

HEBERT LUCHETTI RIBEIRO

**Reconhecimento de Gestos Usando Segmentação de
Imagens Dinâmicas de Mão Baseada no Modelo de
Mistura de Gaussianas e Cor de Pele**

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo como requisito para obtenção do título de mestre em engenharia elétrica.

Área de Concentração: Visão Computacional
Orientador: Professor Doutor Adilson Gonzaga

São Carlos

2006

AGRADECIMENTOS

Ao Professor Dr. Adilson Gonzaga, pela orientação, discussões e atenção neste trabalho.

À Universidade de São Paulo por colocar a minha disposição sua estrutura.

À Libânio Carlos de Souza (Diretoria) e a Omar Socialoto (Gerência) da empresa SMAR Equipamentos Industriais Ltda, por compreenderem a importância de um mestrado, e pelos grandes conhecimentos de vida pessoal e profissional transmitidos, contribuindo para minha formação.

À empresa SMAR Equipamentos Industriais Ltda por liberar-me do trabalho da empresa para que comparecesse às aulas do mestrado, inclusive com ajuda de custo das viagens, e por demonstrarem um verdadeiro interesse na evolução dos seus profissionais para sempre estarem criando e inovando em novas tecnologias, incentivando-os a trabalharem com orgulho e com dedicação.

E a todos que de certa forma contribuíram para a conclusão desta monografia.

RESUMO

RIBEIRO, H. L. **Reconhecimento de gestos usando segmentação de imagens dinâmicas de mãos baseada no modelo de mistura de Gaussianas e cor de pele.** 2006. 144p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP, 2006.

O objetivo deste trabalho é criar uma metodologia capaz de reconhecer gestos de mãos, a partir de imagens dinâmicas, para interagir com sistemas. Após a captação da imagem, a segmentação ocorre nos pixels pertencentes às mãos que são separados do fundo pela segmentação pela subtração do fundo e filtragem de cor de pele. O algoritmo de reconhecimento é baseado somente em contornos, possibilitando velocidade para se trabalhar em tempo real. A maior área da imagem segmentada é considerada como região da mão. As regiões detectadas são analisadas para determinar a posição e a orientação da mão. A posição e outros atributos das mãos são rastreados quadro a quadro para distinguir um movimento da mão em relação ao fundo e de outros objetos em movimento, e para extraír a informação do movimento para o reconhecimento de gestos. Baseado na posição coletada, movimento e indícios de postura são calculados para reconhecimento um gesto significativo.

Palavras-chave: Reconhecimento de Gestos, Segmentação, Cor de Pele, Mistura de Gaussianas, Momentos Invariantes, Visão Computacional, Interação Humano-Computador (IHC), Gestos de Mão.

ABSTRACT

RIBEIRO, H. L. Gesture recognizing using segmentation of dynamic hand image based on the mixture of Gaussians model and skin color. 2006, 144p. Dissertation (Master Degree). Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP, 2006.

The purpose of this paper is to develop a methodology able to recognize hand gestures from dynamic images to interact with systems. After the image capture segmentation takes place where pixels belonging to the hands are separated from the background based on skin-color segmentation and background extraction. The image preprocessing can be applied before the edge detection. The recognition algorithm uses edges only; therefore it is quick enough for real time. The largest blob from the segmented image will be considered as the hand region. The detected regions are analyzed to determine position and orientation of the hand for each frame. The position and other attributes of the hands are tracked per frame to distinguish a movement from the hand in relation to the background and from other objects in movement, and to extract the information of the movement for the recognition of dynamic gestures. Based in the collected position, movement and indications of position are calculated to recognize a significant gesture.

Keywords: Gesture Recognition, Segmentation, Skin Color, Mixture of Gaussians, Invariants Moments, Computer Vision, Human Computer Interaction (HCI), Hand Gesture.

LISTA DE FIGURAS

FIGURA 1.1: ALGUNS GESTOS REPRESENTANDO SINAIS DO ALFABETO.....	3
FIGURA 1.2: EXEMPLO BÁSICO DOS EQUIPAMENTOS NECESSÁRIOS PARA IUC.....	5
FIGURA 1.3: EXEMPLO DE UM GESTO DE MÃO SENDO CAPTURADO POR UM IUC.....	5
FIGURA 2.1: DEXTEROUS HANDMASTER	12
FIGURA 2.2: POWER GLOVE	12
FIGURA 2.3: CYBERGLOVE	12
FIGURA 2.4: VPL DATAGLOVE.....	12
FIGURA 2.5: DIAGRAMA DE BLOCOS DE UM SISTEMA DE VISÃO COMPUTACIONAL	13
FIGURA 3.1: REPRESENTAÇÃO DO ESPAÇO DE CORES DO MODELO RGB. A LINHA PONTILHADA (DIAGONAL DO CUBO) DEFINE OS NÍVEIS DE CINZA.....	22
FIGURA 3.2:REPRESENTAÇÃO DO ESPAÇO DE CORES DO MODELO HSV	24
FIGURA 3.3: ESTRUTURAS PIRAMIDAS DUPLO-HEXACÔNICA (A) E DUPLO-TETRAÉDRICA (B) DO ESPAÇO DE ATRIBUTO DE CORES - HSL E HSI, RESPECTIVAMENTE. FONTE: ADAPTADAS DE FOLEY, ET AL. (1990, p. 594) E GONZALEZ E WOODS (2000, p. 230).....	25
FIGURA 3.4: PROJEÇÃO DO EIXO RGB EM R+B+G=0	26
FIGURA 3.5: ESPAÇO DE CORES YCbCr.....	27
FIGURA 4.1: EXEMPLO DE AGRUPAMENTOS DE DENSIDADES CRIADAS USANDO-SE GMM.....	42
FIGURA 4.2: AGRUPAMENTO DE PIXELS DE COR DE PELE EM RGB.....	43
FIGURA 5.1: TRÊS EXEMPLOS DE IMAGENS SEGMENTADAS COM FALHAS E BURACOS (A,B,C) E DEPOIS AO PÓS-PROCESSAMENTO (D,E,F RESPECTIVAMENTE).	48
FIGURA 5.2: EM (A) É MOSTRADA A IMAGEM SEGMENTADA PÓS-PROCESSADA. EM (B) TODOS OS CONTORNOS GERADOS PELO DETECTOR DE CONTORNOS. EM (C) O MAIOR CONTORNO SOMENTE.	49
FIGURA 5.3: EM (A) A IMAGEM SEGMENTADA SEM PÓS-PROCESSAMENTO. EM (B) A IMAGEM COM CONTORNOS DA IMAGEM SEGMENTADA SEM PÓS-PROCESSAMENTO.....	49
FIGURA 5.4: (A) IMAGEM DA MÃO SEGMENTADA. (B) IMAGEM GERADA PELA APLICAÇÃO DA TRANSFORMADA DA DISTÂNCIA NA IMAGEM SEGMENTADA.	50
FIGURA 5.5: POSIÇÃO DA MÃO DETERMINADA PELO CENTRO DA PALMA E CALCULADO COM TDE	51
FIGURA 5.6: (A) INFLUÊNCIA DO ANTEBRAÇO NO CÁLCULO DA ORIENTAÇÃO DOS SEMI-EIXOS DA ELIPSE. (B) CONTORNO MOSTRANDO QUE O PULSO FOI REMOVIDO. (C) NOVA ORIENTAÇÃO DOS SEMI-EIXOS DA ELIPSE APÓS A REMOÇÃO DO PULSO.....	52
FIGURA 5.7: (A) ORIENTAÇÃO E COMPRIMENTO DOS SEMI-EIXOS DA ELIPSE CALCULADA PELOS MOMENTOS DO CONTORNO. (B) CONTORNO MOSTRANDO QUE O PULSO FOI REMOVIDO. (C) ORIENTAÇÃO E COMPRIMENTO DOS SEMI-EIXOS DA ELIPSE MODIFICADOS APÓS A REMOÇÃO DO PULSO.	52

FIGURA 5.8: EM (A) CONTORNO SEM A REMOÇÃO DO ANTEBRAÇO E COM AS CIRCUNFERÊNCIAS DA PALMA(AZUL) E DE <i>OFFSET</i> (VERMELHO). EM (B) CONTORNO MOSTRANDO QUE O PULSO FOI REMOVIDO EXATAMENTE NAS INTERSECÇÕES COM CIRCUNFERÊNCIA <i>OFFSET</i>	53
FIGURA 6.1: IMAGENS CONTENDO OS 10 CONTORNOS DE GESTOS DE MÃO ARMAZENADOS PARA CASAMENTO DE PADRÕES.....	60
FIGURA 6.2: ELIPSE EQUIVALENTE QUE DESCREVE A POSIÇÃO E A ORIENTAÇÃO DE UM CONTORNO, MOSTRANDO O SEMI-EIXO MAIOR L , O SEMI-EIXO MENOR W E A ORIENTAÇÃO θ	63
FIGURA 6.3: ELIPSE DA IMAGEM MOSTRANDO O SEMI-EIXO MAIOR L , O SEMI-EIXO MENOR W E A ORIENTAÇÃO θ	63
FIGURA 6.4: A) FORMA DO OBJETO. B) REPRESENTAÇÃO GRÁFICA DA CIRCULARIDADE DO OBJETO.....	64
FIGURA 6.5: A) FORMA DO OBJETO. B) REPRESENTAÇÃO GRÁFICA DOS RAIOS MÁXIMO E MÍNIMO DO OBJETO	65
FIGURA 7.1: (A) IMAGEM DE VÍDEO ORIGINAL. (B) SEGMENTAÇÃO DA IMAGEM ORIGINAL USANDO GMM E FILTRO DE COR DE PELE. (C) PÓS-PROCESSAMENTO. (D) EXTRAÇÃO DO CONTORNO DA MÃO. (E) REMOÇÃO DO PUNHO DA MÃO.	71
FIGURA 7.2: REPRESENTAÇÃO DO CASAMENTO DE PADRÕES ENTRE OS 10 CONTORNOS DE CLASSES DE GESTOS DE MÃO ARMAZENADOS E O CONTORNO OBTIDO, DETECTANDO-SE A CLASSE PARAR.	71
FIGURA 7.3: EM (A) TEM-SE A IMAGEM ORIGINAL, EM (B) A IMAGEM SEGMENTADA POR COR DE PELE, E EM (C) A IMAGEM COM CONTORNO RESULTANTE DA IMAGEM SEGMENTADA.....	75
FIGURA 7.4: EXEMPLO DE UMA IMAGEM SEGMENTADA COM FALHAS E BURACOS (A) E DEPOIS DO PÓS- PROCESSAMENTO (B).....	76
FIGURA 7.5: (A) CONTORNO DA MÃO. (B) CONTORNO COM REMOÇÃO DO PUNHO DA MÃO.....	77
FIGURA 7.6: Os 10 CONTORNOS DE GESTOS DE MÃO USADOS PARA CASAMENTO DE PADRÕES.	80
FIGURA 7.7: CONTORNO DE ENTRADA PARA RECONHECIMENTO.....	82
FIGURA 7.8: TESTES DE UM VÍDEO MOSTRANDO IMAGENS ORIGINAIS NA COLUNA (A). NAS COLUNAS (B,D,F) MOSTRAM IMAGENS SEGMENTADAS POR GMM SEM LIMIAR DE COR DE PELE. NAS COLUNAS (C,E,G) SÃO MOSTRADOS OS CONTORNOS. EM (B,C) SÃO RESULTADOS BASEADOS NO MODELO DE STAUFFER E GRIMSON (1999). EM (D,E) BASEADO EM POWER E SCHOONES (2002) E EM (F,G) BASEADO EM KADEWTRAkUPONG E BOWDEN (2001).....	86
FIGURA 7.9: IMAGENS COM LIMIAR DE COR DE PELE, BASEADO NO MODELO DE STAUFFER E GRIMSON (1999) (B) E SUAS SIMPLIFICAÇÕES REALIZADAS POR POWER E SCHOONES (2002) (C) E KADEWTRAkUPONG E BOWDEN (2001) (D).....	87
FIGURA 7.10: QUADROS COM SUBTRAÇÃO DE FUNDO POR GMM COM PARÂMETROS IGUAIS PARA AS TRÊS ABORDAGENS. EM (A) SÃO MOSTRADOS QUADROS OBTIDOS USANDO STAUFFER E GRIMSON (1999), EM (B) USANDO POWER E SCHOONES (2002) E EM (C) USANDO KADEWTRAkUPONG E BOWDEN (2001).....	88
FIGURA 7.11: TAXA DE QUADROS/SEGUNDO DAS ABORDAGENS DE GMM SEM PÓS-PROCESSAMENTO.	91
FIGURA 7.12: TAXA DE NÚMERO DE QUADROS POR SEGUNDO PARA O ALGORITMO GMM EM SUAS TRÊS ABORDAGENS COM PÓS-PROCESSAMENTO.....	91

