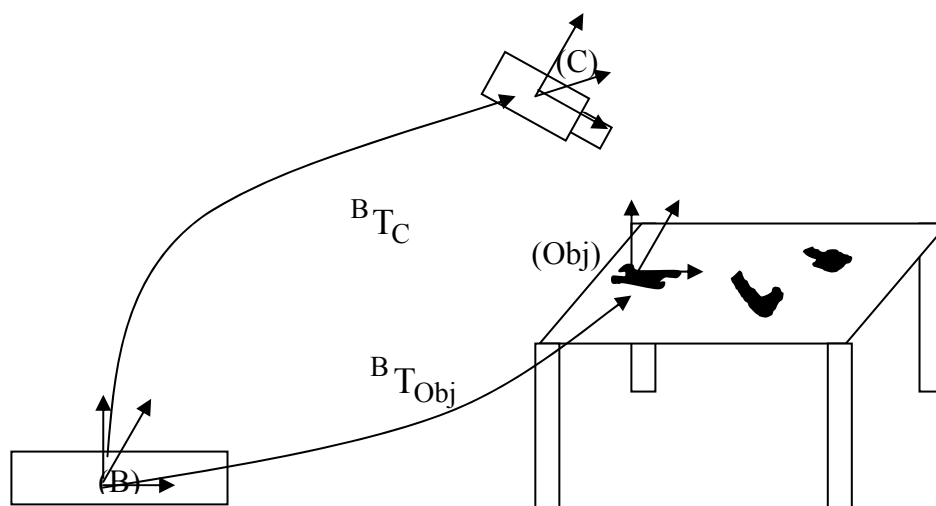


Figura 5.2 – Sistemas de coordenadas. (B) Base do manipulador, (Obj) objeto e (C) Câmera.

Supondo que o modelo de formação da imagem é a projeção perspectiva, o vetor de parâmetros característicos da imagem é dado por $\mathbf{c} = [u \ v]^T$, onde u e v são dados pela equação (5.6).

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} fK_x & 0 & C_x & 0 \\ 0 & fK_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot ({}^C T_B) \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (5.6)$$

, onde $\mathbf{P} = [X \ Y \ Z]^T$ é expresso em coordenadas do mundo (B), origem do referencial do manipulador, ${}^C T_B$ é a matriz de transformação do sistema de coordenadas da base do manipulador (B) para o sistema de coordenadas da câmara (C), (K_x, K_y) é a dimensão do pixel/metro da imagem e (C_x, C_y) é a posição do centro da imagem, ou qualquer outro ponto característico da imagem (vértices, por exemplo).

5.3.2 Jacobiano da imagem

Nas aplicações que se utiliza de um sistema de visão para reconhecimento de objetos, é necessário relacionar os movimentos diferenciais dos parâmetros característicos da imagem com os movimentos diferenciais do objeto a seguir visualmente. O Jacobiano da imagem – J_{img} encarrega-se de estabelecer a relação entre o plano da imagem e o equivalente no espaço operacional.

Supondo que o elemento terminal move-se no espaço de trabalho, e que esse movimento será descrito, no sistema de referencial da câmara, por uma velocidade angular $\Omega(t) = [\omega_x \ \omega_y \ \omega_z]^T$ e uma velocidade de translação $T(t) = [T_x \ T_y \ T_z]^T$, então, a velocidade de um ponto $\mathbf{P} = [X \ Y \ Z]^T$ no elemento terminal, expresso no sistema de referencia da câmara pode ser escrito por $\dot{\mathbf{P}} = \mathbf{\Omega} \mathbf{xP} + \mathbf{T}$ (Corke, 1996) ou pelas equações (5.7).

$$\begin{aligned}\dot{X} &= Z\omega_y - Y\omega_z + T_x \\ \dot{Y} &= X\omega_z - Z\omega_x + T_y \\ \dot{Z} &= Y\omega_x - X\omega_y + T_z\end{aligned}\tag{5.7}$$

Aplicando as equações de projeção perspectiva em (5.7) obtem-se as equações (5.8).

$$\begin{aligned}\dot{X} &= Z\omega_y - \frac{vZ}{f}\omega_z + T_x \\ \dot{Y} &= \frac{uZ}{f}\omega_z - Z\omega_x + T_y \\ \dot{Z} &= \frac{vZ}{f}\omega_x - \frac{uZ}{f}\omega_y + T_z\end{aligned}\tag{5.8}$$

Sendo \mathbf{P} um ponto da garra e $\dot{\mathbf{P}}$ a sua velocidade instantânea e se \mathbf{P} se projeta em $\mathbf{c} = [u, v]^T$, então será induzida na imagem uma velocidade $\dot{\mathbf{c}}$ determinada pela equação (5.9).

$$\dot{\mathbf{c}} = \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \mathbf{J}_{img} \begin{bmatrix} T_X \\ T_Y \\ T_Z \\ \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix} \quad (5.9)$$

Considerando $f_x = f K_x$, $f_y = f K_y$, $u_0 = \frac{u - C_x}{f_x}$ e $v_0 = \frac{v - C_y}{f_y}$, é possível escrever o jacobiano da imagem pela equação matricial (5.10).

$$\mathbf{J}_{img} = \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{u_0}{Z} & -\frac{u_0 \cdot v_0}{f_y} & \frac{f_x^2 + u_0^2}{f_x} & -v_0 \cdot \frac{f_x}{f_y} \\ 0 & \frac{f_y}{Z} & -\frac{v_0}{Z} & -\frac{f_y^2 + v_0^2}{f_y} & \frac{u_0 \cdot v_0}{f_x} & u_0 \cdot \frac{f_y}{f_x} \end{bmatrix} \quad (5.10)$$

5.3.3 Jacobiano Completo

Relacionam-se coordenadas de junta com os parâmetros característicos da imagem, multiplicando o jacobiano da imagem pelo jacobiano do manipulador, gerando assim o jacobiano completo. A relação entre os jacobianos é apresentada na **Figura 5.3** (Almeida, 2004).

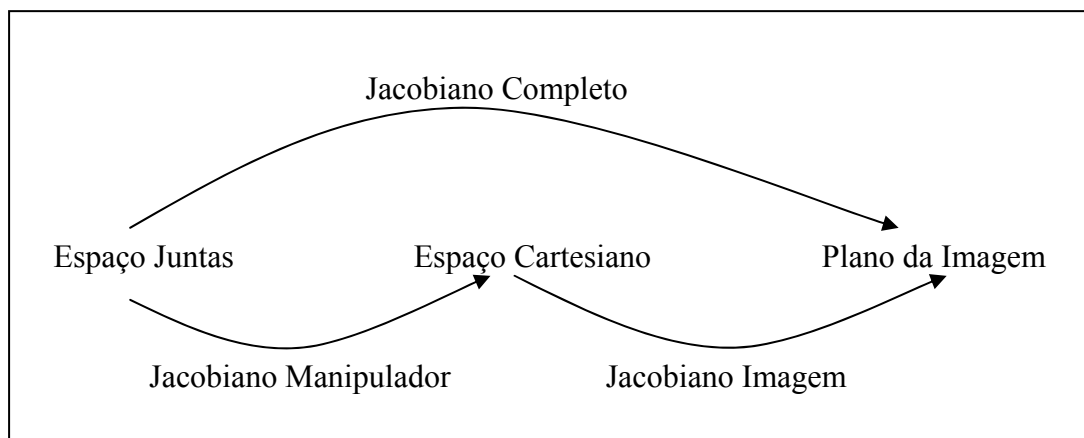


Figura 5.3 – Relação entre os Jacobianos Manipulador, Imagem e Completo.

Tomando $\mathbf{v}_n = [T_x \ T_y \ T_z]^T$ e $\boldsymbol{\omega}_n = [\omega_x \ \omega_y \ \omega_z]^T$ na equação (5.3) e substituindo na equação (5.9), obtém-se a equação (5.11), onde é possível identificar o Jacobiano Completo.

$$\begin{bmatrix} \dot{\cdot} \\ \mathbf{u} \\ \dot{\cdot} \\ \mathbf{v} \end{bmatrix} = \mathbf{J}_{img} \mathbf{J}_M \cdot \dot{\mathbf{q}} \quad (5.11)$$

Sendo o Jacobiano Completo $\mathbf{J} = \mathbf{J}_{img} \mathbf{J}_M$, é possível obter a cinemática inversa pela equação (5.12).

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \begin{bmatrix} \dot{\cdot} \\ \mathbf{u} \\ \dot{\cdot} \\ \mathbf{v} \end{bmatrix} \quad (5.12)$$

A matriz \mathbf{J}^{-1} só é calculável se a matriz \mathbf{J} for quadrada e não singular. Caso a matriz \mathbf{J} não seja quadrada (nxm) \mathbf{J}^{-1} não existe, e apenas através do cálculo do erro dos mínimos quadráticos é possível determinar sua solução (Almeida, 2004), dada pela equação (5.13).

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \cdot \begin{bmatrix} \dot{\cdot} \\ \mathbf{u} \\ \dot{\cdot} \\ \mathbf{v} \end{bmatrix} = \mathbf{J}^+ \cdot \begin{bmatrix} \dot{\cdot} \\ \mathbf{u} \\ \dot{\cdot} \\ \mathbf{v} \end{bmatrix} + (\mathbf{I} - \mathbf{J}^+ \cdot \mathbf{J}) \cdot \mathbf{b} \quad (5.13)$$

, onde \mathbf{J}^+ é a pseudo-inversa de \mathbf{J} e \mathbf{b} um vetor arbitrário (mx1). A solução que minimiza o

erro quadrático resulta num valor para $\dot{\mathbf{q}}$ que minimiza a norma $\left\| \begin{bmatrix} \dot{\cdot} \\ \mathbf{u} \\ \dot{\cdot} \\ \mathbf{v} \end{bmatrix} - \mathbf{J} \dot{\mathbf{q}} \right\|$.

Com relação aos valores de m e n, a pseudo-inversa pode ser definida com as seguintes peculiaridades:

- $m > n$: define-se a matriz pseudo-inversa à esquerda - $\mathbf{J}_L^+ = (\mathbf{J}^T \cdot \mathbf{J})^{-1} \cdot \mathbf{J}^T$
- $n > m$: define-se a matriz pseudo-inversa à direita - $\mathbf{J}_R^+ = \mathbf{J}^T \cdot (\mathbf{J} \cdot \mathbf{J}^T)^{-1}$.

5.4 Sistema de Controle Por Visão

Os sistemas de controle por visão podem ser divididos em dois métodos:

- Sistema dinâmico – Vê-e-Move (*dynamic look-and-move*)

Utiliza-se o controlador do manipulador para a cadeia de controle interna, e a informação visual para o controle exterior.

- Sistema por acompanhamento visual direto – (*direct visual servo*)

Não utiliza o controlador do manipulador e os valores que devem ser aplicados a cada uma das juntas são calculados a partir, exclusivamente, das imagens e da localização (posição e orientação) do manipulador.

A maioria dos sistemas adota o sistema dinâmico, devido a seus manipuladores serem dotados de uma interface que permite enviar comandos incrementais de posição ou velocidade, facilitando a construção do controle por visão. Esses sistemas de controle por visão podem também ser divididos em dois (Almeida, 2004):

- Controle por posição
- Controle por imagem

A **Figura 5.4** apresenta um esquema de sistemas de controle por posição e a **Figura 5.5** apresenta um esquema de sistemas de controle por imagem.

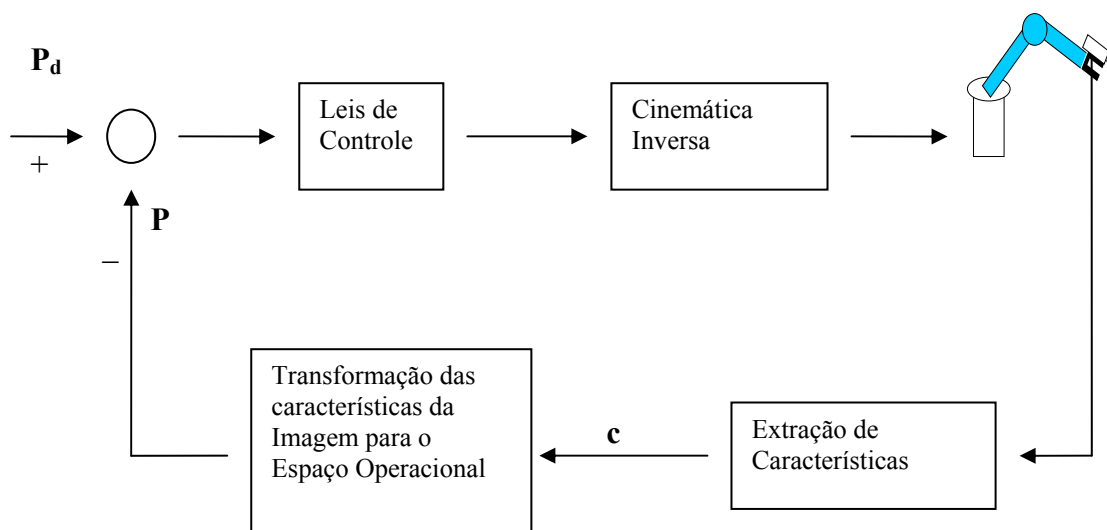


Figura 5.4 – Sistemas de controle por posição (Almeida, 2004).

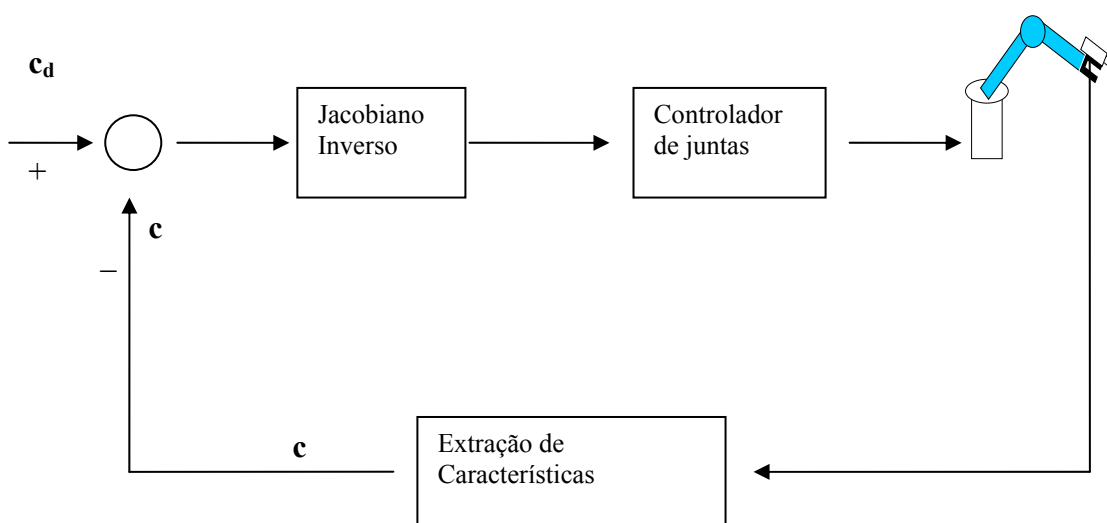


Figura 5.5 – Sistemas de controle por imagem (Almeida, 2004).

No sistema de controle por posição, as características são extraídas da imagem e usadas em conjunto com o modelo geométrico do objeto que será manipulado e o modelo da câmera, sendo possível calcular a distância, no plano da imagem, entre a garra e o objeto. Neste caso, o erro entre a posição corrente \mathbf{P} e a posição desejada \mathbf{P}_d para o manipulador é definido no espaço operacional. Esse sistema é dependente dos parâmetros resultantes da calibração e da cinemática do sistema, ou seja, quanto melhor a qualidade da calibração da câmera e do

manipulador, mais precisos serão os resultados do movimento. Neste sistema é possível descrever as tarefas de posição em coordenadas cartesianas.

O sistema de controle por imagem consiste em controlar a posição e orientação do braço manipulador utilizando informação visual obtida através das características da imagem, \mathbf{c}_d e \mathbf{c} , parâmetros característicos da posição desejada e da posição atual da garra, respectivamente. Este sistema é mais apropriado nos casos em que se desconhece a geometria do modelo. Os movimentos do manipulador modificam a imagem observada no sistema de visão, e essa alteração intervém na cadeia de controle do manipulador. Neste sistema, apesar de ter cálculos mais simples de computação e eliminar a necessidade de calibração da câmera, é não linear e altamente acoplado.

Os sistemas que observam simultaneamente o objeto e o elemento terminal (garra) são denominados de sistemas com elemento terminal (garra) em cadeia fechada – *ECL endpoint closed-loop*, caso contrário, são denominados de sistemas com elemento terminal (garra) em cadeia aberta – *EOL endpoint open-loop*.

5.5 Cinemática do Erro

Em sistemas em que a câmera está instalada no elemento terminal (garra), qualquer movimento do manipulador causa mudanças na imagem, e estas mudanças são observadas no sistema de visão.

Se o sistema é de controle por imagem, por exemplo, o erro é definido em termos da informação visual obtida através das características da imagem. Assim, a posição do objeto a ser manipulado na imagem corresponde ao vetor de parâmetros característicos da imagem $\mathbf{c}_{obj} = [\mathbf{u}_{obj} , \mathbf{v}_{obj}]^T$, equivalente à \mathbf{P}_{obj} , no referencial do mundo (base do manipulador), e a posição do elemento terminal (garra) na imagem corresponde ao vetor de parâmetros característicos da imagem $\mathbf{c}_g = [\mathbf{u}_g , \mathbf{v}_g]^T$, equivalente à \mathbf{P}_g , no referencial do mundo (base do manipulador). O erro será calculado pela diferença desses parâmetros, dada pela equação (5.14).

$$\mathbf{e} = \mathbf{c}_{obj} - \mathbf{c}_g \quad (5.14)$$

O jacobiano completo encarrega-se de estabelecer a relação entre as mudanças observadas no sistema de referência da imagem e as respectivas alterações de posição do manipulador. Então, torna-se necessário calcular o inverso do jacobiano completo.