FIGURA 7.13: : TAXA DE NÚMERO DE QUADROS POR SEGUNDO PARA O ALGORITMO GMM EM SUAS TRÊS ABORDAGENS COM PÓS-PROCESSAMENTO.....	92
FIGURA A.1: FIGURA COM A TELA INICIAL DO PROGRAMA REGEM.....	109
FIGURA A.2: PROPOSTAS DO ALGORITMO GMM.....	109
FIGURA A.3: MENU DE PARÂMETROS INICIAIS COM VALORES DEFAULT.....	109
FIGURA A.4: MENU DE SELEÇÃO DO MODO DE ENTRADA DE SEQÜÊNCIA DE IMAGENS. SÃO BOTÕES PARA CARREGAR UM ARQUIVO DE VÍDEO OU CAPTAR O SINAL DE UMA WEBCAM.....	110
FIGURA A.5: OPÇÕES DE PÓS-PROCESSAMENTO, REMOÇÃO DE PUNHO E CASAMENTO DE PADRÕES.....	110
FIGURA A.6: OPÇÕES DE JANELAS EXIBIDAS DURANTE O PROCESSO.....	110
FIGURA A.7: JANELAS POSSÍVEIS DE SEREM EXIBIDAS PELO PROGRAMA MOSTRANDO A IMAGEM ORIGINAL, SUBTRAÇÃO DE FUNDO USANDO GMM E FILTRO DE COR DE PELE E CONTORNO DA MÃO.....	111
FIGURA A.8: OPÇÕES PARA EXPORTAR OS RESULTADOS DO PROCESSAMENTO.....	112
FIGURA A.9: OPÇÕES DE FIGURAS DE MOSTRAM OS RESULTADOS DE ETAPAS DE PROCESSAMENTO.....	112
FIGURA A.10: CÍRCULO DA PALMA DA MÃO EM AZUL. CÍRCULO OFFSET DA PALMA EM VERMELHO.....	113
FIGURA A.11: ELIPSE E SEMI-EIXOS DA MÃO EM VERDE CALCULADOS POR MOMENTOS DA IMAGEM.....	113
FIGURA A.12: O CONTORNO SEM O PUNHO EM VERDE.....	113
FIGURA A.13: RETÂNGULO EM AZUL QUE REPRESENTA A REGIÃO DE MAIOR <i>BLOB</i> DA IMAGEM.....	113
FIGURA A.14: CONTORNO ENCONTRADO NA IMAGEM À ESQUERDA E À DIREITA O CONTORNO PADRÃO UM SEGURAR DETECTADO CORRETAMENTE.....	114
FIGURA A.15: CONTORNO ENCONTRADO NA IMAGEM À ESQUERDA E À DIREITA O CONTORNO PADRÃO UM DETECTADO ERRONEAMENTE.....	114

LISTA DE TABELAS

TABELA 6.1: EXEMPLO DE VETORES DE CARACTERÍSTICAS.....	67
TABELA 7.1: PARÂMETROS USADOS NO ALGORITMO GMM NAS DIFERENTES ABORDAGENS.....	74
TABELA 7.2: DESCRIÇÕES DOS PARÂMETROS DE GMM.....	74
TABELA 7.3: TABELA MOSTRANDO 5 CLASSES (CANCELAR, DIREITA, DOIS, ESQUERDA E INICIAR) DE VETORES DE CARACTERÍSTICAS PADRÃO E OS VALORES DE SEUS RESPECTIVOS PARÂMETROS.....	81
TABELA 7.4: TABELA MOSTRANDO 5 CLASSES (MOVER, PARAR, QUATRO, SEPARAR E UM) DE VETORES DE CARACTERÍSTICAS PADRÃO E OS VALORES DE SEUS RESPECTIVOS PARÂMETROS.....	81
TABELA 7.5: TABELA EXEMPLO DE UM CONTORNO DE ENTRADA COM SEUS VALORES DO VETOR DE CARACTERÍSTICAS E OS MESMOS NORMALIZADOS.....	82
TABELA 7.6: TABELA COM DIFERENÇAS EUCLIDIANAS ENTRE O VETOR DE ENTRADA E 10 DOS VETORES PADRÃO. A MENOR DISTÂNCIA ENCONTRADA FOI A DA CLASSE MOVER.....	83
TABELA 7.7: TAXAS DE QUADROS/SEGUNDO USANDO OS MÉTODOS GMM. OS VALORES MOSTRADOS SÃO COM PÓS-PROCESSAMENTO, PÓS-PROCESSAMENTO E DETECÇÃO DE PELE E SEM PÓS-PROCESSAMENTO.....	89
TABELA 7.8: VALORES DE PARÂMETROS INICIAIS USADOS PARA TODOS AS ABORDAGENS DE GMM.....	89
TABELA 7.9: TABELA COM VALORES TOTAIS DOS 10 TESTES REALIZADOS COM OS 100 VÍDEOS DE GESTOS.	93
TABELA 7.10: RESULTADO DOS TESTES FEITOS PARA A AVALIAÇÃO DO RECONHECIMENTO DOS GESTOS PREDEFINIDOS. PARA CADA UM DOS 10 GESTOS É APRESENTADA UMA ESTATÍSTICA INDICANDO O NÚMERO E A PORCENTAGEM DE ACERTOS E ERROS AO LONGO DO TESTE.....	93
TABELA 7.11: TABELA DE MÉDIAS DE TODOS OS TESTES REALIZADOS E CONTABILIZADO POR TODOS OS GESTOS.....	93
TABELA B.1: DADOS GERADOS DO BLOCO 1 DE TESTES.....	115
TABELA B.2: DADOS GERADOS DO BLOCO 2 DE TESTES.....	115
TABELA B.3: DADOS GERADOS DO BLOCO 3 DE TESTES.....	116
TABELA B.4: DADOS GERADOS DO BLOCO 4 DE TESTES.....	116
TABELA B.5: DADOS GERADOS DO BLOCO 5 DE TESTES.....	116
TABELA B.6: DADOS GERADOS DO BLOCO 6 DE TESTES.....	117
TABELA B.7: DADOS GERADOS DO BLOCO 7 DE TESTES.....	117
TABELA B.8: DADOS GERADOS DO BLOCO 8 DE TESTES.....	117
TABELA B.9: DADOS GERADOS DO BLOCO 9 DE TESTES.....	118
TABELA B.10: DADOS GERADOS DO BLOCO 10 DE TESTES.....	118
TABELA B.11: DADOS DE VALORES TOTALIZADOS GERADOS A PARTIR DE TODOS OS 10 BLOCOS DE TESTES.....	118
TABELA B.12: VALORES DE ACERTOS E ERROS REALIZADOS NOS TESTES DO BLOCO 1.....	119
TABELA B.13: VALORES DE ACERTOS E ERROS REALIZADOS NOS TESTES DO BLOCO 2.....	119
TABELA B.14: VALORES DE ACERTOS E ERROS REALIZADOS NOS TESTES DO BLOCO 3.....	120

TABELA B.15: VALORES DE ACERTOS E ERROS REALIZADOS NOS TESTES DO BLOCO 4.....	120
TABELA B.16: VALORES DE ACERTOS E ERROS REALIZADOS NOS TESTES DO BLOCO 5.....	120
TABELA B.17: VALORES DE ACERTOS E ERROS REALIZADOS NOS TESTES DO BLOCO 6.....	121
TABELA B.18: VALORES DE ACERTOS E ERROS REALIZADOS NOS TESTES DO BLOCO 7.....	121
TABELA B.19: VALORES DE ACERTOS E ERROS REALIZADOS NOS TESTES DO BLOCO 8.....	121
TABELA B.20: VALORES DE ACERTOS E ERROS REALIZADOS NOS TESTES DO BLOCO 6.....	122
TABELA B.21: VALORES DE ACERTOS E ERROS REALIZADOS NOS TESTES DO BLOCO 7.....	122
TABELA B.22: VALORES DE ACERTOS E ERROS REALIZADOS EM TODOS TESTES DOS 10 BLOCOS.....	122

SUMÁRIO

AGRADECIMENTOS	v
RESUMO	ix
ABSTRACT	xi
LISTA DE FIGURAS	xiii
LISTA DE TABELAS.....	xvii
1 INTRODUÇÃO	1
1.1 INTERAÇÃO USUÁRIO COMPUTADOR	1
1.2 GESTOS	1
1.3 PROPRIEDADES DOS GESTOS	2
1.4 VISÃO COMPUTACIONAL	3
1.5 SEGMENTAÇÃO DA IMAGEM	4
1.6 OBJETIVOS	5
1.7 ORGANIZAÇÃO DA MONOGRAFIA	7
2 INTERFACES BASEADAS EM VISÃO	9
2.1 INTRODUÇÃO	9
2.2 CONCEITOS BÁSICOS	9
2.3 REQUISITOS FUNCIONAIS	10
2.4 REQUISITOS NÃO-FUNCIONAIS	11
2.5 VISÃO COMPUTACIONAL	11
2.6 AQUISIÇÃO DE DADOS	13
2.7 PRÉ-PROCESSAMENTO.....	14
2.8 SEGMENTAÇÃO	14
2.9 EXTRAÇÃO DE CARACTERÍSTICAS	15
2.9.1 <i>Abordagem Baseada na Aparência</i>	16
2.9.2 <i>Abordagem Baseada nas Características</i>	17
2.10 CLASSIFICADOR	18
2.11 INTERFACES COMANDADAS POR GESTOS.....	19
3 ESPAÇOS DE CORES.....	21
3.1 INTRODUÇÃO	21
3.2 ESPAÇOS DE CORES	21
3.2.1 <i>Espaço RGB</i>	22
3.2.2 <i>Espaço RGB Normalizado</i>	23
3.2.3 <i>Espaços HSV, HSL e HSI</i>	23

3.2.4	<i>Espaço YCrCb</i>	26
3.2.5	<i>Espaços YIQ e YUV</i>	27
3.3	DETECÇÃO DE COR DE PELE.....	28
3.3.1	<i>Trabalhos Relacionados</i>	28
3.3.2	<i>Faixas de Agrupamentos de Pele em Espaço de Cores</i>	30
4	SEGMENTAÇÃO USANDO GMM.....	33
4.1	INTRODUÇÃO.....	33
4.2	SEGMENTAÇÃO.....	33
4.3	TRABALHOS RELACIONADOS	35
4.4	MODELO DE CENA DE FUNDO.....	37
4.5	MODELO DE MISTURAS DE GAUSSIANAS – GMM	38
4.6	LIMIAR DE COR DE PELE.....	42
5	PÓS-PROCESSAMENTO	45
5.1	INTRODUÇÃO	45
5.2	IMPORTÂNCIA DO CONTORNO	45
5.3	PÓS-PROCESSAMENTO	46
5.4	DETECÇÃO DO CONTORNO	48
5.5	POSIÇÃO E ORIENTAÇÃO DA MÃO	49
5.6	TRANSFORMADA DA DISTÂNCIA	50
5.7	INFLUÊNCIA DO ANTEBRAÇO NA POSIÇÃO DA MÃO	51
5.8	REMOÇÃO DO ANTEBRAÇO	52
6	RECONHECIMENTO DE GESTOS.....	55
6.1	INTRODUÇÃO.....	55
6.2	CASAMENTO DE PADRÕES	55
6.3	MOMENTOS DE IMAGEM.....	56
6.4	GESTOS ESTÁTICOS E DINÂMICOS	59
6.5	GESTOS USADOS COMO PADRÃO	59
6.6	EXTRAÇÃO DE CARACTERÍSTICAS	60
6.6.1	<i>Momentos Geométricos ou Invariante s</i>	61
6.6.2	<i>Orientação</i>	62
6.6.3	<i>Comprimentos dos Semi-Eixos da Elipse</i>	63
6.6.4	<i>Circularidade</i>	64
6.6.5	<i>Excentricidade</i>	64
6.6.6	<i>Raio de Giro</i>	65
6.6.7	<i>Dispersão</i>	65
6.6.8	<i>Maior e Menor Momentos de Inércia</i>	66
6.7	CLASSIFICAÇÃO.....	66

6.8	ALGORITMO DE CLASSIFICAÇÃO.....	67
7	METODOLOGIA, RESULTADOS E CONCLUSÕES.....	69
7.1	INTRODUÇÃO	69
7.2	METODOLOGIA	70
7.2.1	<i>Aquisição da Seqüência de Imagens</i>	72
7.2.2	<i>Segmentação da Imagem.....</i>	72
7.2.2.1	Passos do Algoritmo GMM	72
7.2.2.2	Limiar de Cor de Pele	74
7.2.3	<i>Pós-processamento.....</i>	75
7.2.3.1	Extração do Contorno	76
7.2.3.2	Remoção do Antebraço.....	76
7.2.4	<i>Reconhecimento</i>	78
7.2.4.1	Determinação de Gestos de Mão	78
7.2.4.2	Vetor de Características.....	78
7.2.4.3	Classificação	79
7.2.4.4	Exemplo de Classificação	80
7.2.4.5	Definição dos Contornos Padrão.....	83
7.2.5	<i>Material.....</i>	84
7.3	RESULTADOS E DISCUSSÕES	85
7.3.1	<i>Escolha do Algoritmo GMM</i>	85
7.3.2	<i>Resultados da Classificação.....</i>	92
7.4	CONCLUSÕES E TRABALHOS FUTUROS	94
REFERÊNCIAS BIBLIOGRÁFICAS		99
APÊNDICE A – PROGRAMA REGEM.....		107
A.1	RECONHECIMENTO DE GESTOS DE MÃOS – REGEM.....	107
A.2	INTERFACE GRÁFICA	107
A.2.1	<i>Opções de Configuração</i>	107
A.2.2	<i>Opção GMM.....</i>	108
A.2.2.1	Algoritmo de Segmentação usando GMM.....	108
A.2.2.2	Parâmetros Iniciais Para GMM.....	108
A.2.2.3	Seqüência de Imagens.....	109
A.2.3	<i>Opção Pós-processamento</i>	110
A.2.4	<i>Opção de Janelas</i>	111
A.2.5	<i>Opção Exportação.....</i>	112
A.2.6	<i>Opção Figuras.....</i>	113
APÊNDICE B – TABELAS DE CLASSIFICAÇÕES DE GESTOS		115

1 INTRODUÇÃO

1.1 Interação Usuário Computador

Em nossa vida diária atuamos com outras pessoas e objetos para realizar uma variedade de ações que são importantes para nós. Computadores possuem uma crescente influência em muitos aspectos da nossa vida, por exemplo, a forma de comunicação, a forma de realizar nossas ações, e a forma como atuamos com nosso ambiente. Deste princípio, é usado o conceito de Interação Usuário-Computador (IUC). Embora o computador tenha avançado formidavelmente, a IUC ainda se baseia sobre simples dispositivos mecânicos (como teclado, mouse, joysticks) os quais reduzem a efetividade e naturalidade de tal interação.

Recentemente houve um interesse crescente em introduzir outros meios de interação usuário-computador para o campo da IUC. Este novo meio inclui uma classe de dispositivos baseados no movimento espacial da mão humana: gestos de mão.

1.2 Gestos

Na literatura não existe uma definição do termo “gesto” que seja unanimemente aceito. A sua definição é muito dependente da área em que é abordado e do objetivo do trabalho que é realizado. Mas podemos definir um gesto como sendo o movimento do corpo, especialmente dos braços e mãos, com a finalidade de exprimir idéias, sinais ou mímica. Eles variam desde simples ações, como apontar objetos, até movimentos mais complexos que expressam sentimento ou permitem a comunicação com outras pessoas.

Um gesto de mão é classificado como estático ou dinâmico. Um gesto estático (ou postura) é uma configuração particular da mão e sua pose, representado por uma simples imagem. Um gesto dinâmico é um gesto em movimento, representado por uma seqüência de imagens. Para explorar o uso de gestos em IUC é necessário prover os meios pelos quais eles podem ser interpretados por computadores. A interpretação de gestos em IUC requer que configurações dinâmicas e/ou estáticas da mão, braço ou

corpo sejam medidos pelo computador. Uma primeira abordagem trata este problema fazendo uso de dispositivos mecânicos que medem a mão e/ou ângulo da união do braço e a posição espacial. Esta abordagem não satisfaz o requerimento de naturalidade que requer uma IUC. Esta limitação pode ser resolvida com uma abordagem baseada em visão computacional.

1.3 Propriedades dos Gestos

Um gesto é motivado por uma intenção de se realizar uma certa tarefa: indicação, rejeição, agarrando, puxando uma flor, ou simplesmente arranhando com a mão. Desde a intenção inicial até a realização final, os gestos seguem um padrão característico no espaço e tempo. Kendon (1986) distingue três fases do movimento que compreende um gesto simples: preparação, curso e retração. Propriedades deste padrão são universais e permanentes, e podem ser usadas para descrever qualquer gesto particular. Quek (1994) desenvolveu o seguinte conjunto de regras para a detecção de gestos baseado no padrão de gesto anterior:

- Os gestos estão contidos em movimentos.
- Os gestos começam com um movimento inicial lento desde a posição de descanso (imóvel), continua com uma fase com velocidade crescente (segue o curso), e conclui retornando para posição de descanso.
- A mão assume uma particular configuração durante o curso.
- Movimentos lentos entre posições de descanso não são gestos.
- Gestos de mão deverão estar restringidos dentro de um certo volume - espaço de trabalho.
- Gestos de mão estáticos requerem um período finito de tempo para serem reconhecidos.
- Movimentos repetitivos podem ser gestos.

Ele também sugere que, com a exceção de alguns sinais na quirologia (linguagem de sinais de mãos) (figura 1.1), o movimento de dedos individuais pode representar um gesto só quando a mão estiver imóvel. Isso permite que gestos de mão possam ser modelados:

- **Gestos Estáticos da Mão:** são caracterizados pela postura da mão determinada por uma particular configuração do dedo-polegar-palma.
- **Gestos Dinâmicos da Mão:** são caracterizados por uma configuração inicial e final do curso da mão e pelo movimento geral para seguir o curso. A configuração da mão pode mudar profundamente o movimento, mas a mudança não contém informação do gesto e assim pode ser desconsiderada.

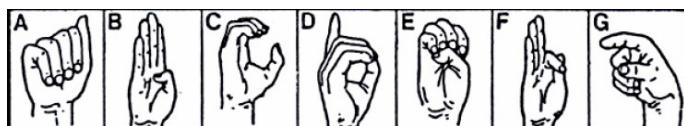


Figura 1.1: Alguns gestos representando sinais do alfabeto.

1.4 Visão Computacional

O trabalho apresentado nesta dissertação trata de um problema fundamental e muito amplo da visão computacional que é o reconhecimento de gestos por computador. Dentro deste domínio existem outros problemas que vão desde a aquisição de imagens, segmentação, descrição e reconhecimento de gestos, até a interpretação da imagem.