Em sistemas de controle por imagem, observam-se as variações da velocidade de um ponto na imagem e pretende-se saber quais os diferenciais necessários aos movimentos das juntas, a fim de deslocar o manipulador para o ponto pretendido, utilizando-se do inverso do jacobiano (equação 5.12).

Dado que a função de erro ($\mathbf{e} = \mathbf{c}_{obj} - \mathbf{c}_g$), a lei de controle proporcional será dada pela equação (5.15).

$$\dot{\mathbf{q}} = \mathbf{K}(\mathbf{J}^{-1})\mathbf{e} \quad (5.15)$$

, onde \mathbf{K} é uma constante de ganho com dimensões apropriadas.

6. EXPERIMENTOS

Neste capítulo serão apresentados os resultados dos processamentos necessários ao reconhecimento de objetos – *Identificação*.

6.1 Identificação dos Objetos

Para a identificação dos objetos, foram executadas as tarefas em duas fases: a primeira fase está relacionada ao processamento off-line e a segunda fase ao processamento on-line.

6.1.1 Processamento Off-line – Geração do Banco de Invariantes

Os primeiros experimentos foram executados para gerar um banco de dados de modelos. Esse banco de dados é um conjunto de vetores invariantes que estão associados aos modelos dos objetos que foram previamente selecionados. No capítulo 3 (figura 3.4), o fluxograma para o processamento off-line apresenta a seqüência de tarefas necessárias à geração do vetor de invariantes desses modelos.

a) Objetos

Foram utilizados três modelos de objetos. A escolha se deu pela complexidade da geometria das peças e pela proximidade de objetos industriais comuns. A **Figura 6.1** apresenta as imagens das peças escolhidas. Essas imagens já foram atenuadas (extração de ruídos) com um filtro gaussiano (desvio padrão = 1,5).

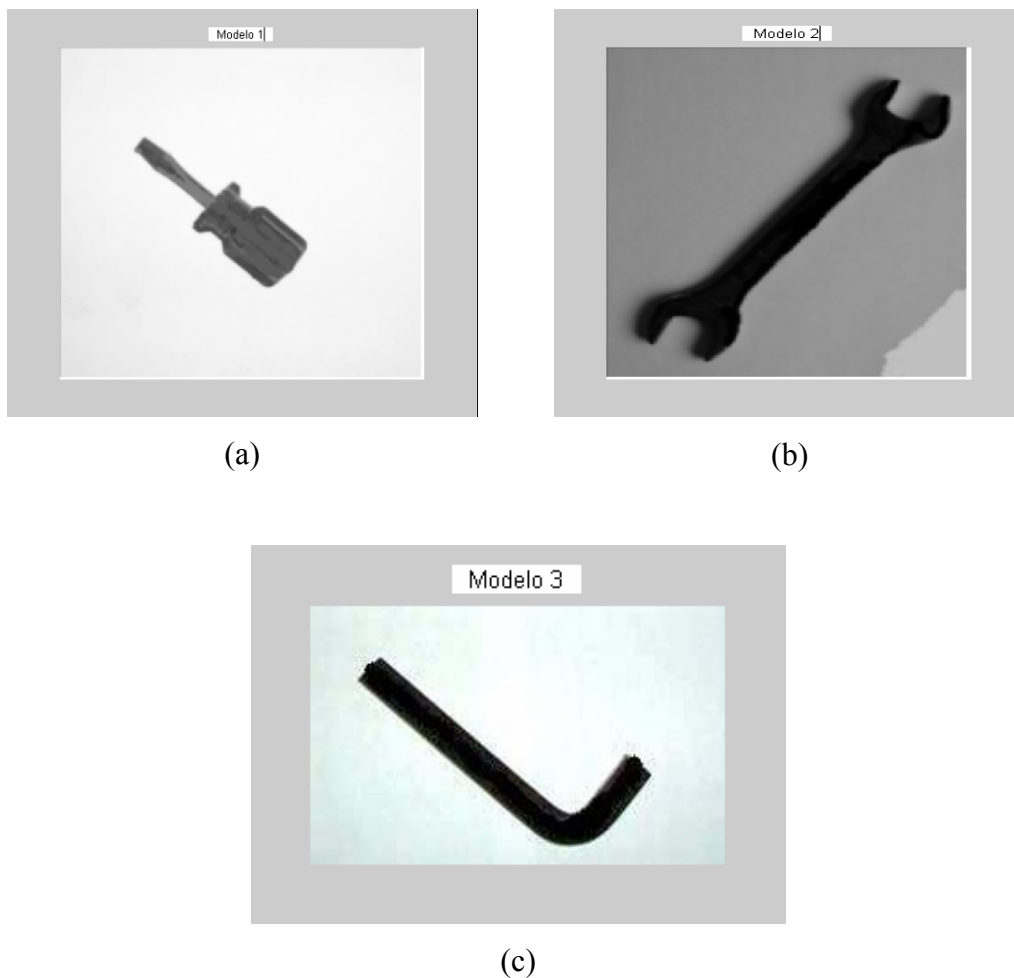
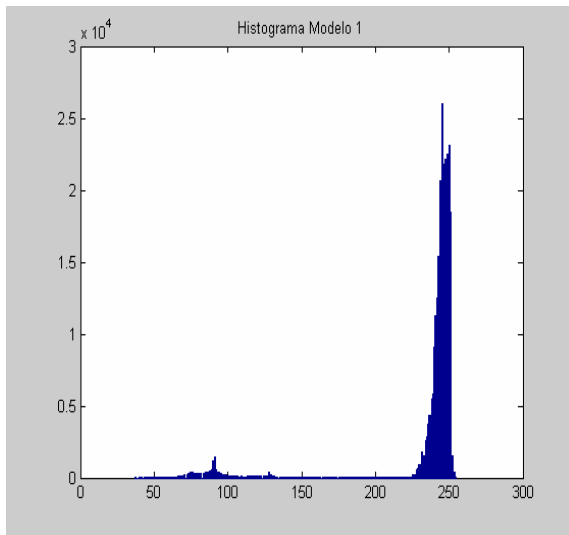


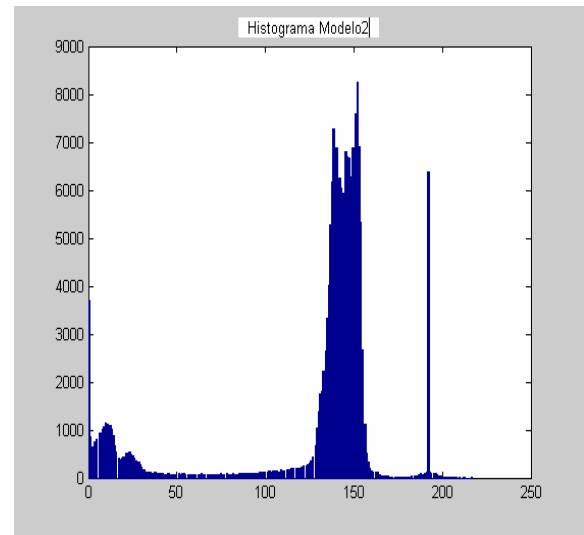
Figura 6.1 – Modelos de Objetos Utilizados. (a) Modelo 1. (b) Modelo 2. (c) Modelo 3.

(b) Extração de Vértices

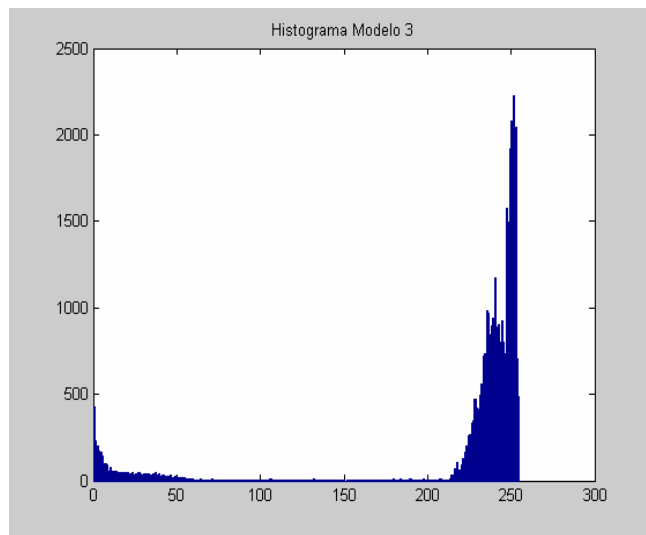
Para a extração de vértices foi utilizada a técnica de Trucco, apresentada no capítulo 2, item 2.4.2. A **Figura 6.2**, a **Figura 6.3** e a **Figura 6.4** apresentam o histograma, a imagem binarizada e o resultado da extração de vértices das imagens de cada modelo utilizado nos experimentos. O histograma da imagem é utilizado para a identificação do limiar de binarização. A **Figura 6.2** (a), (b) e (c) mostram o histograma dos modelos 1, 2 e 3, respectivamente.



(a)



(b)



(c)

Figura 6.2 – Histogramas. (a) Histograma Modelo 1. (b) Histograma Modelo 2. (c) Histograma Modelo 3.

A **Figura 6.3** (a), (b) e (c) mostram as imagens binarizadas. Para o modelo 1 foi utilizado limiar = 140, o modelo 2 o limiar = 90 e o modelo 3 o limiar = 170.

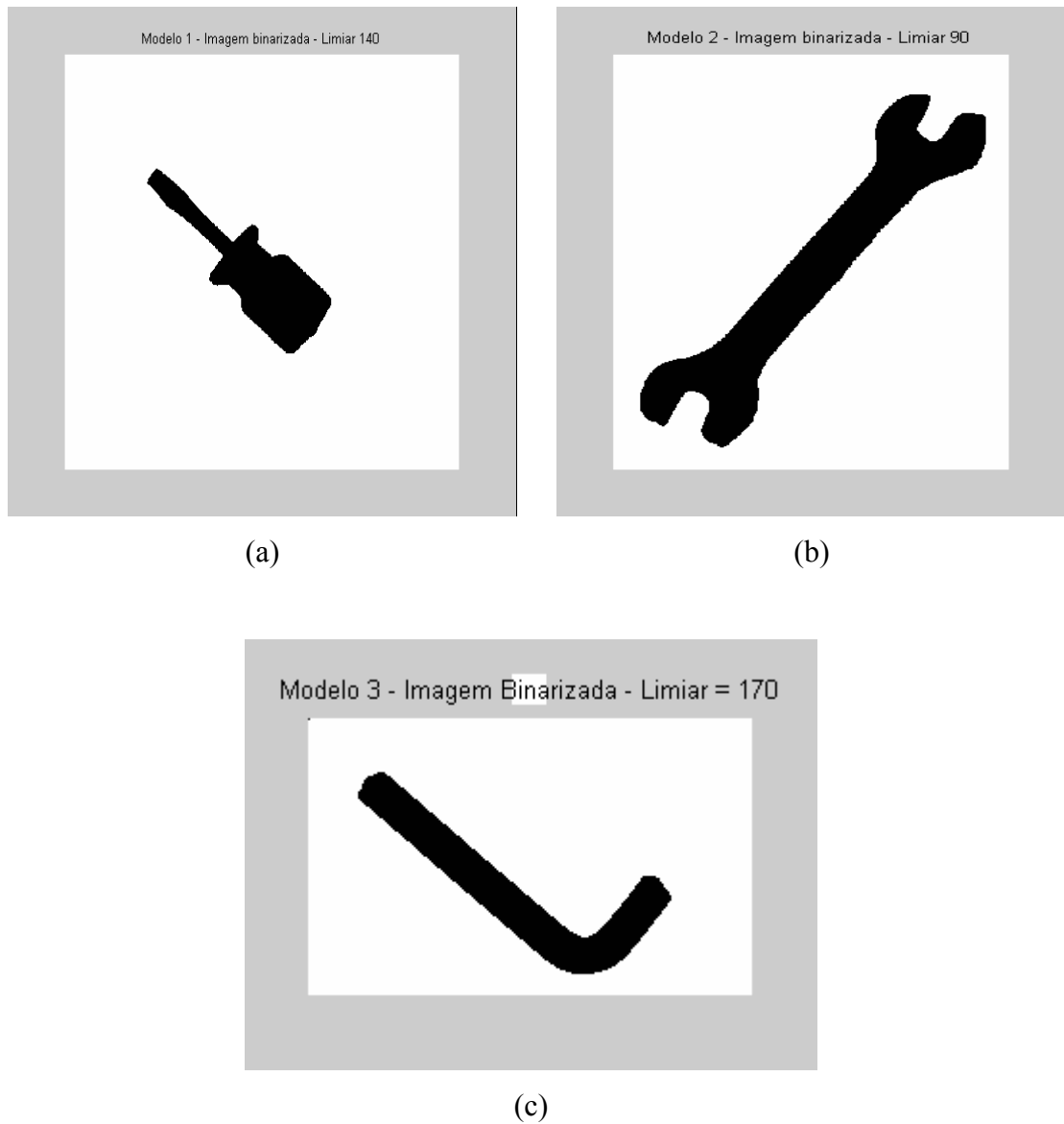


Figura 6.3 – Imagem binarizada. (a) Imagem binarizada do modelo 1. (b) Imagem binarizada do modelo 2. (c) Imagem binarizada do modelo 3.

A **Figura 6.4** (a) e (b) mostram os resultados da extração dos vértices para os modelos 1, 2 e 3, respectivamente ($\tau_1= 1,0527E+5$, $\tau_2= 1,0921E+5$ e $\tau_3= 6,2502E+4$). A Tabela 6.1, a Tabela 6.2 e a Tabela 6.3 mostram as posições dos vértices em pixel dos modelos 1, 2 e 3, respectivamente.

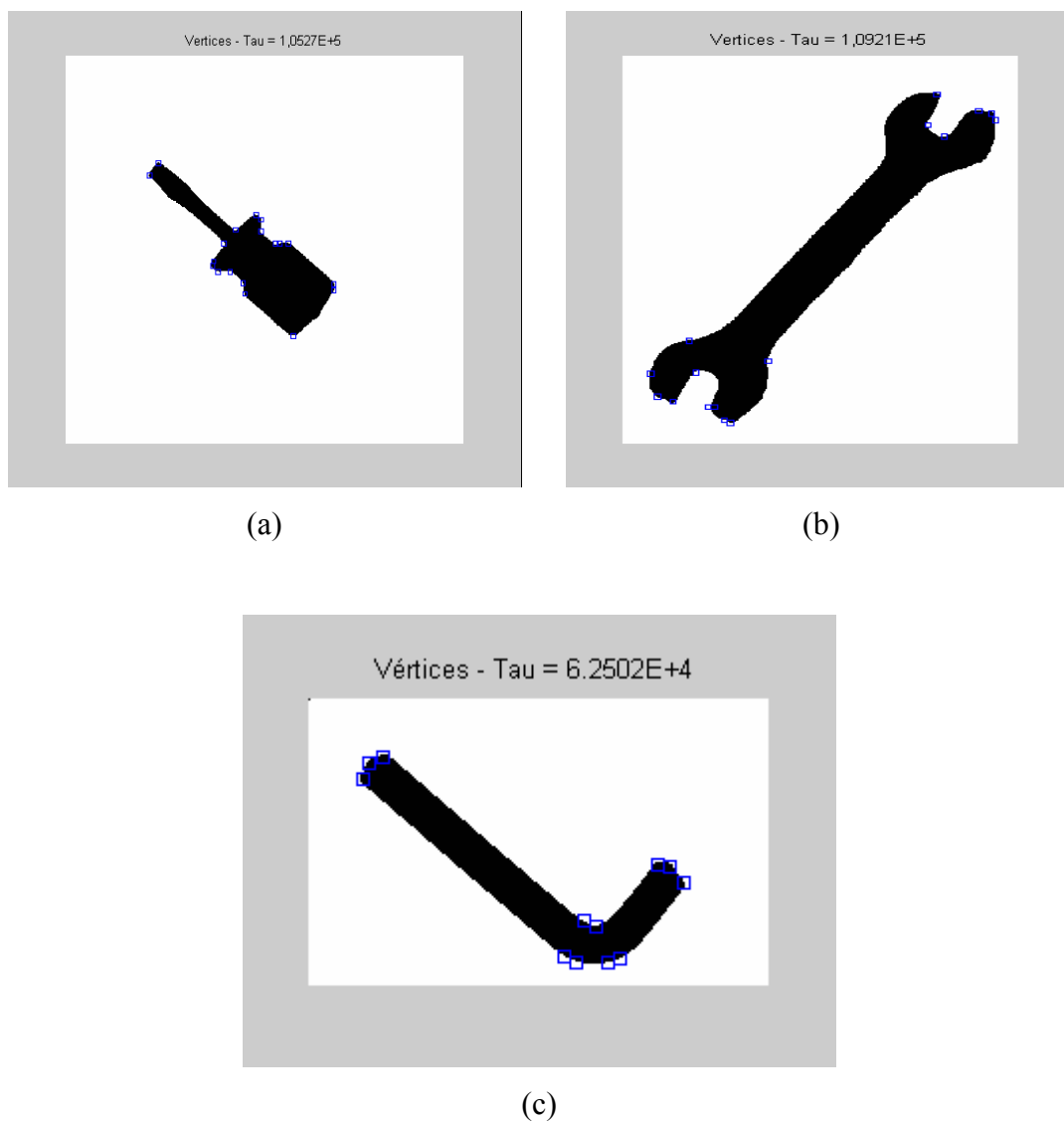


Figura 6.4 – após a extração dos vértices. (a) Modelo 1 – Vértices Tau (τ) = 1,0527E+5. (b) Modelo 2 – Vértices Tau (τ) = 1,0921E+5. (c) Modelo 3 – Vértices Tau (τ) = 6.2502E+4.