A visão computacional tem como objetivo final usar o computador para emular resultados próximos a visão humana, incluindo a fase do aprendizado sendo capaz de fazer inferências e tomar ações baseadas nas entradas visuais. A visão computacional pode ser dividida em três tipos de processos computacionais: nível baixo, intermediário e alto (GONZALEZ; WOODS, 2002).

- **Nível Baixo:** Este processo comprehende aquisição e pré-processamento de imagens. A aquisição permite representar as imagens de maneira interna no computador, através de uma matriz de valores, chamados pixels: cada pixel codifica a intensidade das cores (imagens coloridas) ou a intensidade luminosa (imagens em níveis de cinza). Esta informação é o resultado de um processo óptico e eletrônico cujo objetivo é transformar o sinal óptico, captado pelo ponto de vista da câmera ou do sistema óptico do scanner, numa imagem digitalizada, registrado nessa matriz de valores (imagem). O pré-processamento de imagens pode consistir, por exemplo, de redução de ruído ou borramento, realce do contraste e aumento da nitidez da imagem.

- **Nível Intermediário:** Este processo compreende a segmentação, representação, descrição e reconhecimento.
 - Segmentação consiste em separar na imagem regiões ou objetos.
 - Representação consiste na forma como representamos os objetos contidos na imagem. Neste trabalho, representamos os objetos através do seu contorno, reduzindo consideravelmente a informação presente na imagem.
 - Descrição consiste no cálculo das características dos objetos observados.
 - Reconhecimento é o processo que atribui um rótulo a um objeto presente na imagem, baseado na informação fornecida pelas suas características. O processo de nível intermediário é caracterizado pelas entradas geralmente serem imagens e as saídas atributos extraídos das imagens, como por exemplo, o contorno e a identidade de cada objeto.
- **Nível Alto:** Este processo compreende a interpretação da imagem. A interpretação consiste em atribuir um significado a uma imagem, numa linguagem verbal e não geométrica ou matemática. Pode também ser denominado por compreensão de imagens ou análise de cenas e usa toda a informação gerada pelos dois processos anteriores em uma tentativa de interpretar o conteúdo de uma imagem.

1.5 Segmentação da Imagem

A segmentação é parte crucial do processo, porque se não ocorrer essa segmentação da imagem corretamente, uma análise posterior pode ser impossível. As regiões dos pixels que correspondem à mão podem ser extraídas pela segmentação da cor da pele ou pela subtração do fundo. Então o contorno da mão é extraído para o processamento seguinte. As regiões detectadas são analisadas para determinar a posição e a orientação da mão.

Um método comum para segmentação de regiões em movimento em seqüências de imagem em tempo real envolve a subtração de fundo ou o limiar de erro entre uma estimativa da imagem sem objetos em movimento e a imagem atual. As numerosas abordagens para esse problema diferem no tipo de modelo de fundo usado e no procedimento usado para atualizar o modelo.

Considere “objeto de interesse”, no decorrer deste trabalho e no contexto de visão computacional, o que se pretende referenciar em oposição ao “fundo”. Por isso, será utilizada a palavra “objeto de interesse” ou mesmo “objeto” para referir, por exemplo, a mão que se pretende localizar na imagem capturada.

1.6 Objetivos

Este trabalho está focado no reconhecimento de gestos por computador através de gestos de uma mão sem nenhum dispositivo (mecânico, magnético ou óptico) colocado na mão para poder interagir com o computador (mão “limpa”) (figuras 1.2 e 1.3). Em termos de custo são levados em conta dispositivos (câmeras) de prateleira, facilmente accessíveis. O custo do sistema deve ser baixo e com boa performance, daí a utilização de *webcams* e bibliotecas de visão computacional gratuitas.

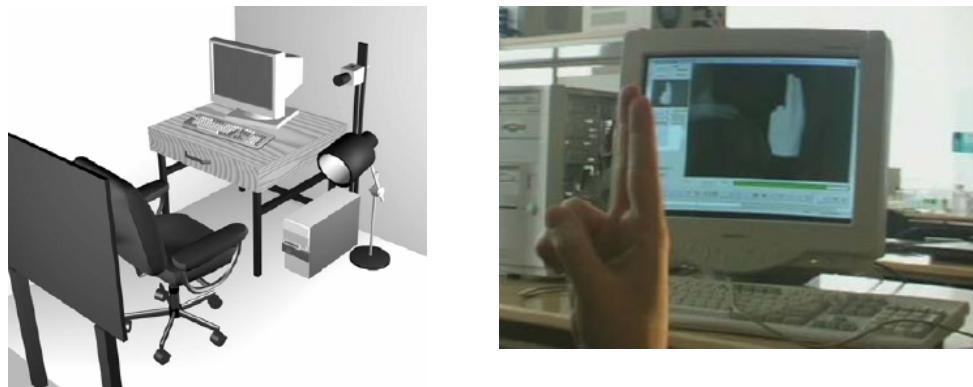


Figura 1.2: Exemplo básico dos equipamentos necessários para IUC.

Figura 1.3: Exemplo de um gesto de mão sendo capturado por um IUC.

O objetivo principal deste trabalho é mostrar que através do uso de diferentes técnicas de visão computacional combinadas com a utilização de dispositivos de baixo custo como câmeras *webcam*, é possível construir um mecanismo de interação usuário-computador sem considerar grandes restrições de ambiente, iluminação, cor dos objetos, etc. de forma que a interação com o computador seja feita da maneira mais natural possível. Para isto foi construída uma interface baseada em gestos da mão na qual os gestos e as posições das mãos são diretamente utilizados para interagir com as aplicações.

O escopo desta pesquisa é limitado ao reconhecimento de alguns gestos da mão, ou seja, identificação de qual gesto dentre os gestos predefinidos está sendo mostrado pelo

Introdução

usuário. Às vezes são realizados alguns ajustes baseados nas características dos ambientes, mas tais ajustes são pequenos e feitos numa fase de instalação de forma a não comprometer a usabilidade do sistema, principalmente relacionado com a iluminação da mão. Para avaliar o desempenho do sistema de interação baseado em gestos da mão e por sua vez avaliar o mecanismo de interação usuário-computador sem restrições significativas, uma aplicação foi desenvolvida para demonstrar o sistema em funcionamento.

Usa-se a forma geométrica da mão para interagir com sistemas computacionais na forma de gestos, como na linguagem de sinais de mão. O algoritmo de reconhecimento será baseado somente em contornos, portanto rápido o bastante para se trabalhar em tempo real. No contexto deste trabalho, serão restritos os gestos executados apenas pelas mãos, mais especificamente, por uma só mão. A primeira preocupação foi implementar algoritmos que tivessem um bom desempenho em tempo real e um modelo robusto a fim de realizar a segmentação de objetos de interesse em ambientes internos, sujeitos a diferentes condições de iluminação. Várias implementações que utilizam mistura de Gaussianas para reconhecimento de gestos usam a tecnologia DirectX da Microsoft para inserir os filtros de imagem entre os quadros na seqüência de imagens em vídeo. O DirectX é composto por um conjunto de tecnologias projetadas para transformar computadores rodando Windows em plataformas ideais para execução de programas ricos em elementos multimídia com gráficos coloridos, vídeo, animação 3D, e áudio. Oferece melhorias em desempenho dos jogos e em compatibilidade com hardware de aceleração para elementos gráficos 3D, colocando o software em contato direto com o hardware da máquina, resultando em imagens e efeitos sonoros mais realistas. Inclui suporte a sombreamento de pixels e vértices.

A proposta deste trabalho foi capturar quadro a quadro uma seqüência de imagens e realizar o tratamento em cada um, sem dependência de qualquer tecnologia do sistema operacional ou de alguma ferramenta proprietária. Para isso utilizou-se a biblioteca OpenCV (INTEL, 2004), gratuita e compatível com diversos sistemas operacionais, como o Linux. Os trabalhos tradicionais de segmentação utilizam-se de pré-treinamento, visando obter histogramas de comparações para modelo de formas de mãos e de cor de pele. Na proposta apresentada neste trabalho, as análises são feitas em tempo real, ou seja, sem um prévio treinamento para comparações de cor de pele e de formas de mãos.

As etapas a serem abordadas são:

1. Captação da imagem através de uma *webcam* para posterior processamento quadro a quadro.
2. Processamento dos quadros através de segmentação para que os pixels que pertençam às mãos do usuário sejam separados do fundo pela segmentação por subtração do fundo e filtro de cor de pele.
3. Detecção da maior área da imagem segmentada considerando-a como região da mão e extração do contorno da mão para o processamento seguinte.
4. Análise dos contornos detectados para determinar a posição, a orientação da mão e suas propriedades (momentos invariantes).
5. Rastreamento do movimento da mão para determinação de posição e de outros atributos das mãos rastreados quadro a quadro. Essas informações de movimento são úteis para o reconhecimento de gestos dinâmicos.
6. Determinação da existência de um gesto significativo baseado na posição, movimento, atributos e indícios de postura.
7. Interpretação do gesto significativo e comparação com gestos armazenados.

1.7 Organização da Monografia

O trabalho está estruturado segundo a seqüência de capítulos relacionados a seguir:

- **Capítulo 1: Introdução.** Breve introdução ao tema da dissertação
- **Capítulo 2: Interfaces Baseadas em Visão.** Neste capítulo é feita uma revisão de alguns conceitos básicos para explicar e contextualizar as interfaces baseadas em visão computacional. São abordados alguns requisitos funcionais e não-funcionais dos sistemas de visão computacional em tempo real. Um diagrama de blocos de um sistema de visão por computador, no contexto de reconhecimento de gestos, é explicado. São apresentados também alguns trabalhos relacionados a interfaces comandadas por gestos.
- **Capítulo 3: Espaços de Cores.** Este capítulo apresenta estudos sobre algumas características das cores e suas diferentes formas de representação através dos espaços de cores. São apresentados também alguns espaços de cores importantes

para a detecção de pele, bem como as transformações entre espaços de cores e suas representações gráficas. A detecção de pele é usada no auxílio da segmentação de imagens visando a separação da região da mão do restante da imagem.

- **Capítulo 4: Segmentação Usando GMM.** O objetivo deste capítulo é avaliar metodologias capazes de segmentar dinamicamente imagens de mãos visando a aplicação em IUC (Interação Usuário-Computador). Descrevem-se aqui diferentes abordagens de um algoritmo de segmentação de objetos de interesse através da estimação de um modelo robusto de cena de fundo utilizando uma mistura de distribuições de Gaussianas adaptativas para modelar cada pixel baseado no modelo proposto por Stauffer e Grimson (1999) e comparando-o com diferentes abordagens e modificações realizadas por Power e Schoones (2002) e KadewTraKuPong e Bowden (2001).
- **Capítulo 5: Pós-processamento.** Neste capítulo são descritas as técnicas utilizadas para recuperar as falhas resultantes da segmentação e ruídos na sua morfologia. Como primeiro passo são considerados os filtros morfológicos, a detecção do contorno e coleta e representação dos pontos que definem o contorno da mão. Também será mostrado como obter o contorno mais representativo da região segmentada e que tenha a forma da mão. É apresentada também a motivação geral da detecção de contorno considerando-se a sua importância no plano deste trabalho.
- **Capítulo 6: Reconhecimento de Gestos.** No decorrer deste capítulo é proposto um modelo para o reconhecimento de gestos de mão baseado no seu contorno. Além dos gestos, são apresentados como são detectadas a posição, a orientação da mão e uma série de características relevantes. Explica-se um subsistema capaz de realizar comparações entre um contorno de mão qualquer, definida por um conjunto de pontos (polígono que representa o contorno da mão), e padrões previamente estabelecidos e finalmente alguns gestos pré-estabelecidos são reconhecidos.
- **Capítulo 7: Metodologia, Resultados e Conclusões.** Capítulo com a implementação final do trabalho. São discutidas as metodologias, apresentados os resultados, discussões, as conclusões e possíveis melhorias.
- **Referências Bibliográficas**

2 INTERFACES BASEADAS EM VISÃO

2.1 Introdução

Neste capítulo, será feita a revisão de alguns conceitos básicos para explicar e contextualizar as interfaces baseadas em visão computacional. São abordados alguns requisitos funcionais e não-funcionais dos sistemas de visão computacional em tempo real. Um diagrama de blocos de um sistema de visão por computador, no contexto de reconhecimento de gestos, é explicado. São apresentados também alguns trabalhos relacionados a interfaces comandadas por gestos.

2.2 Conceitos Básicos

Os sistemas de interação baseados em visão computacional não utilizam dispositivos de rastreamento explícitos como, por exemplo, dispositivos de rastreamento ligados ao computador e ao corpo dos usuários. Eles utilizam apenas câmeras para a capturar imagens, e técnicas de processamento de imagens e reconhecimento de padrões para o rastreamento dos objetos (HANDENBERG, 2001).

É considerado um mecanismo de interação baseado em visão computacional quando a câmera é a única fonte de captura de informação, mesmo sendo utilizadas câmeras infravermelhas, sensíveis à temperatura, baseadas em distância, e outros tipos.

Quando o sistema possui uma perfeita sincronização entre suas representações física e virtual e os objetos físicos são detectados continuamente em tempo real, o sistema é denominado *fortemente acoplado* (FITZMAURICE; ISHII; BUXTON, 1995). Existe sempre, em aplicações reais, uma latência (*delay*) entre a modificação do mundo físico e a adaptação da representação virtual no computador. O termo *perfeitamente sincronizado* é usado para definir essa latência.

2.3 Requisitos Funcionais

Os requisitos funcionais podem ser definidos como sendo um conjunto de serviços que o sistema deve fornecer. Em sistemas de software existem diferentes níveis de abstração em que esses serviços podem ser desenvolvidos. Aqui, serão considerados apenas os mais básicos. Bérard (1999) identifica três serviços que os sistemas de interação usuário-computador baseados em visão computacional devem fornecer, estes são: detecção, identificação e rastreamento.

- **Detecção:** A detecção determina a presença ou ausência de uma determinada classe de objetos na imagem. Tais classes de objetos poderiam ser partes do corpo, mãos, braços, etc. Tendo como referência a imagem inteira, o processo de detecção deve ser capaz de detectar na imagem a classe de objeto que se está procurando. Uma forma de facilitar o processo de detecção é limitar o número de objetos que podem estar presentes na cena em um determinado momento. As técnicas de detecção mais conhecidas são as baseadas em cor ou movimento.
- **Identificação:** A identificação determina qual objeto, dentre um conjunto conhecido de objetos, está presente na cena. Perante a presença de objetos compostos, por exemplo, uma mão com os dedos, a identificação deve permitir determinar partes desses objetos, tais como os dedos. Outros exemplos são a identificação de símbolos escritos (STAFFORD-FRASER, 1996) de palavras na linguagem de sinais (STANER; WEAVER; PENTLAND, 1998) ou de palavras para o reconhecimento de voz. Para o nosso caso, o processo de detecção encontra a pose da mão.
- **Rastreamento:** Os objetos de interesse podem não permanecer no mesmo lugar ao longo do tempo, devido a essa característica o processo de rastreamento utiliza as informações dos dois processos anteriores para manter o foco nos objetos de interesse. Em nosso trabalho o rastreamento se refere à captura das posições do centróide da mão.

2.4 Requisitos Não-Funcionais

É indispensável definir alguns requisitos não-funcionais que estabeleçam a qualidade mínima com que os serviços devem ser implementados. Porque muitos desses sistemas de interação podem demorar horas para cumprir com todos os requisitos funcionais. Os requisitos não-funcionais considerados são: latência, resolução e estabilidade.

- **Latência:** A latência é o tempo transcorrido entre a ação do usuário e a resposta do sistema. A latência é uma característica inerente a todo sistema, não existem sistemas sem latência. Essa latência deve possuir um valor aceitável de acordo com a tarefa que o sistema deve desempenhar. A interação deve conseguir uma latência o menos perceptível para o usuário. No trabalho de Handenberg (2001) é discutido um experimento que visa encontrar um valor máximo para essa latência, o qual é determinado como sendo próximo a 50 ms ou 20Hz.
- **Resolução:** A resolução espacial (número de pixels nas imagens) deve permitir uma representação adequada do ambiente capturado. Idealmente, esse número deveria ser igual ao número de pixels existente nos monitores, mas os sistemas de captura que utilizamos no nosso trabalho como *web* câmeras não possuem ainda essa resolução e estão limitados a resoluções menores.
- **Estabilidade:** A estabilidade dos sistemas refere-se às flutuações significativas nos valores capturados (gestos ou posições de mão e dedos, neste caso). Por exemplo, um sistema pode ser considerado estável se ante um padrão (ex. mão com apenas o dedo indicador) imóvel os valores capturados não mudam significativamente. As possíveis causas de instabilidade são principalmente as flutuações nas fontes de iluminação e o ruído inerente aos dispositivos de captura.

2.5 Visão Computacional

Nos sistemas de reconhecimento de gestos, uma das partes mais importantes é a de aquisição de dados, porque é a responsável por fazer medidas do gesto que se está realizando, por exemplo, sensores óticos, acústicos e eletromagnéticos. Medidas essas que servirão para comparar e distinguir os gestos admitidos.

Quanto aos dispositivos de aquisição de dados, verifica-se a existência de duas correntes importantes: a das luvas instrumentadas e a da visão computacional. Iremos nos ater apenas à corrente de visão computacional.

No mercado existem diferentes dispositivos que permitem aos usuários usar as mãos para interagir com o computador. Alguns exemplos são teclado, *mouse*, *TrackBall*, *Track-Pad*, *Joystick* e controles remotos. Outros, mais sofisticados, incluem *Cyber-Gloves*, *3D-mice* (ex. *Labtec SpaceBall*) e dispositivos de rastreamento magnético (ex. *Polhemus Isotrack*) ou mecânico. Muitos desses dispositivos são mais baratos, confiáveis e fáceis de fazê-los funcionar do que os atuais sistemas baseados em visão computacional. A evolução dos sistemas de visão computacional, entretanto, promete resultados melhores num futuro próximo. Exemplos de luvas instrumentadas são mostrados nas figuras 2.1 a 2.4.

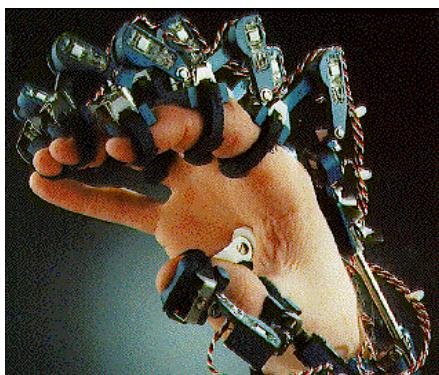


Figura 2.1: Dexterous HandMaster



Figura 2.2: Power Glove



Figura 2.3: CyberGlove

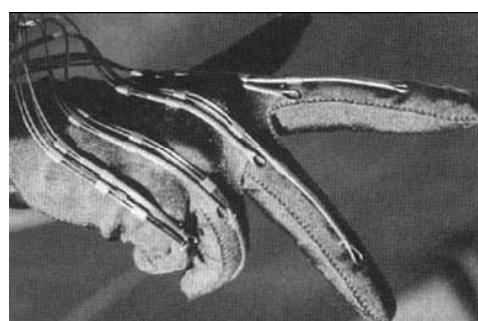


Figura 2.4: VPL DataGlove

As luvas instrumentadas são capazes de dar informação referente ao estado das junções dos dedos, que auxilia no reconhecimento do gesto. Contudo, obter esta mesma

informação, sem necessitar da luva e dos cabos que a ligam ao sistema de análise de dados, é bastante mais confortável e, além disso, permite que os usuários se movimentem com maior liberdade. Assim, surge uma outra corrente: a utilização de visão computacional em sistemas de reconhecimento de gestos. A principal vantagem da visão computacional é a não intrusividade por isso foi a opção escolhida para a utilização de nosso trabalho.

Utilizando os recursos de visão computacional, vários sistemas de reconhecimento de gestos de mão desenvolvidos são baseados na extração de algumas propriedades que estão associadas com as imagens de gestos de mão. As propriedades analisadas variam de propriedades geométricas básicas (análise de momentos da imagem) (LIAO, 1993) até propriedades que são o resultado de uma análise mais complexa (momentos de Zernike e redes neurais) (TEAGUE, 1980). O que é comum em todas as abordagens é que não resultam na estimação dos parâmetros reais da mão, como ângulos de junções. Os sistemas que usam esta análise são usados para simples rastreamento da mão e uma classificação de gestos mais complexa.

Um sistema de visão por computador, no contexto de reconhecimento de gestos, tem o diagrama de blocos da Figura 2.5, comum a muitas outras aplicações.