Tabela 6.1 – Vértices do modelo 1.

Posições dos vértices em pixel – Imagem (i,j)																			
i	142	158	210	216	229	231	247	247	247	247	270	276	284	284	299	300	308	313	368
j	127	115	260	267	232	267	216	286	293	304	202	201	208	225	242	366	366	245	311

Tabela 6.2 – Vértices do modelo 2.

Posições dos vértices em pixel – Imagem (i,j)																
i	48	68	71	79	85	99	348	373	387	388	416	422	429	429	445	448
j	311	352	365	369	302	318	66	144	73	28	35	50	85	91	101	107

Tabela 6.3 – Vértices do modelo 3.

Posições dos vértices em pixel – Imagem (i,j)												
i	30	33	41	84	85	93	112	115	130	131	133	133
j	38	31	28	176	182	189	139	145	129	157	135	151

c) Geometria dos Modelos – Utilizando Vértices

A configuração geométrica dos modelos, ou seqüência dos pontos, deve ser obtida através de uma ordenação consistente. Conforme apresentado no item 3.2, a ordenação deve ser feita no sentido horário e forma um polígono fechado. O polígono deve gerar uma geometria que seja o mais parecido possível com o formato da peça.

Três processamentos foram importantes nessa fase: o primeiro na ordenação dos vértices no sentido horário, o segundo na eliminação dos vértices colineares e finalmente a eliminação de vértices com uma distância menor que uma distância mínima.

A Tabela 6.4, a Tabela 6.5 e a Tabela 6.6 mostram os vértices ordenados dos modelos 1, 2 e 3, respectivamente. Neste caso já foram eliminados os vértices que estão no interior da peça e também já foram ordenados no sentido horário.

Tabela 6.4 – Vértices do modelo 1.

Posições dos vértices após ordenação (pixel)														
i	142	229	210	216	247	300	368	313	299	284	276	270	247	158
J	127	232	260	267	286	366	311	245	242	225	201	202	216	115

Tabela 6.5 – Vértices do modelo 2.

Posições dos vértices após ordenação (pixel)														
i	48	85	99	68	71	79	373	448	445	387	422	416	388	348
J	311	302	318	352	365	369	144	107	101	73	50	35	28	66

Tabela 6.6 – Vértices do modelo 3.

Posições dos vértices após ordenação (pixel)											
i	30	112	115	84	85	93	131	133	130	41	33
J	38	139	145	176	182	189	157	151	129	28	31

A Figura 6.5 (a), (b) e (c) mostram os modelos 1, 2 e 3, respectivamente, com os vértices após a eliminação dos pontos colineares e dos pontos com uma distância inferior a um valor mínimo (para os modelos 1 e 2 foi utilizado $D_{\min} = 20$, para o modelo 3 foi utilizado $D_{\min} = 6.15$).

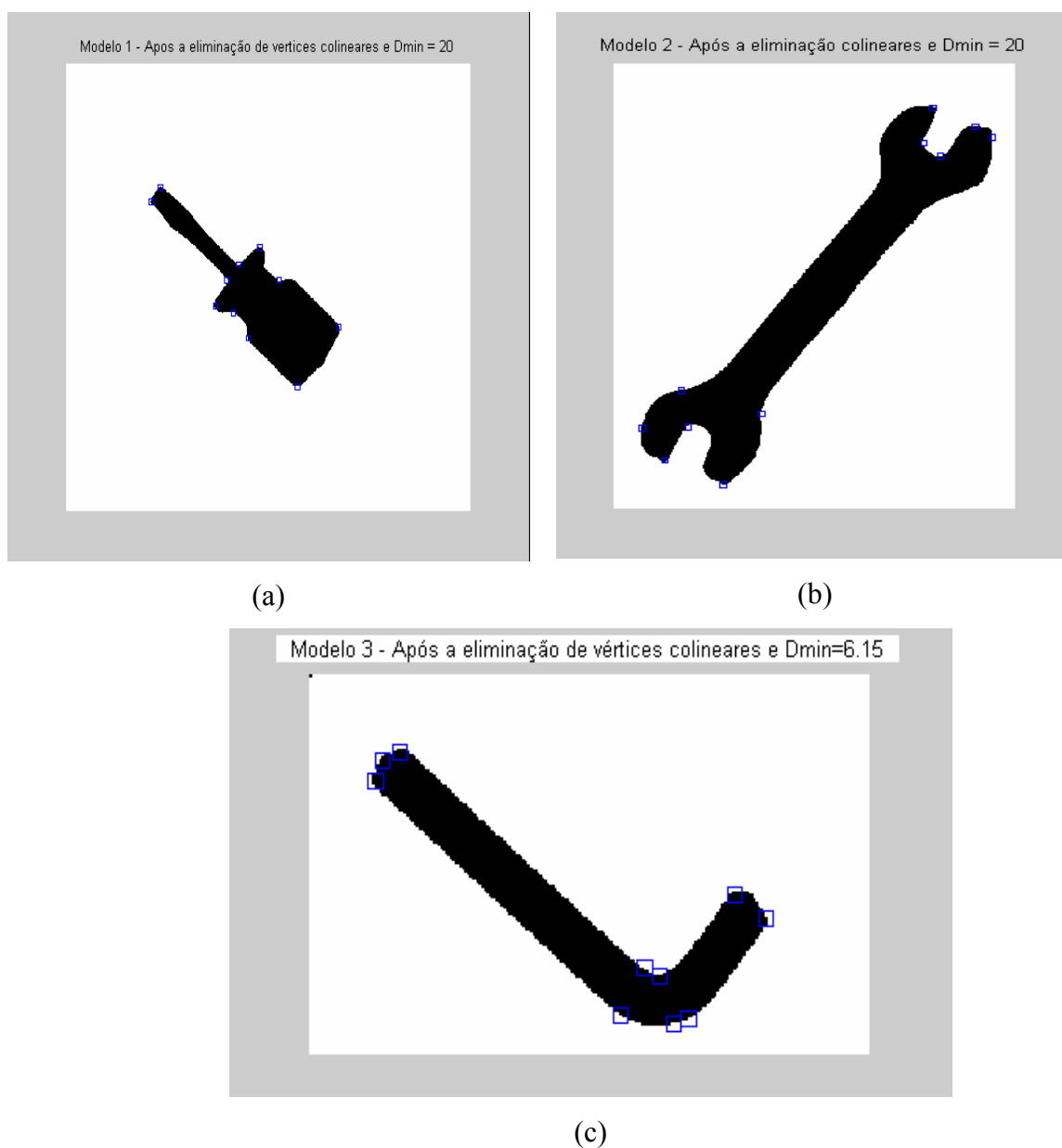


Figura 6.5 – Imagem dos objetos com os vértices necessários à geração da geometria do modelo. (a) Modelo 1. (b) Modelo 2. (c) Modelo 3.

A Tabela 6.7, a Tabela 6.8 e a Tabela 6.9 mostram os pontos obtidos na geração da geometria dos modelos 1, 2 e 3, respectivamente. E, finalmente os modelos 1, 2 e 3 estão sendo apresentados pela geometria, na **Figura 6.6** (a), (b) e (c), respectivamente.

Tabela 6.7 – Vértices do modelo 1.

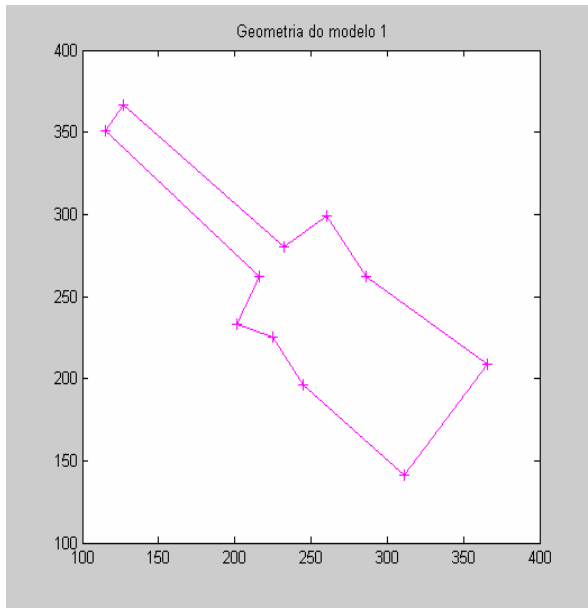
Posições dos vértices após ordenação (pixel)											
I	142	229	210	247	300	368	313	284	276	247	158
J	127	232	260	286	366	311	245	225	201	216	115

Tabela 6.8 – Vértices do modelo 2.

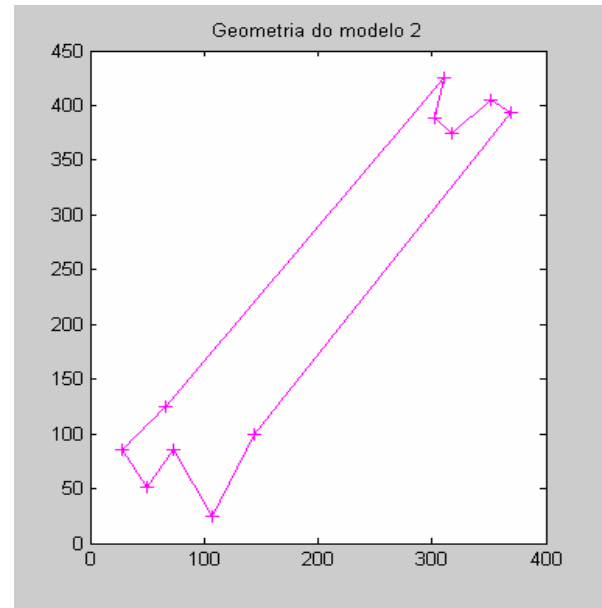
Posições dos vértices após ordenação (pixel)											
I	48	85	99	68	79	373	448	387	422	388	348
J	311	302	318	352	369	144	107	73	50	28	66

Tabela 6.9 – Vértices do modelo 3.

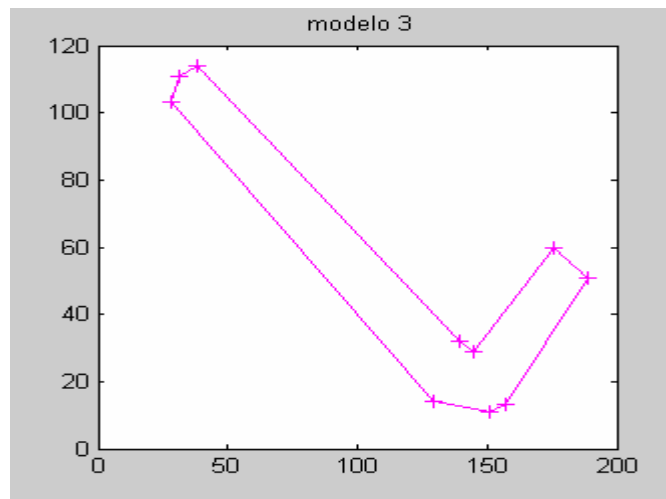
Posições dos vértices após ordenação (pixel)										
I	30	112	115	84	93	131	133	130	41	33
J	38	139	145	176	189	157	151	129	28	31



(a)



(b)



(c)

Figura 6.6 – Geometria dos Objetos. (a) Modelo 1. (b) Modelo 2. (c) Modelo 3.

d) Vetor de Invariantes

Utilizando-se a equação 3.3 obtiveram-se os resultados dos vetores invariantes (g_i) para os modelos 1, 2 e 3. A Tabela 6.10 apresenta esses resultados.

Tabela 6.10 – Resultado dos cálculos de Projetiva Invariante para os Modelos 1, 2 e 3.

Invariantes					
Modelo 1		Modelo 2		Modelo 3	
I ₁	I ₂	I ₁	I ₂	I ₁	I ₂
0.9911	1.9798	1.0171	0.7444	1.0029	0.92155
0.2780	-1.1803	-0.4724	-0.6600	-0.13124	-5.6
1.4426	0.6128	1.0493	0.9023	0.89796	1.286
0.6614	1.6768	-0.2135	-3.1862	0.21648	5.6422
0.8942	0.0226	0.9602	1.1625	0.83861	1.8127
-68.6991	-0.0071	-6.0180	0.6891	0.96136	1.0562
-1.2742	-1.8784	-0.0954	-2.2746	0.58054	1.7774
1.5304	0.7964	-0.0548	-16.8704	0.48163	17.31
-0.0169	-25.8764	0.9397	0.3705	1.002	0.99306
0.9791	0.4578	-0.4917	-2.0555	0.021808	35.872
3.1488	0.4904	0.9688	2.0055		

6.1.2 Processamento On-Line – Identificação de Objetos

Após a geração do banco de modelos (vetor de invariantes) inicia-se o processo de identificação de objetos numa imagem. O fluxograma apresentado no capítulo 3 (**Figura 3.8**) apresenta uma seqüência de tarefas necessárias à identificação de cada objeto, quando comparado a todos os modelos armazenados no processamento off-line.

Foram utilizadas várias imagens para testar o algoritmo de identificação. Uma imagem contendo apenas a ferramenta modelo 1, uma imagem contendo apenas a ferramenta modelo 2 e outra imagem contendo vários tipos de ferramentas.

a) Imagem da Ferramenta Modelo 1 – Translação

O experimento realizado utilizando-se da imagem do modelo 1 tem como objetivo testar o algoritmo com a imagem que gerou o banco de invariantes, porém, essa imagem será transladada. Para que esse experimento tornasse mais completo seria necessário que se obtivessem mais pontos característicos que quando da geração do banco de invariantes, então foi necessário flexibilizar os critérios (redução do limiar para obtenção de vértices e redução da distância entre pontos). Com a obtenção de mais pontos característicos, coube ao algoritmo identificar a melhor seqüência, ou seja, identificar o objeto através da seqüência

de pontos que mais se aproxima da geometria do modelo. Os resultados estão apresentados a seguir:

- A **Figura 6.7 (a)** mostra a imagem original do objeto, a **Figura 6.7 (b)** apresenta o resultado da extração e seleção de vértices que compõem a geometria do objeto a ser identificado, na imagem binarizada, e a **Figura 6.8** mostra a geometria do objeto e a tabela de posições em pixels ordenada no sentido horário.

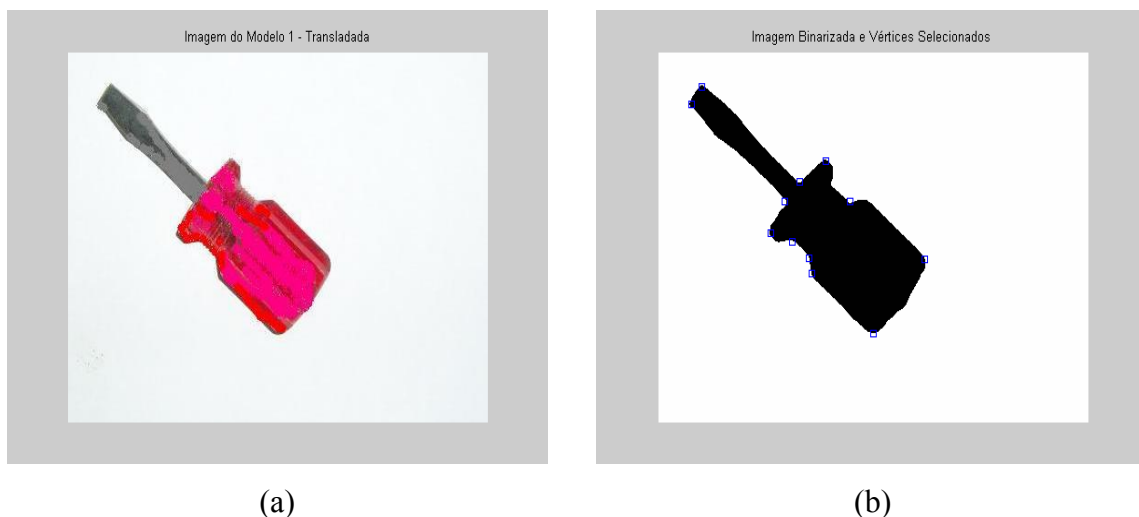


Figura 6.7 – (a) Imagem original do objeto. (b) Imagem binarizada após a extração dos vértices.