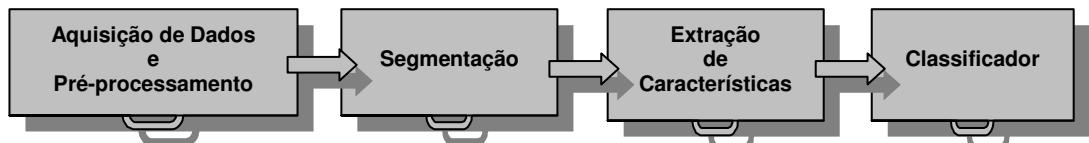


Figura 2.5: Diagrama de blocos de um sistema de visão computacional

2.6 Aquisição de Dados

A captura de imagem por uma câmera de vídeo ou mesmo uma câmera fotográfica digital constitui a primeira fase de um sistema de visão computacional. É aqui que se tomam algumas decisões importantes que influenciarão o desempenho do sistema, como por exemplo, a estrutura de iluminação, restrições de movimentos, posicionamento relativo da câmera e objeto que se deseja capturar, freqüência de amostragem e gestos admitidos.

2.7 Pré-processamento

Pode também existir pré-processamento, por exemplo, equalização do histograma de brilhos ou operações pontuais de manipulação de brilho e contraste para minimizar problemas de iluminação e realçar certos objetos de interesse. Possui a tarefa de preparar os quadros da seqüência para os processamentos posteriores suprimindo o ruído.

2.8 Segmentação

As técnicas utilizadas para a segmentação podem ser divididas em dois grupos: cor ou movimento.

- **Cor:** Segmentação através da cor assume que o objeto que se pretende detectar tem uma cor diferente dos restantes elementos presentes na cena capturada, o que nem sempre é verdade. O modelo de cor que muitas vezes se usa é HSV (*Hue, Saturation, Value*), ou também YUV (luminância, croma vermelho, croma azul) (STARK, 1995) visto que, como separam a componente de luminância das de crominância, conseguem ser um pouco menos sensíveis à variação da iluminação da cena do que o modelo RGB (*Red, Green, Blue*) (YANG, 1996).

A cor da pele humana varia extremamente entre indivíduos e sob alteração de iluminação. Algoritmos avançados de segmentação, que podem lidar com isso, foram propostos, entretanto estes são exigentes computacionalmente e ainda são sensíveis a rápidas mudanças ou variações de luz. Além disso, a segmentação pela cor da pele pode ser confundida por objetos no fundo com uma cor similar. A subtração do fundo trabalha somente em um fundo conhecido ou no mínimo um fundo estático, e consequentemente não é utilizável para uso móvel. Existem alternativas que usam marcadores nos dedos (ULHAAS; SCHMALSTIEG, 2001) ou usam luz infravermelha para realçar os elementos de pele na imagem (OKA; SATO; KOIKE, 2002).

- **Movimento:** A utilização do movimento para detectar a localização da mão, tem o pressuposto de que o único objeto em movimento na cena é a mão. Através de diferença de quadros (*frames*) consecutivos detectam-se zonas onde houve variações

de tonalidade e que, por isso, têm grande probabilidade de serem zonas onde está, ou por onde passou, a mão. Os objetos móveis na seqüência de vídeo podem ser detectados pelo cálculo das diferenças dos interquadros e fluxo ótico. Em Wong e Spetsakis (2002), é apresentado um sistema capaz de rastrear objetos móveis em um fundo móvel com câmera portátil. Entretanto, tal sistema pode não detectar uma mão imóvel ou determinar qual dos diversos objetos que se movem é a mão.

A segmentação ainda é um problema em aberto visto que a utilização da cor ou do movimento, ou a conjugação das duas, não consegue resolver totalmente o problema devido a, entre outras dificuldades, variações de cor da pele, deficiente iluminação, sombras e presença de fundos complexos (onde existem objetos com cor semelhante à dos procurados ou onde o fundo não é uniforme).

Ainda nos referindo ao grupo de segmentação de tipo cor, a segmentação utilizada nesse trabalho é a segmentação de objetos por subtração de fundo utilizando o Modelo de Mistura de Distribuições de Gaussianas (GMM). É uma abordagem em que se modela cada pixel como uma mistura de Gaussianas usando uma aproximação quadro a quadro para atualizar o modelo. O método de segmentação que explora a mistura de Gaussianas consegue resultados bem satisfatórios mesmo quando a imagem de entrada não é de alta qualidade. As distribuições Gaussianas do modelo de mistura adaptável são então avaliadas para determinar qual será a distribuição mais provável em representar o fundo.

Cada pixel é classificado baseado na distribuição de Gaussianas que o representa mais eficazmente como sendo parte da cena de fundo. O método resulta em um rastreador em tempo real estável para uso em ambientes externos que trata com confiabilidade, mudanças de luminosidade, movimentos repetitivos e desordenados, e mudanças de cenas em longos períodos (STAUFFER; GRIMSON, 1999).

2.9 Extração de Características

Num sistema de reconhecimento de gestos usando visão computacional, para extração de características, tipicamente têm-se duas abordagens relativas à natureza das técnicas utilizadas:

- **Baseada na Aparência (*appearance-based*) ou Vista (*view-based*):**

Nas técnicas baseadas na aparência, a imagem é encarada como um ponto num espaço N dimensional, em que N representa o número de pixels pertencentes à imagem.

- **Baseada nas Características (*feature-based*):**

As técnicas baseadas em características, tal como o nome indica, são extraídas características intrínsecas dos objetos presentes na imagem, tais como momentos, retângulos envolventes, áreas, contornos, cantos, etc.

Apesar de existir alguma ambigüidade nas técnicas pertencentes a cada uma destas abordagens, ao longo deste trabalho se utiliza a classificação que se entendeu mais coerente com as nossas necessidades (baixo custo computacional, sem pré-treinamento), no caso a abordagem baseada nas características.

2.9.1 Abordagem Baseada na Aparência

A maior parte das técnicas utilizadas nesta abordagem inserem-se no grupo das transformações para os chamados espaços próprios, caso de: PCA (*Principal Component Analysis*), e ICA (*Independent Component Analysis*).

Birk, Moeslund e Madsen (1997) utilizaram extensivamente PCA, tanto para determinar a orientação, num plano 2D, dos gestos capturados, como para transformar os dados (pixels da zona da mão) para um espaço que realce as diferenças (MEF – *Most Expressive Features*, segundo Swets e Weng (1996)). Para reduzir os dados com que o classificador tem de lidar, experimentaram várias técnicas para a escolha das características mais discriminantes (MDF – *Most Discriminat Features*, segundo Swets e Weng (1996)) e acabaram por escolher um método (o chamado *m-method*), que se baseia no valor relativo da variância das componentes desprezadas, porque dá resultados semelhantes aos outros métodos analisados e é de menor complexidade. A taxa de sucesso foi da ordem dos 99% num conjunto de 25 gestos e 20 imagens de treino para cada gesto. Deve-se mencionar que estes resultados não foram obtidos em tempo-real e utilizaram as imagens de treino também para teste. Uma das conclusões que tiram é que o PCA tem um desempenho muito ruim na fase de treino. Necessita de muitas imagens de treino e é extremamente lento. No entanto em funcionamento *online* consegue desempenhos de tempo-real a 14Hz (desconhecem-se as características do computador).

Histograma de orientação é outra das técnicas das baseadas na aparência e que foi a escolhida, para obtenção de vetores de características por Freeman e Roth (1995) para o seu trabalho de reconhecimento de gestos a fim de controlar um guindaste virtual ou jogar um jogo. Um dos fatores que realçam é a capacidade desta técnica poder ser aplicada em tempo-real e ter alguma imunidade à iluminação. Outro grupo das técnicas baseadas em aparência são as que têm na base o princípio de *template matching*. A imagem capturada é comparada com imagens padrão de gestos e é classificada na classe do padrão a que mais se assemelha.

Darrell, Essa e Pentland (1996) usam um conjunto de vistas representativas que servem para se confrontar com a imagem capturada, a fim de se determinar a combinação linear das imagens padrão que melhor representa a imagem capturada.

Neste mesmo trabalho, Darrell, Essa e Pentland (1996) chamam atenção para o fato da correlação normalizada ser menos precisa que metodologias baseadas em espaços próprios, apesar de ser mais adequada para fins de tempo real.

2.9.2 Abordagem Baseada nas Características

Alguns grupos de características que se podem obter são, por exemplo, os momentos geométricos de imagem, momentos de *Hu* e momentos de Zernike. Teh e Chin (1988) estudaram momentos geométricos de imagem, momentos de Legendre, momentos de Zernike, pseudomomentos de Zernike, momentos rotacionais e momentos complexos quanto à capacidade de representação da imagem, sensibilidade ao ruído e redundância de informação.

De entre os momentos estudados os melhores resultados foram obtidos pelos momentos de Zernike (TEAGUE, 1980). Uma das características principais dos momentos de Zernike é a sua invariância à rotação. Os momentos de Zernike de duas imagens idênticas, mas com diferentes rotações em uma imagem plana, apenas diferem num fator de fase, a sua amplitude é semelhante.

Uma versão adaptada dos momentos de Zernike foi utilizada por Schlenzig, Hunter e Jain (1994) no seu trabalho de reconhecimento de gestos para extração de características da imagem após segmentação.

Além dos momentos existem muitas outras características que podem ser extraídas de uma imagem, tais como: contorno, descritores de forma, área, perímetro, retângulo envolvente, elipse ajustada e centróide.

2.10 Classificador

Geralmente, são aplicados algoritmos clássicos na área de classificador de padrões. São Modelos Escondidos de Markov, correlação e redes neurais. Especialmente os dois primeiros tem sido usados com sucesso enquanto redes neurais têm problemas em modelar padrões não gestuais (LEE; KIM, 1999).

Os classificadores mais adotados são os que recorrem a Modelos Escondidos de Markov (*HMM – Hidden Markov Models*). A sua vantagem deve-se ao fato de conseguirem modelar não só a parte dinâmica de baixo-nível do gesto, mas também em certos gestos, a sua semântica (WU; HUANG, 1999).

Starner e Pentland (1995) foram dos primeiros a utilizar modelos de Markov em linguagem gestual americana (ASL). O seu modelo era constituído por quatro estados, com um de salto de transição para abranger o caso de abreviaturas, e as observações assumidas como Gaussianas, independentes e multidimensionais. Os resultados obtidos foram da ordem de 97% para o caso de frases gramaticalmente corretas.

Pavlovic, Sharma e Huang (1997) indicam que um dos problemas da utilização de HMM assumindo, para maior eficiência na parte de treino, que a função de distribuição de probabilidade das observações pode ser modelada por uma mistura de Gaussianas, o que nem sempre é verdade.

Quando é relevante analisar a semântica dos gestos pode-se utilizar uma máquina de estados finita (*FSM – Finite State Machine*). E é isso que adotam Davis e Shah (1995) para o reconhecimento de um conjunto de gestos simples conseguindo uma taxa de sucesso muito próxima de 100%.