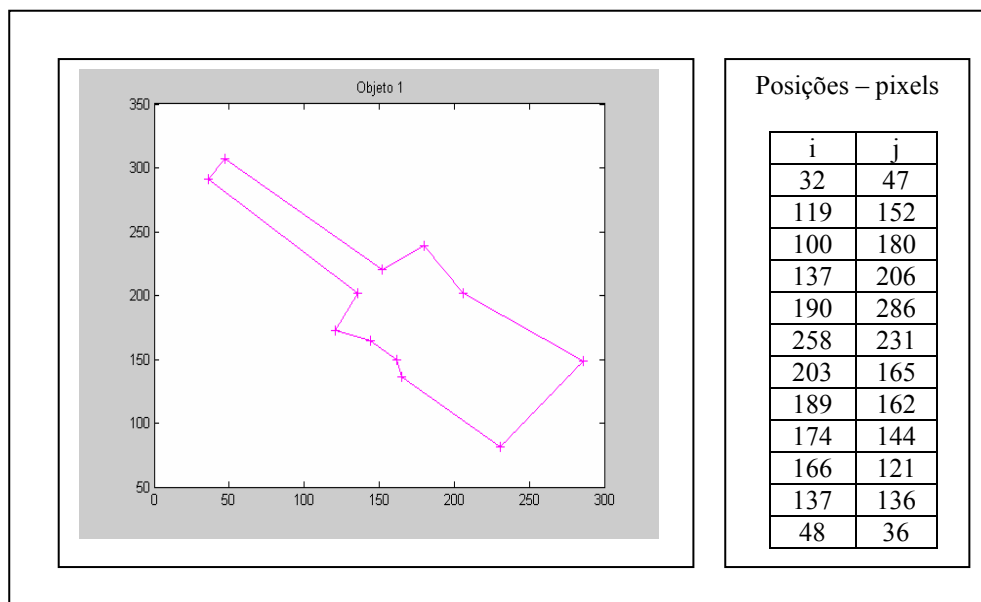


Figura 6.8 – Geometria do objeto e tabela dos vértices extraídos.

• A geometria do objeto foi formada com $M=12$ vértices e a geometria dos modelos 1 e 2 com $N=11$ vértices. Então seriam necessárias ${}_{12}C_{11}=12$ combinações para identificar o objeto. Para o modelo 3, a geometria é formada com $N=10$ vértices, e neste caso, seriam necessárias ${}_{12}C_{10}=66$ combinações para identificar o objeto. As tabelas 6.11, 6.12 e 6.13 mostram os erros obtidos entre os invariantes do objeto e os modelos 1, 2 e 3, respectivamente, para cada combinação. De cada tabela é possível identificar o erro mínimo das comparações: – Objeto com o modelo1, obteve-se erro mínimo igual 1.4854 – Objeto com o modelo2, obteve-se erro mínimo = 137.46 – Objeto com o modelo3, obteve-se erro mínimo = 26.951. O menor dos erros mínimos ($E_{\min}=1.4854$) é que identifica o objeto.

Tabela 6.11 – Erros encontrados em K-combinações do Objeto com o Modelo 1.

Resultados dos Erros – (K-combinação) $K = {}_{12}C_{11} = 12$									
Colunas de 1 a 10									
11970	11878	91.065	9.9912	1.4854	492.93	1014.1	4670.8	4670.4	4672.3
Colunas de 11 a 20									
4676.4	4819.3								

Tabela 6.12 – Erros encontrados em K-combinações do Objeto com o Modelo 2

Resultados dos Erros – (K-combinação) $K = {}_{12}C_{11} = 12$									
Colunas de 1 a 10									
168.4	211.35	159.13	209.75	139.92	137.46	144.11	220.78	222.82	219.25
Colunas de 11 a 20									
189.99	162.99								

Tabela 6.13 – Erros encontrados em K-combinações do Objeto com o Modelo 3

Resultados dos Erros – (K-combinação) $K = {}_{12}C_{11} = 66$									
Colunas de 1 a 10									
120.08	55.507	161.45	121.37	334.41	162.9	94.777	256.42	161.05	223.11
Colunas de 11 a 20									
26.951	159.42	164.92	162.45	163.49	28.512	154.29	164.48	160.5	162.77
Colunas de 21 a 30									
158.21	83.038	208.81	173.99	273.24	243.28	163.46	174.23	79.174	357.39
Colunas de 31 a 40									
322.56	421.96	392.04	311.77	322.44	76.512	107.44	163.88	129.04	228.43
Colunas de 41 a 50									
198.7	118.29	126.51	52.683	46.675	102.66	108.66	74.013	172.78	143.05
Colunas de 51 a 60									

62.894	67.989	52.833	55.553	57.188	120.3	92.86	61.076	159.16	131.42
Colunas de 61 a 70									
51.26	58.727	104.67	188.83	94.225	57.137				

• A **Figura 6.9** apresenta o resultado da comparação do objeto da **Figura 6.8** com os três modelos armazenados no processamento off-line. O objeto foi identificado como uma chave de fenda.

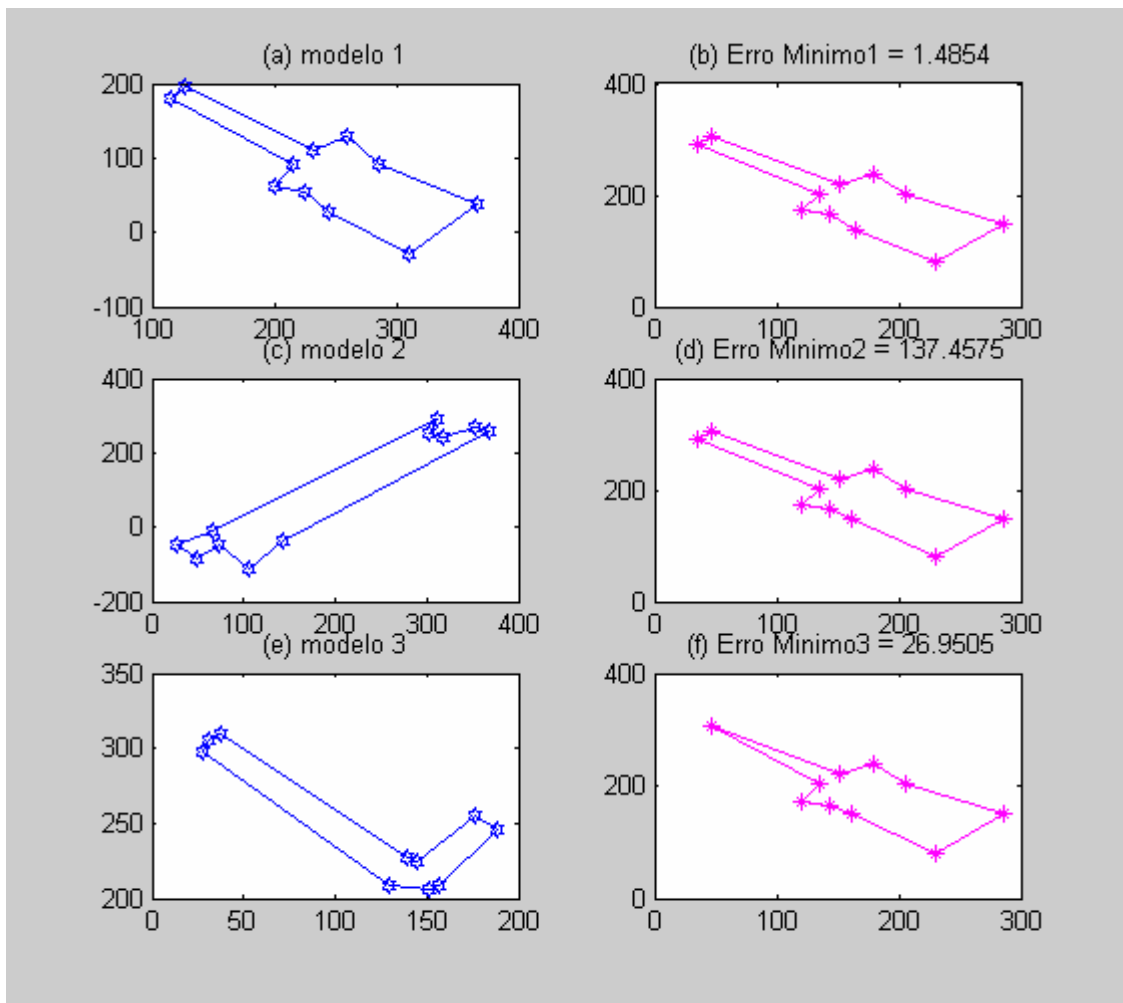


Figura 6.9 – Resultado das comparações – Identificação do Objeto

b) Imagem da Ferramenta Modelo 1 – Rotação 180°

O experimento foi realizado utilizando-se da imagem do modelo 1 e tem como objetivo testar o algoritmo com a imagem que gerou o banco de invariantes, porém essa imagem será rotacionada 180°. Neste experimento, também foi necessário flexibilizar os critérios para obtenção de mais pontos, e assim, o algoritmo deverá identificar a melhor seqüência, ou

seja, identificar o objeto através da seqüência de pontos que mais se aproxima da geometria do modelo.

A seqüência dos vértices que formaram a geometria da peça não iniciava na mesma posição quando da geração do modelo. Para resolver esse problema foi necessário descobrir qual seqüência de 5 vértices do objeto que mais se aproximava da seqüência inicial do modelo, e assim promover uma reordenação destes vértices. O critério utilizado foi a comparação das projetivas invariantes do objeto com as primeiras projetivas invariantes (I_1 e I_2) do modelo, o menor erro localizava a posição inicial da sequencia de vértices.

Os resultados da identificação do objeto estão apresentados a seguir:

- A **Figura 6.10(a)** mostra a imagem original do objeto, a **Figura 6.10(b)** apresenta o resultado da extração e seleção de vértices que compõem a geometria do objeto a ser identificado, na imagem binarizada, e a **Figura 6.11** mostra a geometria do objeto e a tabela de posições em pixels ordenada no sentido horário.

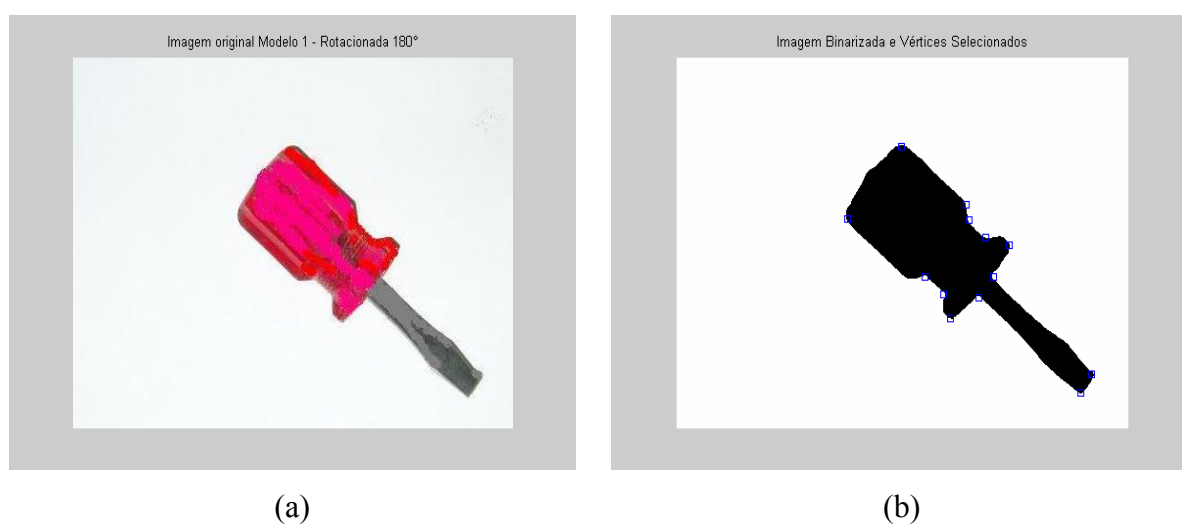


Figura 6.10 – (a) Imagem original do objeto. (b) Imagem binarizada após a extração dos vértices.

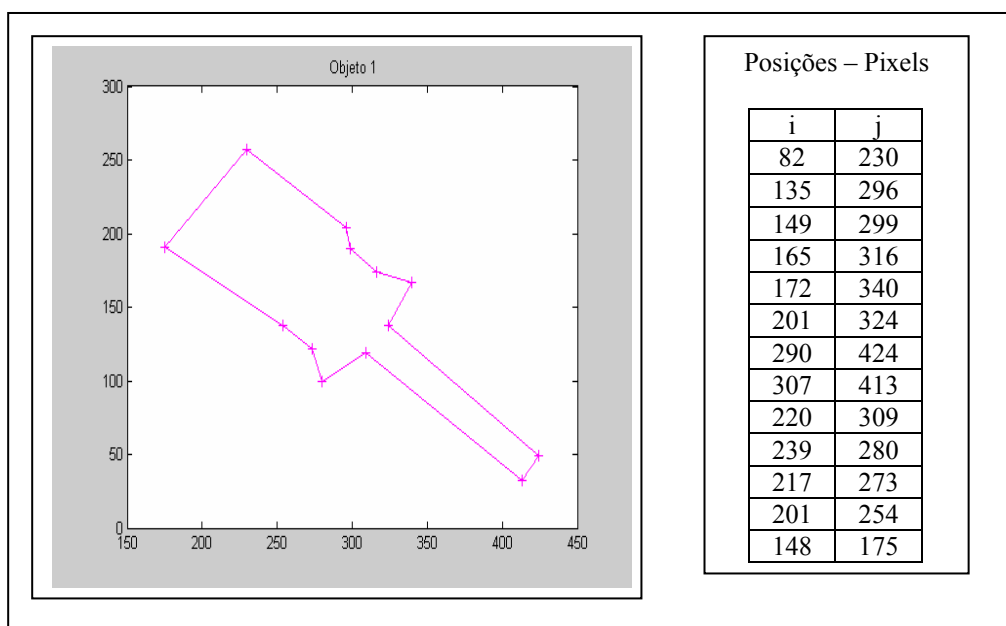


Figura 6.11 – Geometria do objeto e tabela dos vértices extraídos.

- A Tabela 6.14 mostra a nova seqüência dos pontos, alterada para que se coincida com a ordem dos pontos utilizados na geração do modelo.

Tabela 6.14 – Vértices do Objeto (pixel)

Nova seqüência dos vértices – após re-ordenação												
307	220	239	217	201	148	82	135	149	165	172	201	290
413	309	280	273	254	175	230	296	299	316	340	324	424

- A geometria do objeto foi formada com $M=13$ vértices e a geometria dos modelos 1 e 2 com $N=11$ vértices, então seriam necessárias ${}_{13}C_{11}=78$ combinações para identificar o objeto. Para o modelo 3, a geometria é formada com $N=10$ vértices, então neste caso, seriam necessárias ${}_{13}C_{10}=286$ combinações para identificar o objeto. As tabelas 6.15, 6.16 e 6.17 mostram os erros obtidos entre os invariantes do objeto e os modelos 1, 2 e 3, respectivamente, para cada combinação. De cada tabela é possível identificar o erro mínimo das comparações: – Objeto com o modelo1, obteve-se erro mínimo igual 8.9477 – Objeto com o modelo2, obteve-se erro mínimo =66.707 – Objeto com o modelo3 erro mínimo = 29.276. O menor dos erros mínimos ($E_{\min} = 8.9477$) é que identifica o objeto.

Tabela 6.15 – Erros encontrados em K-combinações do Objeto com o Modelo 1

Resultados dos Erros – (K-combinação) $K = {}_{13}C_{11} = 78$									
Colunas de 1 a 10									
20277	20286	20189	20101	20065	20371	20185	20090	20397	23546
Colunas de 11 a 20									
24019	23931	26183	20235	20234	527.21	439.46	211.6	189.56	146.67
Colunas de 21 a 30									
196.3	3772.2	3684.4	601.5	513.63	506.01	1791.9	743.92	4002.4	3914.7
Colunas de 31 a 40									
831.76	744.11	736.54	2175.6	766.68	3521.6	3275.1	3187.1	104.13	16.478
Colunas de 41 a 50									
8.9477	1394.4	766.7	1668.3	1668.2	3282.6	3192.4	109.61	21.957	14.427
Colunas de 51 a 60									
1399.8	770.87	1670.2	1668.2	1669.7	13583	13541	10466	10375	10367
Colunas de 61 a 70									
11753	780.89	4227.9	5414.9	1672.9	1673.3	13561	13581	10550	10563
Colunas de 71 a 78									
10556	11941	969.33	4416.7	5622.3	1865.1	2049.8	7968.9		

Tabela 6.16 – Erros encontrados em K-combinações do Objeto com o Modelo 2

Resultados dos Erros – (K-combinação) $K = {}_{13}C_{11} = 78$									
Colunas de 1 a 10									
13917	14298	13739	13804	13872	13623	13733	13843	13607	13767
Colunas de 11 a 20									
13642	13666	13719	13727	13719	66.707	109.3	158.78	251.42	188.74
Colunas de 21 a 30									
182.6	758.52	801.11	821.55	915.33	851.99	899.2	171.2	1075.9	1118.5
Colunas de 31 a 40									
1138.9	1232.6	1169.3	1259	172.76	3512	78.047	121.64	142.08	235.81
Colunas de 41 a 50									
172.5	188.63	174.25	197.76	200.57	87.531	124.1	144.49	238.22	174.91
Colunas de 51 a 60									
191	177	196.47	196.04	196.2	18588	18635	18652	18691	18627
Colunas de 61 a 70									
18644	132.43	1634.8	2334.4	136.36	108.11	18599	18639	18691	18639
Colunas de 71 a 78									
18592	18608	96.828	1601	2305.3	104.24	112.21	16494		

Tabela 6.17 – Erros encontrados em K-combinações do Objeto com o Modelo 3.