Bobick e Wilson (1997) recorrem a uma abordagem também baseada em estados e utilizam *Dynamic Time Warping* para comparar a seqüência de estados capturada com as obtidas na fase de treino.

2.11 Interfaces Comandadas por Gestos

Uma grande parte da literatura sobre reconhecimento de gestos trata de identificar conjuntos de gestos dinâmicos como comandos individuais para um computador ou com o objetivo final de compreender linguagem de sinais. Um exemplo que propõe reconhecer a linguagem de sinais para computadores *desktop* e portáteis foi adotado por Starner *et al.* (1998).

O reconhecimento é baseado na segmentação da cor da pele para extrair a posição, forma, movimento e orientação das mãos. As mãos são modeladas como elipses, e o sistema pode obter um desempenho bom sem modelagem individual dos dedos. Usando modelos ocultos de Markov (HMM - *Hidden Markov Models*) é obtido reconhecimento contínuo de sentenças de linguagem de sinais, embora o vocabulário seja limitado a quarenta palavras.

O reconhecimento baseado na aparência de gestos estáticos é apresentado em (BIRK; MOESLUND; MADSEN, 1997), no qual as letras do alfabeto da mão são reconhecidas pela análise dos componentes principais (PCA) e por um classificador Bayesiano. A aparência dos sinais individuais é aprendida a partir de um grande número de imagens de treinamento. O PCA é usado para criar um espaço com dimensionalidade baixa onde as mãos localizadas nos quadros de vídeo podem ser comparadas com as classes que representam os gestos definidos.

As classes e o classificador correspondente são criados em um processo de aprendizagem externo. Este é o princípio das *eigen-hands* inspiradas pelas *eigen-faces*, que são usados no reconhecimento de faces (TURK; PENTLAND, 1991). O problema principal com estes modelos baseados na aparência é que são dependentes de múltiplas vistas no treinamento (FILLBRANDT; AKYOL; KRAISS, 2003).

Além do trabalho de como detectar e reconhecer gestos existem pesquisas feitas sobre os elementos intuitivos e naturais dos gestos (NIELSEN *et al.*, 2003), e como os gestos e a linguagem do corpo são usados como parte de uma comunicação inter pessoal (CASSEL, 1998).

3 ESPAÇOS DE CORES

3.1 Introdução

Este capítulo apresenta estudos sobre algumas características das cores e suas diferentes formas de representação através dos espaços de cores. São apresentados também alguns espaços de cores importantes para a detecção de pele, bem como as transformações entre espaços de cores e suas representações gráficas. A detecção de pele é usada no auxílio da segmentação de imagens visando a separação da região da mão do restante da imagem.

3.2 Espaços de Cores

De acordo com Foley *et al.* (1990), um espaço de cores é um sistema tridimensional de coordenadas, onde cada eixo refere-se a uma cor primária. A quantidade de cor primária necessária para reproduzir uma determinada cor é atribuída a um valor sobre o eixo correspondente.

A luminância é a componente da imagem que só contem as informações de brilho (tons de cinza) de uma imagem, ou seja, os valores entre o preto e o branco. Podemos dizer que a luminância é a principal componente de uma imagem, que se distingue por sua nitidez e qualidade.

A crominância é a componente que agrupa as cores a uma imagem. A crominância ou croma de uma imagem confere o colorido, sem acrescentar, entretanto maior riqueza nos detalhes que já devem estar representados pela luminância. O sinal de croma possui menos largura de faixa do que o sinal de luminância.

Colorimetria, computação gráfica e padrões de transmissão de sinal de vídeo deram origem a muitos espaços de cor com diferentes propriedades. Uma grande variedade deles foi aplicada ao problema de modelagem de cor de pele, inclusive muitas

técnicas de visão computacional utilizam espaços de cores crominantes para a detecção de pele por sua separação bem definida entre crominância e luminância.

3.2.1 Espaço RGB

Um dos espaços de cores utilizados em imagens é o espaço RGB. Este é o modelo de cores utilizado pelos monitores de computador, que possuem três canhões de luz, um para cada componente. É composto por três cores primárias *Red* (vermelho), *Green* (verde) e *Blue* (azul) que são misturadas para produzir uma cor resultante. É um modelo de cores aditivo, isto é, as cores são formadas pela adição das cores primárias e a soma das três cores resulta no branco. Para se obter uma determinada cor neste sistema, é usado um intervalo pré-especificado, normalmente de 0 a 255, sendo que a cor preta é obtida pela combinação (0,0,0), a cor branca (255,255,255) ou o azul (0,0,255). A representação do espaço de cores RGB pode ser ilustrada como um cubo, mostrado na figura 3.1, o chamado cubo RGB, onde nas extremidades (vértices) estão suas cores básicas, mais as secundárias, a preta e a branca (GONZALEZ; WOODS, 2000).

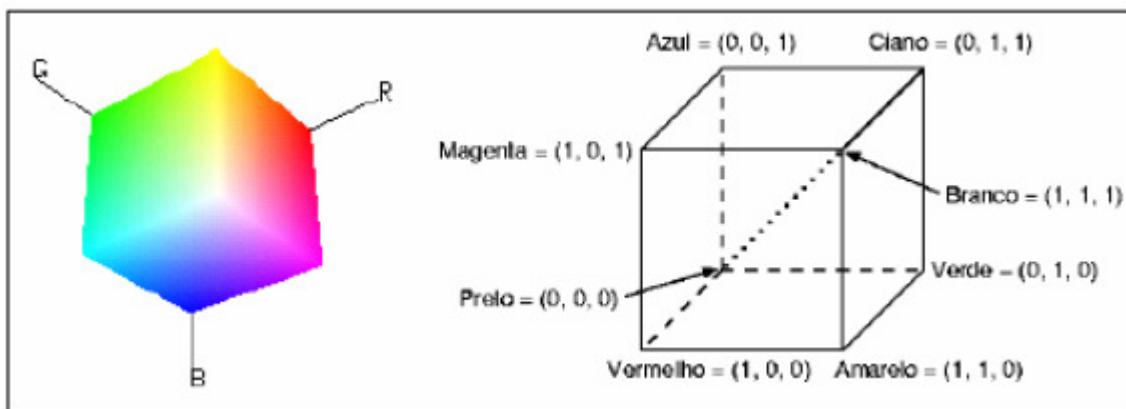


Figura 3.1: Representação do espaço de cores do modelo RGB. A linha pontilhada (diagonal do cubo) define os níveis de cinza.

Cada pixel tem seu próprio valor RGB representado por três bytes, um para cada componente de cor: vermelho, verde e azul. Esses bytes são todos empacotados em um inteiro. Uma vez que cada componente é armazenado como um byte, cada um pode representar 256 diferentes intensidades da cor correspondente. Isso significa que se pode trabalhar com mais de 16,7 milhões de cores. Isso é freqüentemente referenciado como cores reais (*truecolor*). O modelo RGB possui uma desvantagem muito forte: ele não é

bom para definição de cores baseando-se no sistema de percepção visual humano. Isso significa que nada garante que cores próximas dentro do espaço RGB sejam próximas em termos de percepção visual. Fica difícil determinar, visualmente, com exatidão, se uma cor é ou não a de interesse. Para contornar este tipo de problema surgiram outros espaços de cores como veremos nas seções seguintes.

3.2.2 Espaço RGB Normalizado

O RGB normalizado é uma representação facilmente obtida através dos valores RGB após um procedimento de normalização. Os valores r , g e b são determinados pelas equações 3.1, 3.2 e 3.3 respectivamente.

$$r = \frac{R}{R + G + B}, \quad (3.1)$$

$$g = \frac{G}{R + G + B}, \quad (3.2)$$

$$b = \frac{B}{R + G + B}. \quad (3.3)$$

Como a soma dos três componentes normalizados é conhecida ($R + G + B = 1$), o terceiro componente b não mantém nenhuma informação significativa, e pode ser omitido com o intuito de reduzir a dimensão espacial. Os componentes restantes são chamados de “cores puras”, pois a normalização diminui a dependência dos componentes r e g ao brilho do espaço de cores RGB. Uma propriedade notável deste espaço de cores é que, quando ignoramos a luz do ambiente em superfícies opacas, o RGB normalizado é invariante às mudanças de orientação das fontes de luz na superfície (STERN, 2002). Esta característica em conjunto com a simplicidade na transformação ajuda este espaço de cores a ganhar popularidade entre os pesquisadores.

3.2.3 Espaços HSV, HSL e HSI

Espaços baseados em matiz e saturação (*Hue* e *Saturation*) foram apresentados quando havia uma necessidade de se especificar as propriedades das cores numericamente. Esses espaços mostram as cores com valores intuitivos, baseados na idéia de matiz, saturação e valor. A matiz é relacionada à cor em si e define a cor dominante de uma área. Diferencia o azul do vermelho, por exemplo.

A saturação mede a pureza da cor da área. Grosso modo, a saturação é a característica que diferencia a cor rosa da cor vermelha. Enquanto a cor vermelha é uma cor pura, a cor rosa é um vermelho com alguma quantidade da cor branca. A intensidade, brilho ou valor é relacionada à luminância da cor. Diferencia o claro do escuro em cada cor da matiz.

O fato de os componentes serem intuitivos e de existir uma discriminação entre as propriedades da luminância e da crominância faz esse espaço de cores ser popular nos trabalhos de segmentação da cor da pele (BOAZ; ZARIT; QUEK, 1999), (SIGAL, 2000), (MCKENNA, 1998). Porém, existem fortes características não desejáveis destes espaços de cores, incluindo descontinuidade da matiz e o custo computacional no tratamento do “brilho” (luminância, valor ou intensidade). Essas características são um fator agravante quando tentamos realizar as técnicas de visão computacional para detecção de pele.

No espaço de cores HSV (*Hue, Saturation, Value*), trabalha-se com um hexacone (um cone com base hexagonal), no qual um ângulo *H* com respeito ao eixo horizontal determina a matiz de cor desejada, a distância perpendicular do centro até a borda determina a saturação *S* e a distância vertical, determina a luminosidade *V*. A representação desse sistema é ilustrada na figura 3.2.