Resultados dos Erros – (K-combinação) $K = {}_{13}C_{10} = 286$									
Colunas de 1 a 10									
9553.4	9572	9536.9	9623.5	9607.7	9610.1	9638.7	9564.6	9634	9636.4
Colunas de 11 a 20									
9567.4	9582.7	9637.3	9542.7	9623.4	9645.4	9544.5	9623.9	9644	9637.6
Colunas de 21 a 30									
79.915	114.9	139.81	51.076	125.36	132.95	62.693	132.71	132.97	132.2
Colunas de 31 a 40									
47.221	127.72	133.06	131.36	134.74	882.55	898.37	923.62	851.35	937.33
Colunas de 41 a 50									
915.79	859.86	940.36	915	929.28	879.09	960.28	982.89	954.53	1091.3
Colunas de 51 a 60									
36.083	112.76	121.6	120.65	122.11	119.53	1253.2	1269.1	1294.3	1222
Colunas de 61 a 70									
1308	1286.3	1230.7	1311.2	1285.5	1295.1	1298.2	1379.3	1395.1	1372.1
Colunas de 71 a 80									
1703.9	39.629	116.31	125.33	124.62	125.86	123	445.89	522.57	520.48
Colunas de 81 a 90									
538.21	537.56	584.35	134.14	80.398	97.611	178.34	50.534	192	170.29
Colunas de 91 a 100									
59.223	195.21	169.5	179.2	40.758	177.42	193.14	170.28	172.1	29.276
Colunas de 101 a 110									
161.44	170.46	169.26	170.73	167.86	49.486	181.65	179.56	197.29	196.43
Colunas de 111 a 120									
176.77	183.04	44.888	370.01	367.92	385.85	385.14	365.71	371.31	66.755
Colunas de 121 a 130									
99.22	118.52	99.397	74.125	113.08	91.354	82.813	116.29	90.565	100.27
Colunas de 131 a 140									
64.306	98.456	114.16	91.303	92.977	53.72	83.366	92.376	91.173	92.516
Colunas de 141 a 150									
89.77	74.474	104.12	102.02	119.75	119.18	99.301	103.81	59.637	188.78
Colunas de 151 a 160									
186.67	204.6	204.08	184.54	188.75	47.726	74.745	138.82	136.72	154.64
Colunas de 161 a 170									
154.15	134.59	137.66	42.218	36.011	2270.8	2290	2243.8	2242.4	2257.6
Colunas de 171 a 180									
2236.1	2250.4	2260.8	2235.2	2243.6	2231.9	2243	2258.8	2234.6	2236.3
Colunas de 181 a 190									
46.702	53.288	62.376	59.897	61.24	58.447	5473.1	5479.7	5477.6	5494.1

Colunas de 191 a 200									
5493.5	5742.6	68.245	7976.6	7987.7	7985.7	8002.3	8001.8	8571.7	60.022
Colunas de 201 a 210									
1413	66.892	73.868	71.978	88.629	88.135	68.838	68.673	42.683	45.534
Colunas de 211 a 220									
54.173	55.699	53.601	69.689	69.195	49.714	52.131	30.413	34.855	38.135
Colunas de 221 a 230									
2265.6	2237.6	2245.6	2252.9	2256.7	2225.4	2247.9	2245.4	2226.1	2239.7
Colunas de 231 a 240									
2243.6	2232.9	2253.6	2227.7	2230.9	58.436	43.165	57.204	52.927	55.806
Colunas de 241 a 250									
53.014	5486.3	5471	5473.9	5488.6	5489.5	5738.6	65.728	7997.6	7970.3
Colunas de 251 a 260									
7973.2	7988.1	7989.1	8559	48.383	2448.2	83.008	57.429	60.323	75.175
Colunas de 261 a 270									
76.217	56.92	59.074	99.812	206.47	67.41	49.255	52.973	73.614	74.655
Colunas de 271 a 280									
55.175	59.818	61.584	126.54	98.053	13327	13324	13322	13337	13338
Colunas de 281 a 286									
13318	71.94	756.12	1095.1	47.024	57.356				

- A **Figura 6.12** apresenta o resultado da comparação do objeto da **Figura 6.11** com os três modelos armazenados no processamento off-line. O objeto foi identificado como uma chave de fenda.

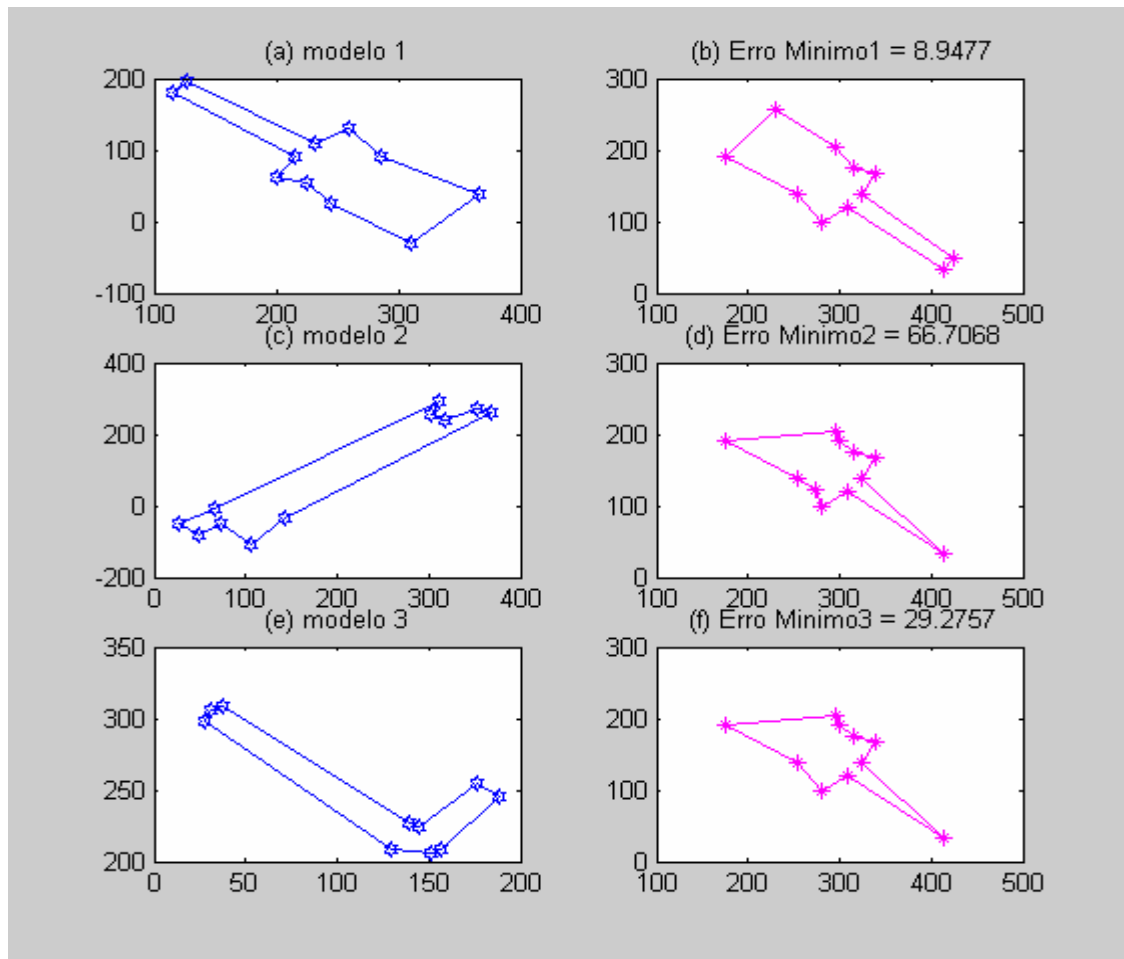


Figura 6.12 – Resultado das comparações – Identificação do Objeto.

c) Imagem da Ferramenta Modelo 2 – Translação e Rotação 90°

Esse experimento foi realizado com o mesmo objetivo do experimento anterior, porém com imagem transladada e rotacionada em 90°. Sendo assim, também foi necessário flexibilizar os parâmetros para a obtenção de mais pontos característicos, e os resultados serão apresentados a seguir:

- A **Figura 6.13(a)** mostra a imagem original do objeto a ser identificado, a **Figura 6.13(b)** apresenta o resultado da extração e seleção de vértices que compõem a geometria do

objeto a ser identificado, na imagem binarizada, e a **Figura 6.14** mostra a geometria do objeto e a tabela de posições em pixels ordenada no sentido horário.

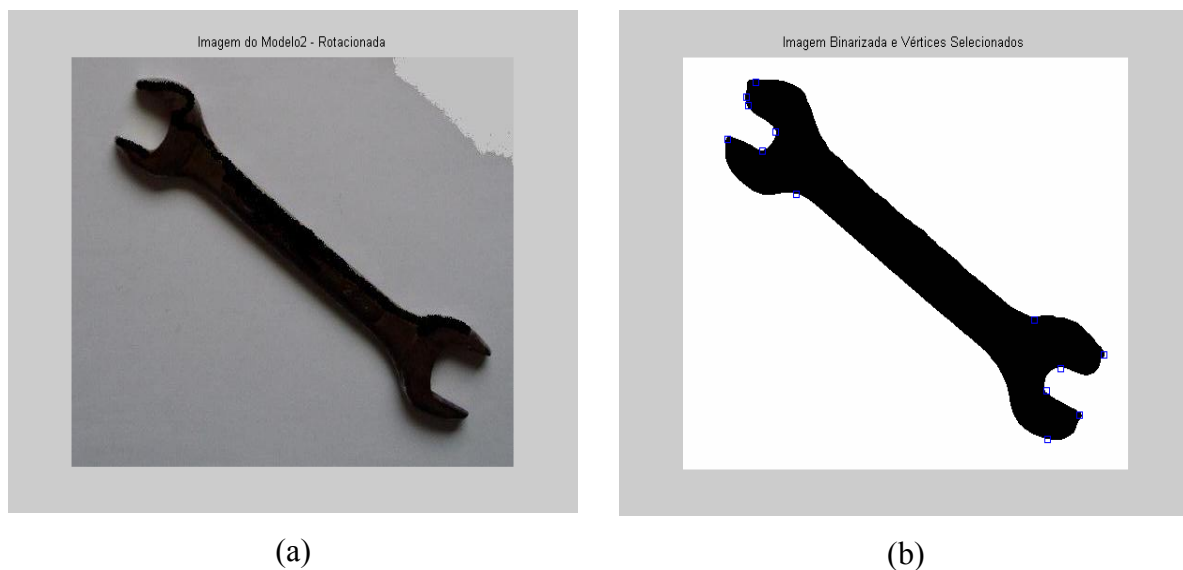


Figura 6.13 – (a) Imagem original do objeto. (b) Imagem binarizada após a extração dos vértices.

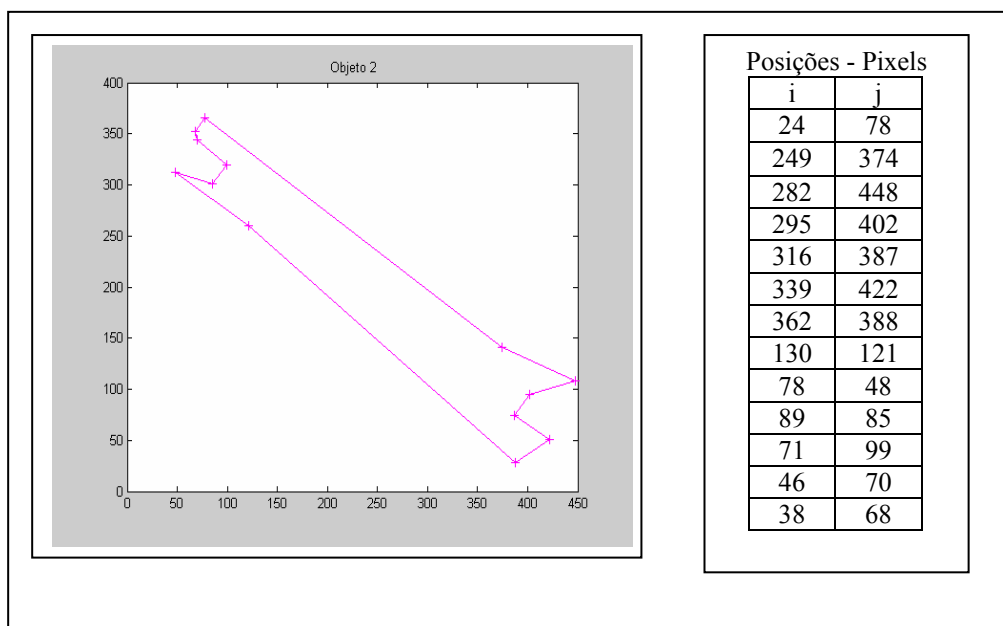


Figura 6.14 – Geometria do objeto e tabela dos vértices extraídos.

- A Tabela 6.18 mostra a nova seqüência dos pontos, alterada para que se coincida com a ordem dos pontos utilizados na geração do modelo.

Tabela 6.18 – Vértices do Objeto (pixel)

Nova seqüência dos vértices – após re-ordenação												
78	89	71	46	24	249	282	295	316	339	354	362	130
48	85	99	70	78	374	448	402	387	422	416	388	121

• A geometria do objeto foi formada com $M=13$ vértices e a geometria dos modelos 1 e 2 com $N=11$ vértices. Então seriam necessárias ${}_{13}C_{11}=78$ combinações para identificar o objeto. Para o modelo 3, a geometria é formada com $N=10$ vértices, então neste caso, seriam necessárias ${}_{13}C_{10}=286$ combinações para identificar o objeto. A Tabela 6.19, a Tabela 6.20 e a Tabela 6.21 mostram os erros obtidos entre os invariantes do objeto e os modelos 1, 2 e 3 respectivamente, para cada combinação. De cada tabela é possível identificar o erro mínimo das comparações: – Objeto com o modelo1, obteve-se erro mínimo igual 44.0654 – Objeto com o modelo2, obteve-se erro mínimo = 14.6696 – Objeto com o modelo3, obteve-se erro mínimo = 16.4554. O menor dos erros mínimos ($E_{\min} = 14.6696$) é que identifica o objeto.

Tabela 6.19 – Erros encontrados em K-combinações do Objeto com o Modelo 1.

Resultados dos Erros – (K-combinação) $K = {}_{13}C_{11} = 78$									
Colunas de 1 a 10									
567.42	557.47	432.98	489.96	431.7	361.31	268.88	206.15	178.35	181.45
Colunas de 11 a 20									
350.8	217.78	174.08	176.05	3238.5	214.21	80.041	44.0654	88.932	330.61
Colunas de 21 a 30									
1031.2	410.01	275.85	228.3	247.66	562.52	536.5	665.05	406.32	272.15
Colunas de 31 a 40									
224.61	240.77	571.85	658.23	676.85	688.59	403.22	269.08	221.53	237.69
Colunas de 41 a 50									
568.7	505.89	642.58	660.9	662.7	406.74	273.38	225.91	242.07	573.07
Colunas de 51 a 60									
509.26	646.46	665.27	660.94	660.03	396.12	265.67	218.52	228.21	559.22
Colunas de 61 a 70									
495.4	631.46	649.72	662.11	659.37	805.02	359.59	286.4	239.36	245.65
Colunas de 71 a 78									
576.46	512.65	648.71	666.92	669.94	670.86	675.41	709.95		

Tabela 6.20 – Erros encontrados em K-combinações do Objeto com o Modelo 2.

Resultados dos Erros – (K-combinação) $K = {}_{13}C_{11} = 78$									
Colunas de 1 a 10									
35.582	76.339	61.129	55.914	58.65	47.484	26.895	46.391	37.452	37.85
Colunas de 11 a 20									
14.6696	45.988	38.067	37.544	81.34	24.525	55.669	71.969	54.711	51.285
Colunas de 21 a 30									
102.67	25.297	56.441	69.981	57.182	54.119	99.753	88.449	47.149	78.292
Colunas de 31 a 40									
91.833	76.274	72.855	191.86	113.46	105.49	46.919	77.599	91.139	75.581
Colunas de 41 a 50									
72.013	89.326	91.59	88.333	94.086	58.145	88.975	102.47	86.915	83.347
Colunas de 51 a 60									
99.957	102.78	100.75	93.843	93.037	43.659	105.67	100.55	84.466	80.898
Colunas de 61 a 70									
97.508	100.54	97.338	105.42	100.68	134.99	44.162	104.09	124.77	85.751
Colunas de 71 a 78									
83.317	99.927	102.96	99.662	101.87	97.122	94.837	90.639		

Tabela 6.21 – Erros encontrados em K-combinações do Objeto com o Modelo 3.

Resultados dos Erros – (K-combinação) $K = {}_{13}C_{10} = 286$									
Colunas de 1 a 10									
64.197	64.538	65.004	214.81	65.868	66.785	129.84	67.108	130.79	143.59
Colunas de 11 a 20									
64.927	73.762	160.06	68.952	160.11	196.65	85.632	6303.4	357.2	345.77
Colunas de 21 a 30									
68.121	103.09	82.178	72.829	80.306	82.157	60.729	77.684	78.64	78.387
Colunas de 31 a 40									
122.54	99.931	104.38	99.801	86.566	64.74	70.335	77.889	65.535	78.694
Colunas de 41 a 50									
80.169	64.645	83.507	81.789	82.411	120	102.15	102.38	101.54	97.947
Colunas de 51 a 60									
112.34	94.073	95.856	91.841	83.081	145.93	86.479	92.074	99.627	84.323
Colunas de 61 a 70									

97.482	97.83	82.548	101.41	99.714	101.21	187.34	169.49	177.73	175.9
Columnas de 71 a 80									
129.5	134.49	116.22	152.8	122.29	97.585	162.07	126.54	108.27	111.42
Columnas de 81 a 90									
109.75	98.778	158.18	157.5	78.817	82.377	81.565	74.627	79.419	79.767
Columnas de 91 a 100									
72.683	83.178	81.481	83.193	105.92	79.707	87.94	86.25	91.678	108.83
Columnas de 101 a 110									
82.191	118.76	88.179	75.982	144.99	104.07	77.431	80.574	78.665	77.72
Columnas de 111 a 120									
140.86	140.93	94.469	42.935	46.078	32.76	26.149	89.579	109.94	94.47
Columnas de 121 a 130									
87.906	89.758	76.538	82.004	74.375	74.643	80.06	78.134	76.358	78.069
Columnas de 131 a 140									
112.78	74.148	82.302	80.612	86.203	115.1	76.039	112.53	81.948	69.839
Columnas de 141 a 150									
138.23	111.1	72.04	75.103	73.195	72.314	134.82	132.34	92.683	38.784
Columnas de 151 a 160									
41.848	28.529	21.906	85.587	105.68	90.171	101.59	42.756	45.996	32.677
Columnas de 161 a 170									
25.98	89.507	109.62	94.257	140.95	47.137	57.995	441.96	45.345	440.7
Columnas de 171 a 180									
439.77	43.687	444.2	441.49	442.77	76.408	440.21	447.43	445.31	450.9
Columnas de 181 a 190									
74.895	438.27	473.84	442.82	430.71	498.4	69.064	432.44	434.58	432.24
Columnas de 191 a 200									
431.35	494.53	494.96	69.546	37.727	39.861	26.11	19.487	100.43	104.11
Columnas de 201 a 210									
87.498	81.667	34.805	36.904	23.153	16.4554	78.568	97.819	82.495	139.62
Columnas de 211 a 220									
86.297	39.429	41.599	27.85	21.153	83.004	101.82	86.484	139.72	139.95
Columnas de 221 a 230									
41.301	57.002	603.48	49.108	600.37	599.98	47.379	603.95	601.58	602.75
Columnas de 231 a 240									
63.599	602.66	609.91	605.13	610.9	62.086	600.72	636.31	602.64	590.71
Columnas de 241 a 250									
658.39	56.176	594.81	596.97	591.98	591.27	654.44	655.12	55.88	61.466
Columnas de 251 a 260									
63.624	47.223	40.773	121.72	125.41	108.86	70.824	37.449	39.585	23.184
Columnas de 261 a 270									
16.66	78.773	97.975	82.634	138.4	74.758	50.272	52.367	35.939	29.415

Colunas de 271 a 280									
91.266	110.24	95.105	139.1	138.75	103.8	370.63	370.92	349.83	343.7
Colunas de 281 a 286									
405.55	425.42	410.02	146.38	180.38	142.98				

• A **Figura 6.15** apresenta o resultado da comparação do objeto da **Figura 6.14** com os três modelos armazenados no processamento off-line. O objeto foi identificado como uma chave de boca.

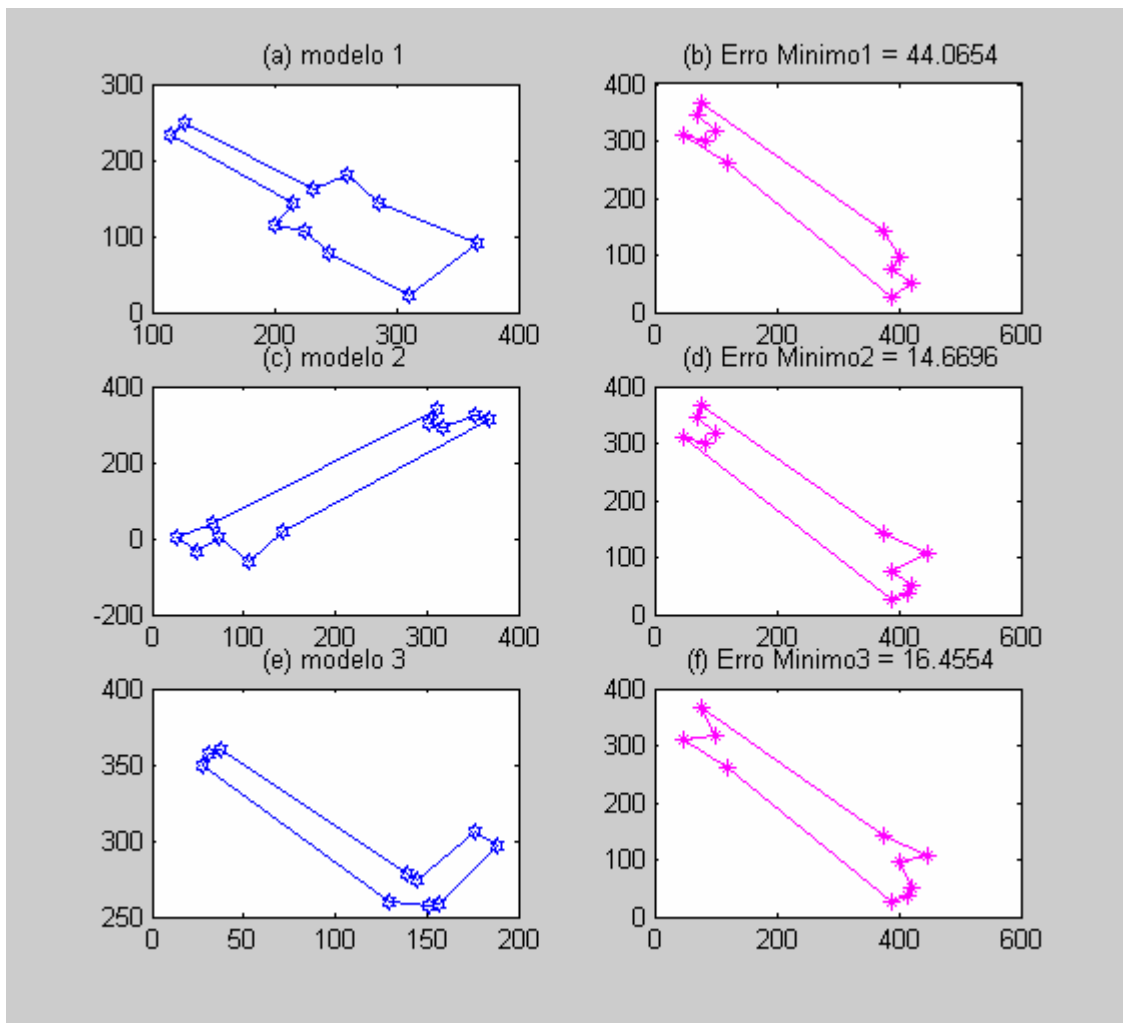


Figura 6.15 – Resultado das comparações – Identificação do Objeto.

d) Imagem da Ferramenta Modelo 3 – Translação e Rotação 90°

Esse experimento foi realizado com o mesmo objetivo dos experimentos anteriores, porém com imagem do modelo 3 transladada e rotacionada em 90°. Sendo assim, também foi

necessário flexibilizar os parâmetros para a obtenção de mais pontos característicos, e os resultados estão apresentados a seguir:

- A **Figura 6.16(a)** mostra a imagem original do objeto após a translação e a rotação, a **Figura 6.16(b)** apresenta o resultado da extração e seleção de vértices que compõem a geometria do objeto a ser identificado, na imagem binarizada, e a **Figura 6.16** mostra a geometria do objeto e a tabela de posições em pixels ordenada no sentido horário.

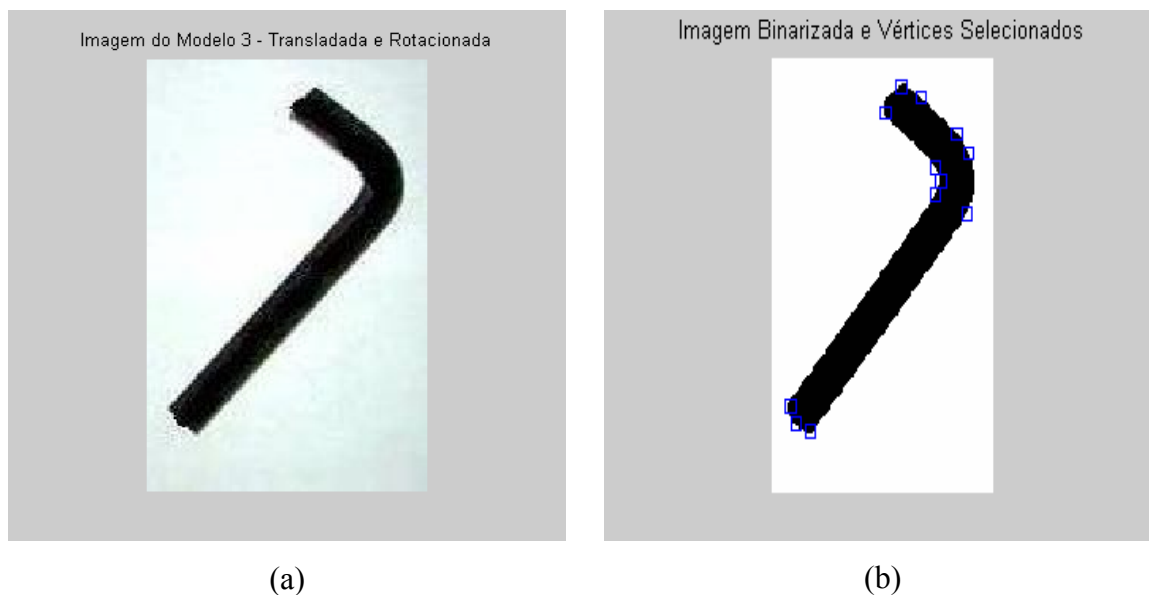


Figura 6.16 – (a) Imagem original do objeto – Transladada e Rotacionada. (b) Imagem binarizada após a extração dos vértices.

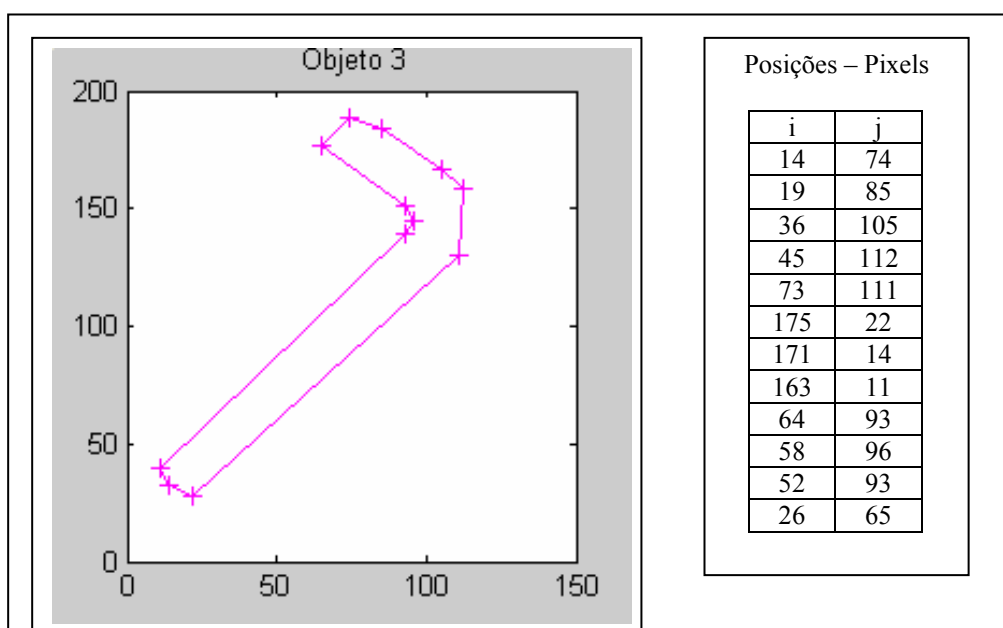


Figura 6.17 – Geometria do objeto e tabela dos vértices extraídos.

- A Tabela 6.22 mostra a nova seqüência dos pontos, alterada para que se coincida com a ordem dos pontos utilizados na geração do modelo.

Tabela 6.22 – Vértices do Objeto (pixel).

Nova seqüência dos vértices – após re-ordenação											
175	171	163	64	58	52	26	14	19	36	45	73
22	14	11	93	96	93	65	74	85	105	112	111

- A geometria do objeto foi formada com $M=12$ vértices e a geometria dos modelos 1 e 2 com $N=11$ vértices. Então seriam necessárias ${}_{12}C_{11}=12$ combinações para identificar o objeto. Para o modelo 3, a geometria é formada com $N=10$ vértices, então neste caso, seriam necessárias ${}_{12}C_{10}=66$ combinações para identificar o objeto. A Tabela 6.23, a Tabela 6.24 e a Tabela 6.25 mostram os erros obtidos entre os invariantes do objeto e os modelos 1, 2 e 3, respectivamente, para cada combinação. De cada tabela é possível identificar o erro mínimo das comparações: – Objeto com o modelo1, obteve-se erro mínimo igual 369.0523 – Objeto com o modelo2, obteve-se erro mínimo = 62.4333 – Objeto com o modelo3, obteve-se erro mínimo = 34.2526. O menor dos erros mínimos ($E_{\min} = 34.2526$) é que identifica o objeto.

Tabela 6.23 – Erros encontrados em K-combinações do Objeto com o Modelo 1.

Resultados dos Erros – (K-combinação) $K = {}_{12}C_{11} = 12$									
Colunas de 1 a 10									
1781.7	1662.6	1841.8	1436.3	369.0523	971.79	455.61	439.91	1951.6	1935.3
Colunas de 11 a 20									
1921.5	1938.6								

Tabela 6.24 – Erros encontrados em K-combinações do Objeto com o Modelo 2.

Resultados dos Erros – (K-combinação) $K = {}_{12}C_{11} = 12$									
Colunas de 1 a 10									
1126.8	1105.7	1106.4	1103.2	68.951	377.42	105.76	62.4333	4745.5	4740.3
Colunas de 11 a 20									
4737.1	4739.8								

Tabela 6.25 – Erros encontrados em K-combinações do Objeto com o Modelo 3.

Resultados dos Erros – (K-combinação) $K = {}_{12}C_{10} = 66$									
Colunas de 1 a 10									
1234.3	1246.7	1263.8	1261.8	1244.6	1242.4	61.618	61.189	56.131	67.45
Colunas de 11 a 20									
224.48	226.33	221.19	231.22	52.822	48.397	39.732	39.22	46.13	43.202
Colunas de 21 a 30									
43.505	43.39	34.726	34.3526	40.434	44.05	39.038	43.268	4658	4653.4
Colunas de 31 a 40									
4653.1	4656	46.646	814.08	42.499	42.497	4684.9	4702.5	4697.6	4700.6
Colunas de 41 a 50									
92.181	858.89	112.78	81.386	1198.3	4673.9	4693.4	4689.9	4692.9	84.472
Colunas de 51 a 60									
851.17	101.08	75.105	1191.8	1171.7	4679.5	4702.9	4697.8	4700.8	92.442
Colunas de 61 a 70									
859.14	114.59	84.208	1202.4	1172.3	1172.8				

- A **Figura 6.18** apresenta o resultado da comparação do objeto da **Figura 6.18** com os três modelos armazenados no processamento off-line. O objeto foi identificado como uma chave Allen

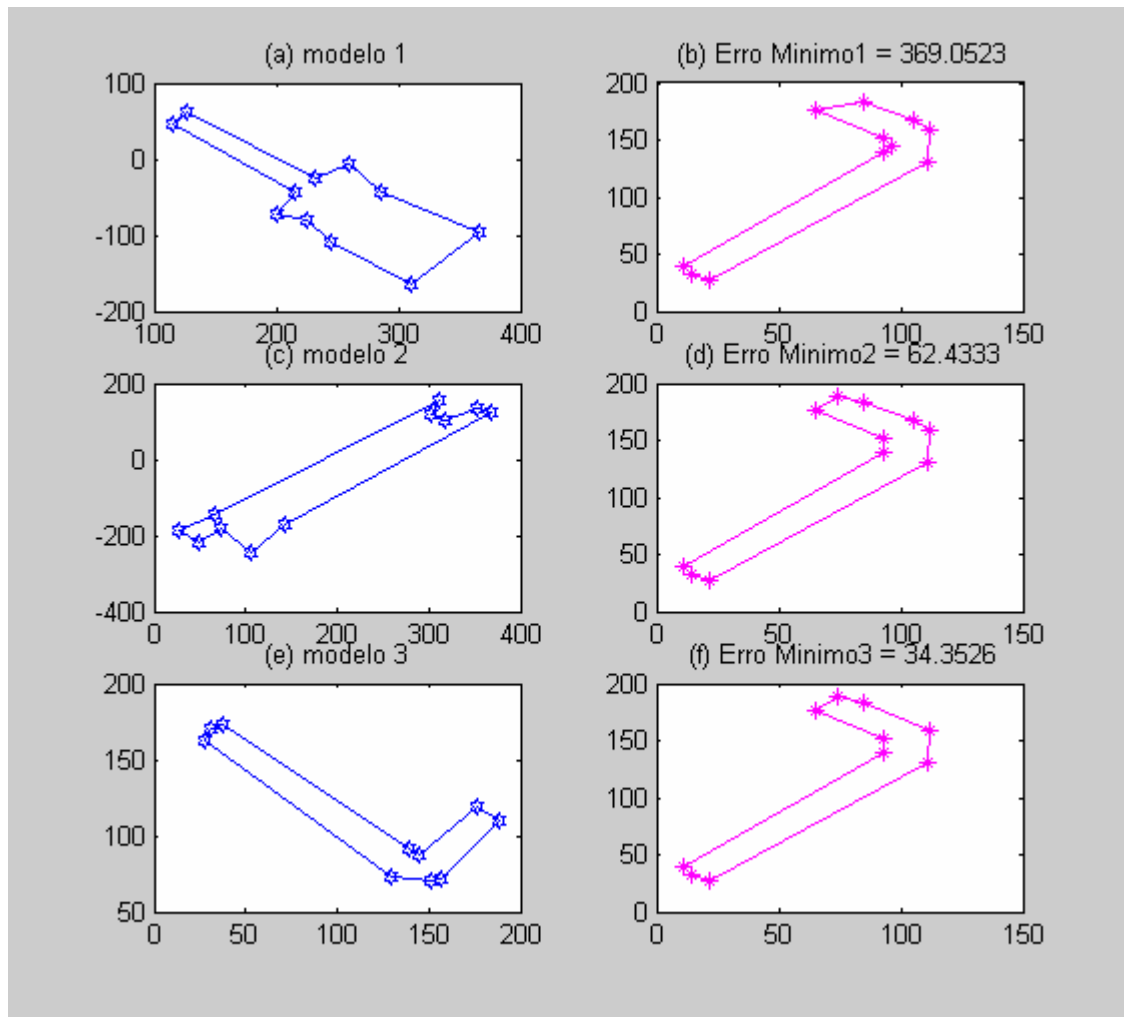


Figura 6.18 – Resultado das comparações – Identificação do Objeto.

e) Imagem de Várias Ferramentas

Utilizou-se para esse experimento uma imagem contendo vários objetos com a finalidade de identificar cada um deles através do algoritmo descrito no capítulo 3. Inicialmente, foi necessário gerar uma lista de pontos característicos (vértices) de cada objeto. Calculam-se então, todos os invariantes de cada lista para que as mesmas sejam re-ordenadas. Essas listas serão re-ordenadas tomando-se como referência a menor invariante gerada em cada uma delas. Para cada lista de pontos, calcula-se o número de combinações possíveis entre o

número de pontos do objeto e o número de pontos do modelo. Em seguida, para cada combinação, calcula-se o vetor de invariantes e um erro associado ao modelo, gerando uma lista de erros. O erro mínimo define a melhor comparação para um determinado modelo. Gera-se uma lista de erros mínimos. O menor dos erros mínimos é que identifica o objeto. Os resultados serão apresentados a seguir:

- A **Figura 6.19** mostra a imagem original. A **Figura 6.20** mostra o histograma dessa imagem, necessário para identificação do limiar que será utilizado para sua binarização. A **Figura 6.21** apresenta a imagem binarizada (limiar = 90) após a extração dos vértices.

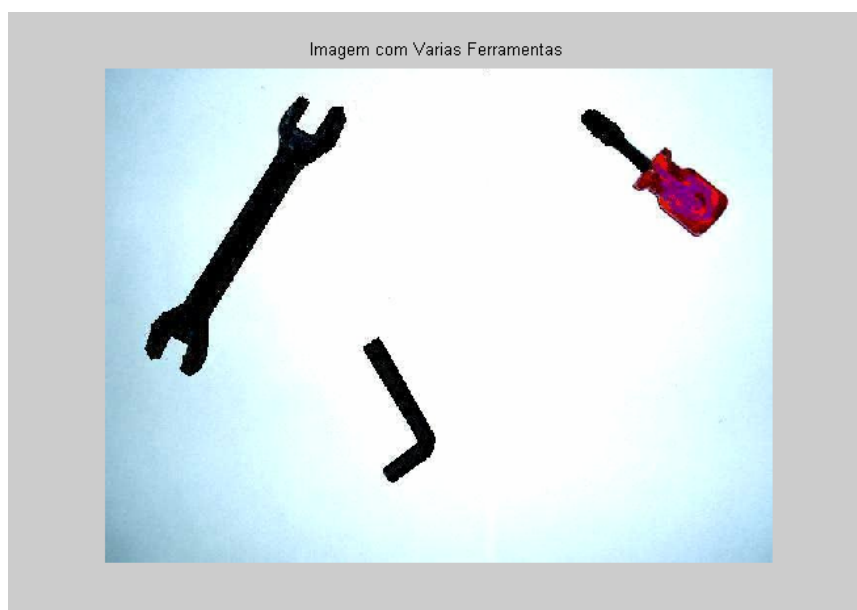


Figura 6.19 – Imagem original.

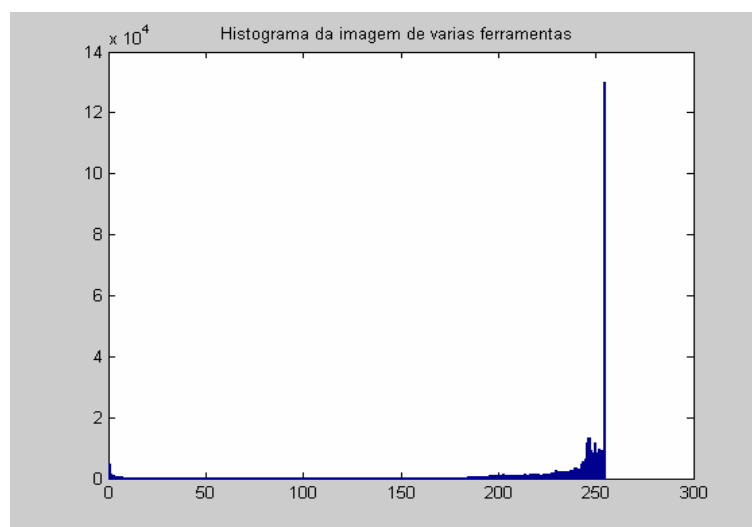


Figura 6.20 – Histograma da imagem original.

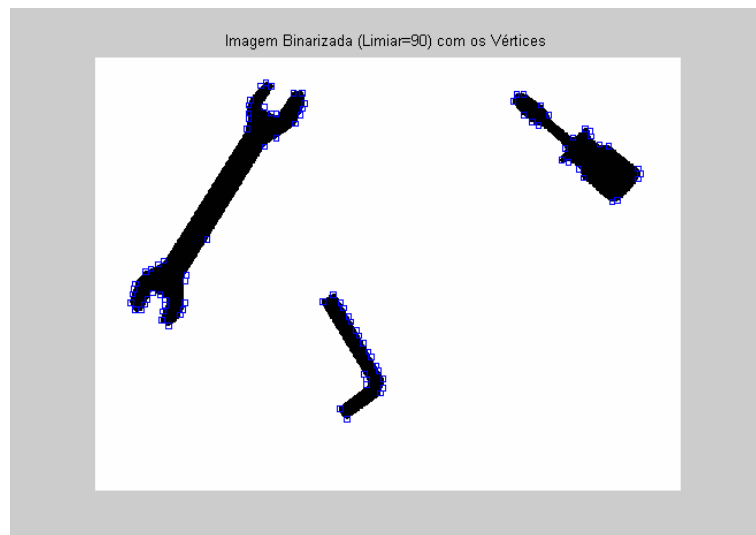


Figura 6.21 – Imagem binarizada e vértices extraídos.

- A **Figura 6.22** apresenta a imagem após a seleção dos vértices do objeto 1. A **Figura 6.23** mostra a geometria do objeto 1 e a Tabela 6.26 contém os pontos característicos (vértices) em pixel.

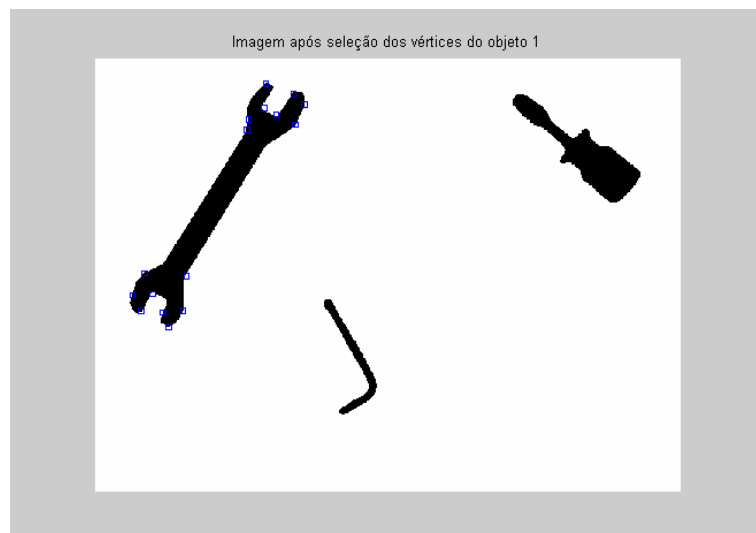


Figura 6.22 – Imagem dos Vértices selecionados do Objeto 1.

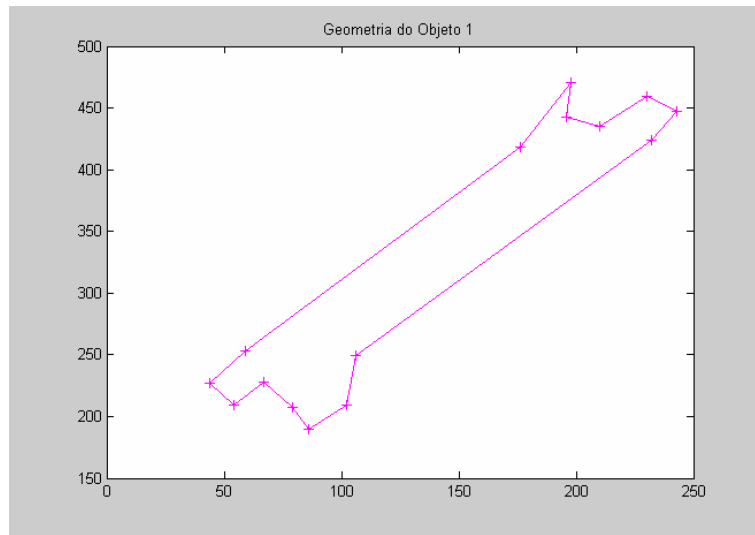


Figura 6.23 – Geometria do Objeto 1.

Tabela 6.26 – Lista dos vértices (pixel) selecionados do objeto 1.

Posições dos vértices após seleção e ordenação – Geometria do Objeto 1															
i	30	58	66	42	54	77	252	292	311	294	273	292	274	248	83
J	198	196	210	230	243	232	106	102	86	79	67	54	44	59	176

- A **Figura 6.24** apresenta a imagem após a seleção dos vértices do objeto 2. A **Figura 6.25** mostra a geometria do objeto 2 e a Tabela 6.27 contém os pontos característicos (vértices) em pixel.

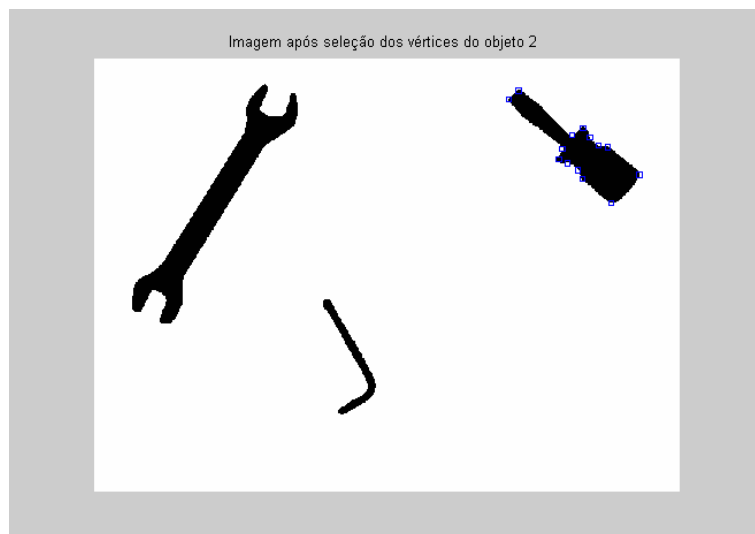


Figura 6.24 – Imagem dos Vértices selecionados do Objeto 2.

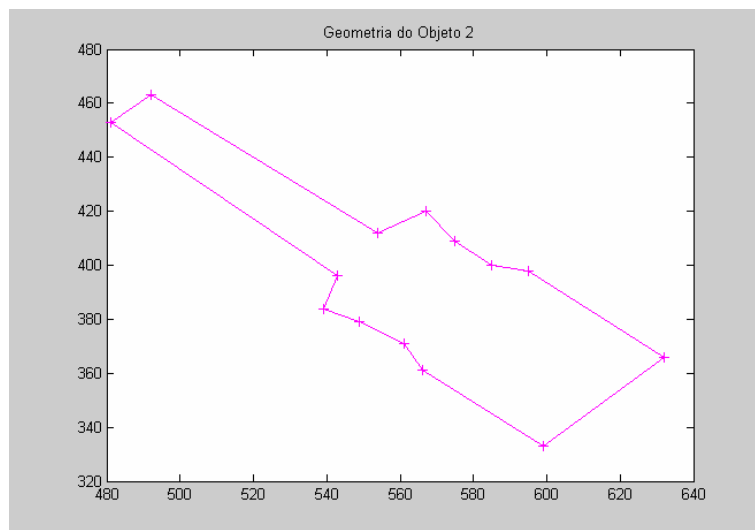


Figura 6.25 – Geometria do objeto 2.

Tabela 6.27 – Lista dos vértices (pixel) selecionados do objeto 2.

Posições dos vértices após seleção e ordenação – Geometria do Objeto 2														
i	38	89	81	92	101	103	135	168	140	130	122	117	105	48
j	492	554	567	575	585	595	632	599	566	561	549	539	543	481

• A Figura 6.26 apresenta a imagem após a seleção dos vértices do objeto 3. A Figura 6.27 mostra a geometria do objeto 3 e a Tabela 6.28 contém os pontos característicos (vértices) em pixel.

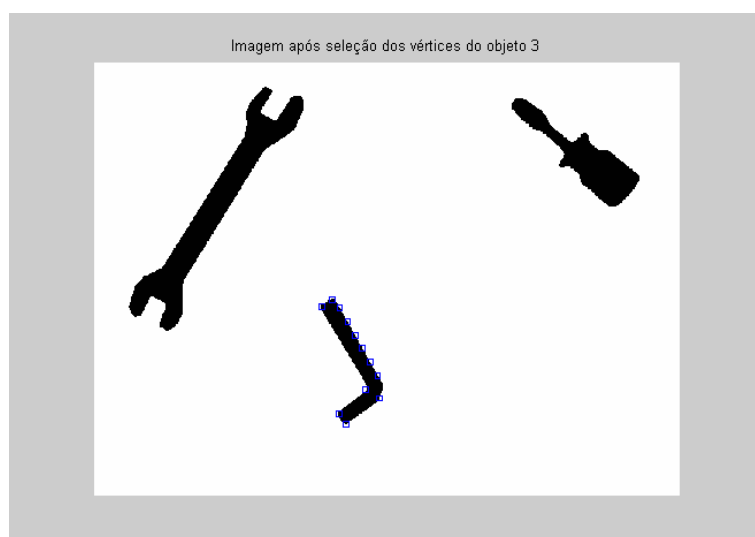


Figura 6.26 – Imagem dos Vértices selecionados do Objeto 3.

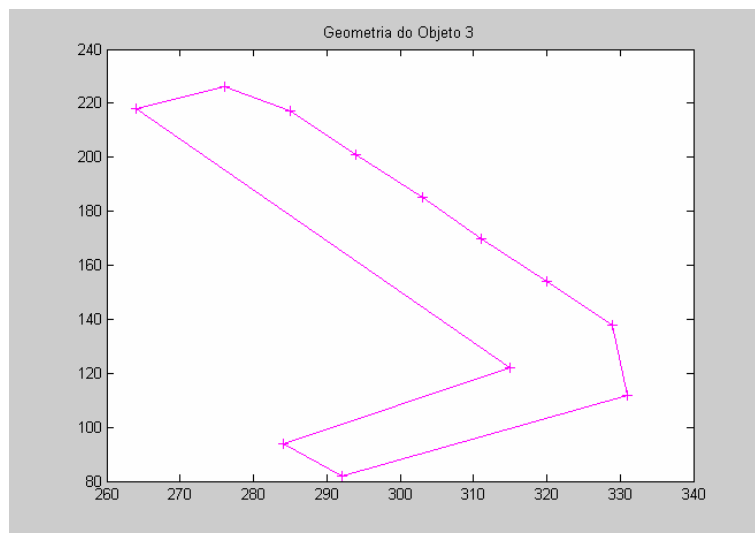


Figura 6.27 – Geometria do objeto 3.

Tabela 6.28 – Lista dos vértices (pixel) selecionados do objeto 3.

Posições dos vértices após seleção e ordenação – Geometria do Objeto 3												
i	300	316	331	347	363	389	419	407	379	283	275	284
j	294	303	311	320	329	331	292	284	315	264	276	285

- A **Figura 6.28** mostra o resultado da comparação entre o objeto 1 e os modelos 1, 2 e 3. O menor dos erros mínimos ($E_{\min} = 15.3029$) é que identifica o objeto.

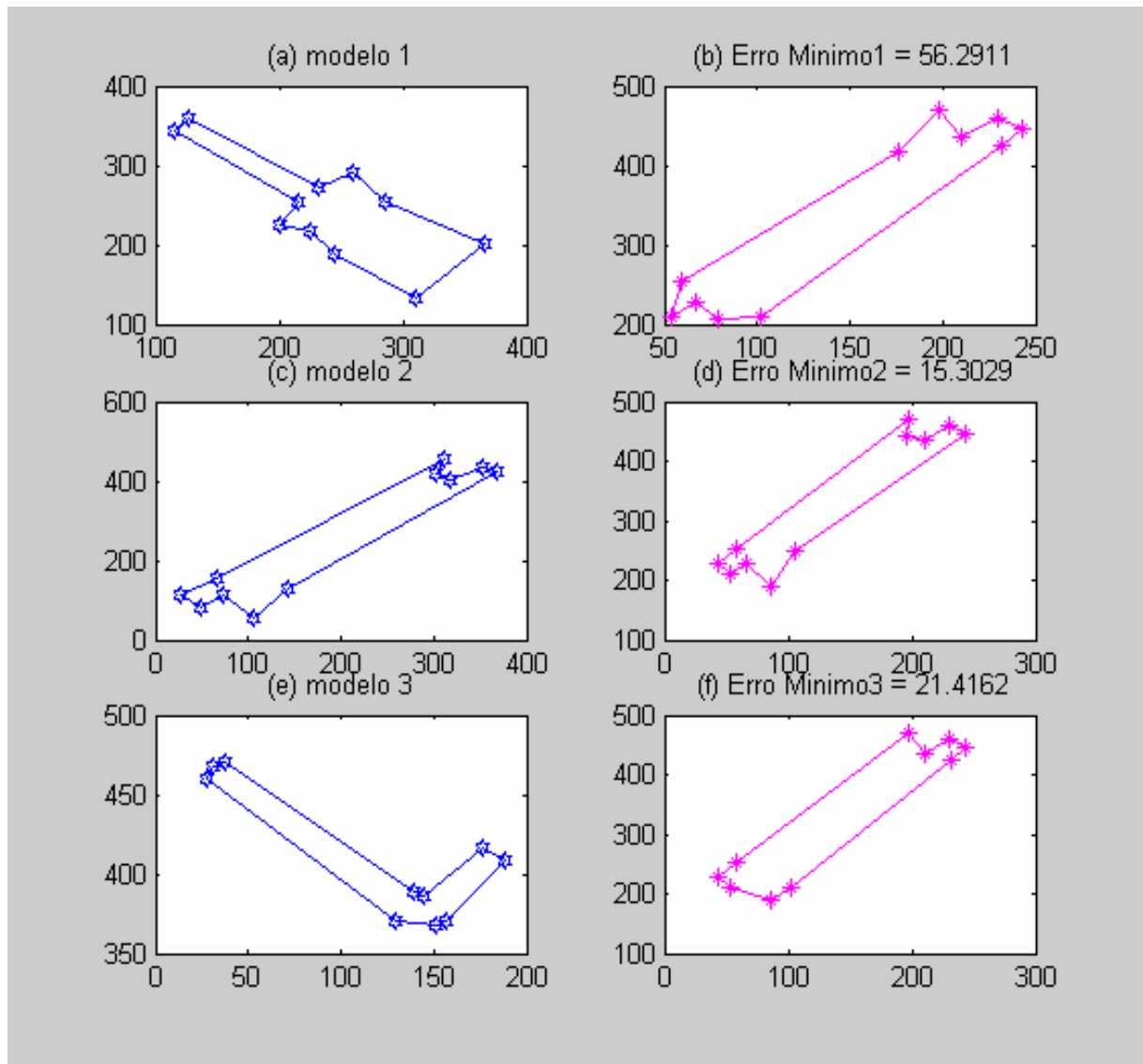


Figura 6.28 – Resultado da comparação do Objeto 1 com os modelos 1, 2 e 3.

- A **Figura 6.29** mostra o resultado da comparação entre o objeto 2 e os modelos 1, 2 e 3. O menor dos erros mínimos ($E_{\min} = 15.347$) é que identifica o objeto.

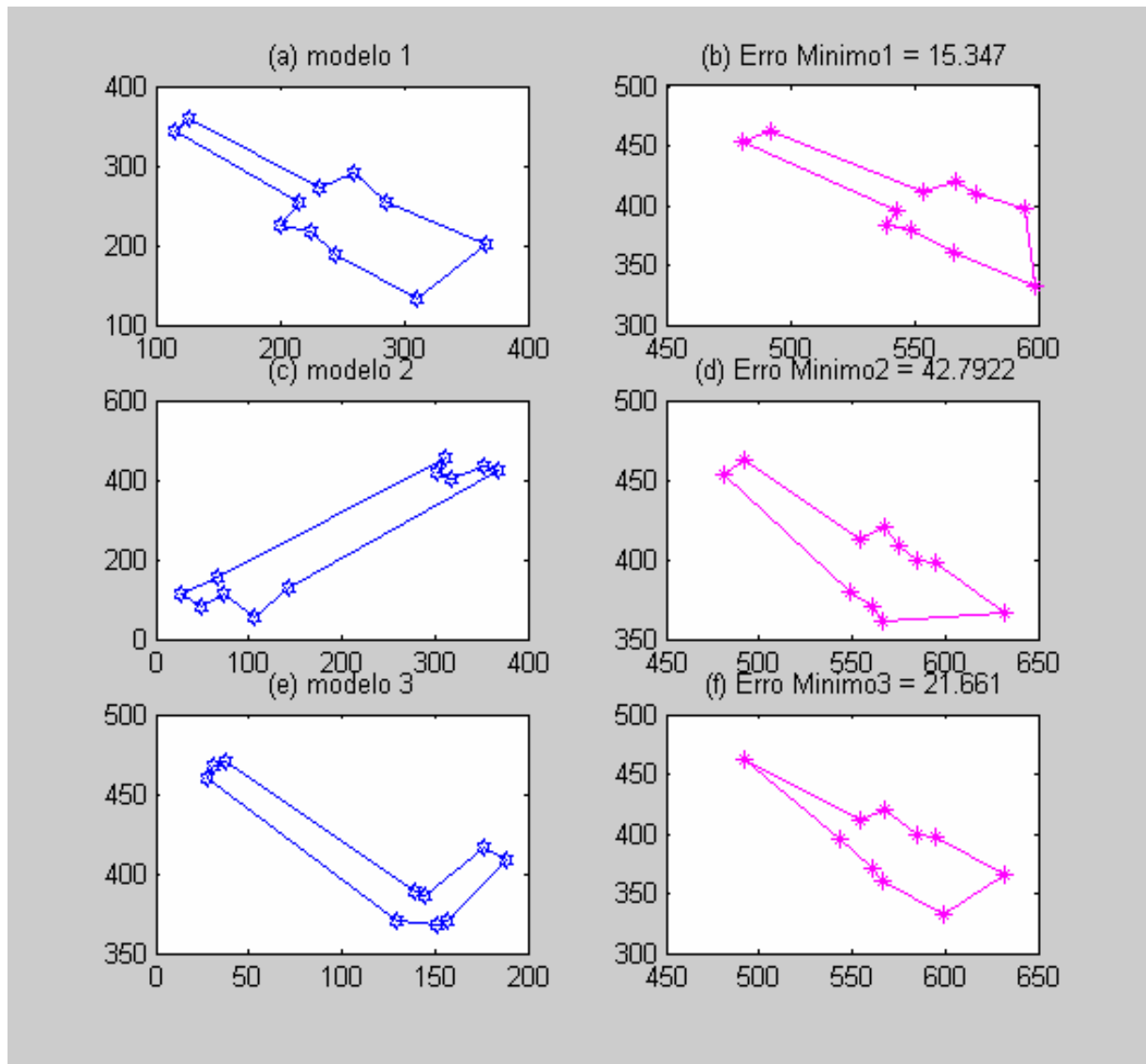


Figura 6.29 – Resultado da comparação do Objeto 2 com os modelos 1, 2 e 3.

- A **Figura 6.30** mostra o resultado da comparação entre o objeto 3 e os modelos 1, 2 e 3. O objeto com menor erro mínimo ($E_{\min} = 34.4528$) é que identifica o objeto.

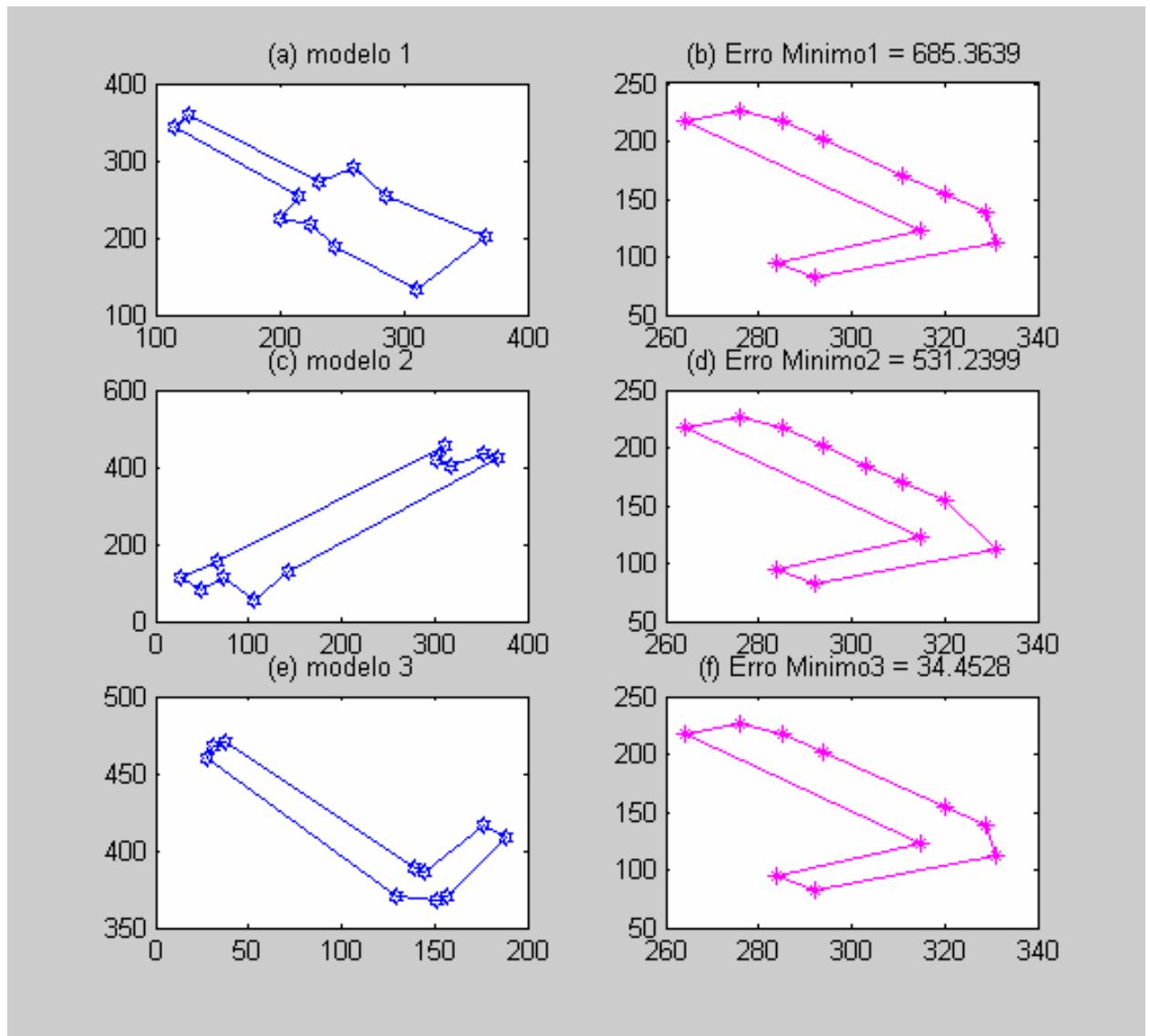


Figura 6.30 – Resultado da comparação do Objeto 3 com os modelos 1, 2 e 3.

Os experimentos apresentaram resultados diversos. A margem de erro é devido à diferença de iluminação das imagens, quando da geração dos modelos e na fase de identificação da mesma peça. Outro detalhe importante a analisar é o posicionamento do objeto, quando este não tem uma geometria simétrica.

7. CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS

O principal foco desta dissertação está em estabelecer movimentos de um manipulador, no sentido de se aproximar de objetos, através da utilização de um sistema de visão monocular, sem a necessidade de conhecer os parâmetros intrínsecos ou extrínsecos do sensor (câmera).

O sistema de visão é responsável pelo reconhecimento de objetos. O reconhecimento consiste em dois processos: *identificação e localização*.

O processo de identificação de objetos foi desenvolvido utilizando-se da técnica de comparação entre projetivas invariantes de 5 pontos coplanares. A criação da lista de vértices, ordenada no sentido horário e formando um polígono fechado, diminui consideravelmente a quantidade de combinações necessárias para a comparação entre objetos extraídos de uma imagem e os objetos armazenados como modelos, e conseqüentemente, reduzindo o tempo necessário de processamento.

O processo de localização utilizou-se da técnica de câmera de perspectiva fraca, onde não é necessário o conhecimento prévio dos parâmetros intrínsecos e extrínsecos da câmera. Nos experimentos relativos à localização dos objetos identificados em uma imagem, concluiu-se que basta conhecer 4 pontos no espaço 3-D em que se encontra o objeto, e seus respectivos pontos na imagem da câmera (pixel), para conhecer a posição do objeto.

Os experimentos foram realizados com objetivo de simular a identificação e localização de objetos através de imagens. Verificou-se a eficiência dos algoritmos com diversos tipos de imagens. Os algoritmos propostos foram desenvolvidos com poucas limitações.

O manipulador é então controlado pelas informações que obtém do sistema de visão e a restrição para o processo de pega do objeto é que este esteja num plano, pois não se utilizou os conceitos de estereoscopia.

Sugerem-se como trabalhos futuros, tornar a técnica de reconhecimento mais flexível, para tanto, utilizar-se-ia de técnicas por comparação baseada em invariantes 3-D. Seria possível também desenvolver o reconhecimento de partes de uma peça, facilitando o reconhecimento destas quando sobrepostas.

Sugere-se também que se desenvolva o processo manipulação da garra (elemento efetuator) por meio de redes neurais. Devem-se testar também outras técnicas extração de vértices para comparar a performance entre elas.

REFERÊNCIAS BIBLIOGRÁFICAS

Almeida, M.G.V.B. (2004), “Controlo de um Manipulador Robótico Usando Visão” Universidade Técnica de Lisboa. Dissertação para obter grau de mestre.

Bandlow, T., Hauk, A., Einsele, T. e Farber, G. (1998) “Recognising Objects by Their Silhouette”. *Imacs Conf. on Comp. Eng. In Systems Appl.* Pp 774-799.

Beaudet, P.R. (1978), “Rotational Invariant Image Operators”. *International Conference Pattern Recognition*. Pp 579-583.

Cai, L.D. (1990) “Scale-based surface understanding Using Diffusion Smoothing” PhD- Thesis- Department of Artificial Intelligence, University of Edinburgh.

Canny J. (1986) “A Computational Approach to Edge Detection”. *IEEE – Transactions on Pattern Analysis and Machine Intelligence*, Vol PAMI-8 pp 679-698.

Chin, R.T. e Dyer, C.R. (1986) “Model-based Recognition in Robot Vision”. *Computing Surveys*, vol 18 n° 1 pp 67-108.

Corke, P.I. (1996) “A Tutorial on Vision Servo Control”. *IEEE – Transactions on Robotics and Automation*, Vol. 12

Crane, C. e Duffy, J. (1998). “Kinematic Analysis of Robot Manipulation” Cambridge University Press.

Deriche, R. e Giraudon, G. (1993) “A computational Approach for Corner and Vertex Detection” – *International Journal of Computer Vision* 10:2, pp 101-124.

Gonzalez, R. C. e Woods, R. E. (2000) “Processamento de Imagens Digitais” – Editora Edgard Blucher LTDA.

Hartley, R. and Zisserman, A. (2000) “Multiple View Geometry in Computer Vision”, Cambridge University Press.

Horald, R., Conio, B. e Lebouleux, O. (1989) “ Na Analytic Solution for the Perspective 4-Point Problem”. Computer Vision Graphics and Image Process. Vol 47 pp 33-44.

Kitchen, L. e Rosenfeld, A. (1982) “Gray Level Corner Detection”. Pattern Recognition Letters, pp 95-102.

Latil, P. (1959). “ O Pensamento Artificial”. IBRASA.

Michel, C. (1852) “Traité de Géométrie Supérieure” – Bachelier, Paris – Electronic Edition Available at the Bibliothèque Nationale de France (www.bnf.fr).

Murray, R.M., Li, Z. e Satry, S.S. (1994) “ A Mathematical Introduction to Robotic Manipulation”, CRC Press.

Noble, J.A.(1988) “Finding Corners”. Image and Vision Computing, vol. 6 pp121-128.

Song, B.S., Yun, I.D. e Lee, S.U. (2000) “A Target Recognition Technique Employing Geometric Invariants”. Pattern Recognition, 33. pp 413-425.

Spong, M. W. e Vidyasagar, M. (1989) “Robot Dynamics and Control”. Editora John Wiley & Sons.

Trucco, E. e Verri, A. (1998). “Introductory Techniques for 3-D Computer Vision” Prentice Hall.

Vicent, M.A., Gil, P. e Torres, F. (2002). “Objects Recognition by Means of Projective Invariants Considering Corner-Points” The 10th. International Conf. in Central Europe on Computer Graphics Visualization and Computer Vision. Vol 10 N° 1-2.