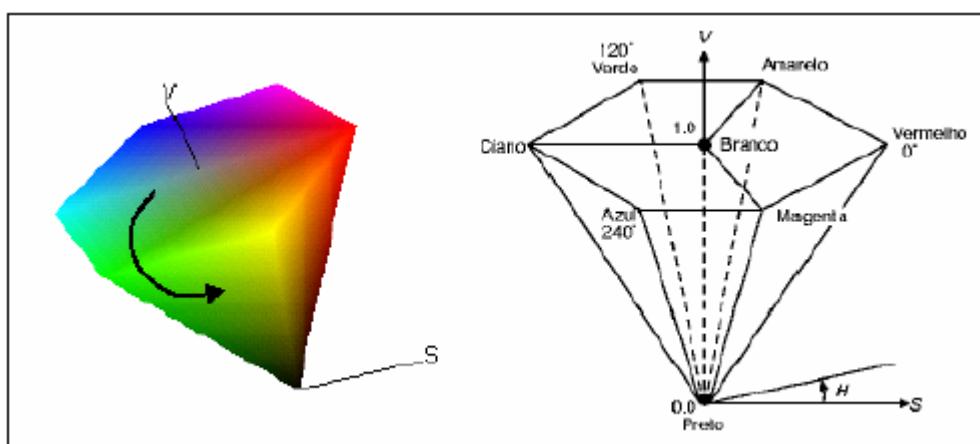


Figura 3.2:Representação do espaço de cores do modelo HSV

A transformação do espaço RGB para o espaço de cores HSV é efetuada pelas equações 3.4 a 3.6:

$$H = \arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{((R-G)^2 + (R-B)(G-B))}}, \quad (3.4)$$

$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B}, \quad (3.5)$$

$$V = \frac{1}{3}(R + G + B). \quad (3.6)$$

Existem ainda as representações piramidais duplo-hexacônica e duplo-tetraédrica, ilustradas na figura 3.3, referentes aos espaços de atributos HSL ("Hue", "Saturation", "Lightness") e HSI ("Hue", "Saturation", "Intensity"). Assim como o HSV, essas duas representações derivam do cubo referente ao espaço de cores RGB a partir de algumas transformações aplicadas em suas faces e arestas. Apesar das diferenças geométricas dessas três representações do espaço de atributos, não há significativa diferença entre os atributos matiz e saturação obtidos através dos métodos citados.

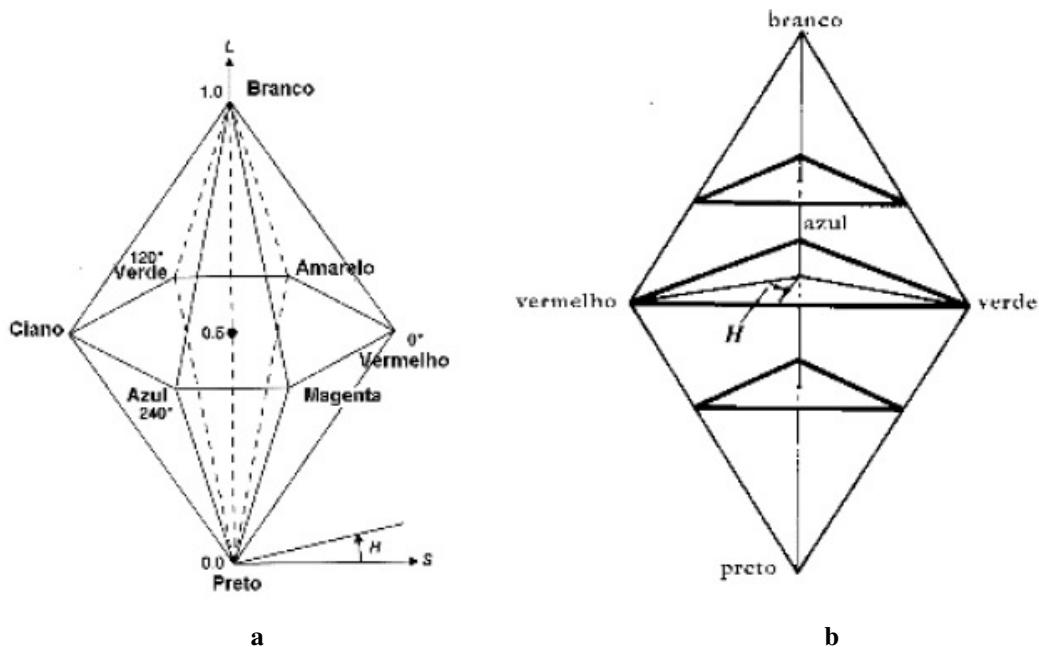


Figura 3.3: Estruturas piramidais duplo-hexacônica (a) e duplo-tetraédrica (b) do espaço de atributo de cores - HSL e HSI, respectivamente. FONTE: Adaptadas de Foley, et al. (1990, p. 594) e Gonzalez e Woods (2000, p. 230).

O espaço HSL é uma das possíveis variantes do HSV, em que as componentes de saturação e luminância são calculadas de maneira um pouco diferente. As equações abaixo 3.7 a 3.9 descrevem as diferenças entre os espaços de cores HSL e HSV:

- Para o espaço de cores HSL:
 - Saturação = $\max(R,G,B) - \min(R,G,B)$

$$\text{Saturação} = \max(R,G,B) - \min(R,G,B) \quad (3.7)$$

$$\text{Pureza} = (\max(R,G,B) + \min(R,G,B))/2 \quad (3.8)$$

- Para o espaço de cores HSV:
 - Saturação = $(\max(R,G,B) - \min(R,G,B)) / \max(R,G,B)$

$$\text{Saturação} = (\max(R,G,B) - \min(R,G,B)) / \max(R,G,B) \quad (3.9)$$

O modelo HSI é transformado a partir do modelo RGB usando as equações 3.10, 3.11 e 3.12.

$$H = \arctan'(\sqrt{3}(G - B), 2R - G - B) \quad (3.10)$$

$$S = 1 - \min(R, G, B) / I \quad (3.11)$$

$$I = (R + G + B) / 3 \quad (3.12)$$

onde $\arctan'(y,x)$ fornece o ângulo entre eixo horizontal e a linha $(0,0) - (x,y)$ (figura 3.4).

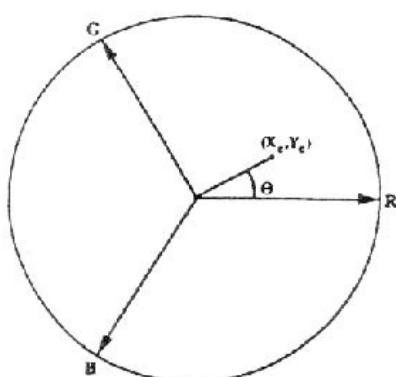


Figura 3.4: Projeção do eixo RGB em $R+B+G=0$

3.2.4 Espaço YCrCb

YCbCr é um sinal codificado RGB não-linear, comumente utilizado pelos estúdios de televisão europeus e para trabalhos de compressão de imagens (arquivos MPEG são compactados em YCbCr). A cor é representada por brilho ou luma, que é a luminância calculada de RGB não linear, criados com pesos na soma dos valores de RGB (Y), e diferença de duas cores Cr e Cb que são formados pela subtração de luma

dos componentes de vermelho ($R-Y$) e azul ($B-Y$). Este espaço pode ser equacionado diretamente a partir do RGB como mostrado pelas equações 3.13 a 3.15.

$$Y = 0.299R + 0.587G + 0.114B \quad (3.13)$$

$$Cr = (R - Y) 0.713 + 128 \quad (3.14)$$

$$Cb = (B - Y) 0.564 + 128 \quad (3.15)$$

A figura 3.5 mostra graficamente o espaço de cores YCbCr. É possível observar a variação da luminância pelo eixo Y . Observem que para Y mínimo a cor mostrada é preta para qualquer crominância e para Y máximo a cor mostrada é branca para qualquer crominância.

A simplicidade na transformação e a separação explícita dos componentes luminância e crominância faz este espaço de cores se tornar atrativo para modelagem das cores dos pixels de pele (PHUNG, BOUZERDOUM; CHAI, 2002), (ABDEL-MOTTALEB; HSU; JAIN, 2002), (BOAZ; ZARIT; QUEK, 1999).

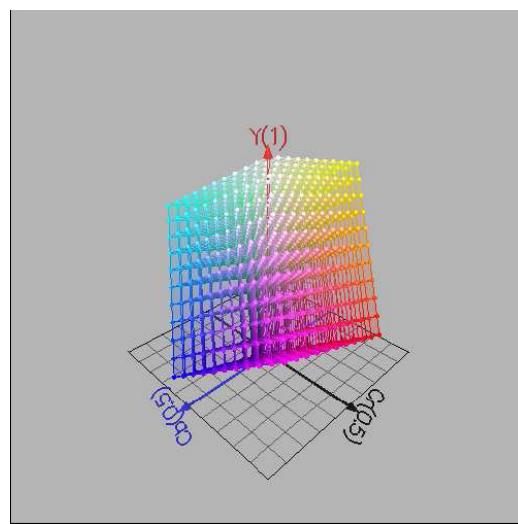


Figura 3.5: Espaço de cores YCbCr.

3.2.5 Espaços YIQ e YUV

Existem também outros espaços usados como sistemas de cor para TV. O sistema americano utiliza o espaço **YIQ**. Este espaço é descrito por três componentes: A luminância Y e duas componentes de crominância I e Q que carregam a informação de cor. A matriz de transformação do espaço de cores RGB para YIQ é dada pela equação 3.16:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.16)$$

O sistema europeu utiliza-se do espaço de cores **YUV**, cuja matriz de transformação a partir do espaço de cores RGB é dada pela equação 3.17:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.17)$$

3.3 Detecção de Cor de Pele

A detecção de mãos tem sido tópico de extensiva pesquisa e muitas estratégias baseadas em técnicas heurísticas e de padrões de reconhecimento vêm sendo propostas para alcançar soluções robustas e precisas. Entre métodos de detecção de característica de mão, os que usam a cor da pele como indicador de detecção, ganharam forte popularidade. A cor permite processamento rápido e é bastante robusta às variações geométricas de padrões de mão. Ao construir um sistema, usando a cor da pele como característica para detecção, geralmente enfrentam-se três problemas principais.

- Primeiramente, como escolher o melhor espaço de cores.
- Como a distribuição da cor da pele seria exatamente modelada.
- E finalmente, qual seria a maneira de processar os resultados da segmentação da cor para a detecção da mão.

3.3.1 Trabalhos Relacionados

Garcia e Tziritas (1999) aplicaram a técnica de quantização de vetores para identificar a cor da pele nos espaços HSV e YCbCr, sendo que neste, a intensidade tem pouca influência na distribuição projetada no plano de crominância *CbCr*. A nuvem de pontos varia seu formato no eixo *Y*, levando a dois modelos distintos para um envoltório da nuvem: um para $Y > 128$ e outro para $Y \leq 128$;

Schumeyer e Barner (1998) relatam a dependência entre intensidade e cromaticidade, no espaço de cores RGB, e citam a preferência de outros autores na

escolha de espaços que separam estas componentes. Particularmente, são enfatizadas as vantagens do espaço $L^*a^*b^*$, devido a sua linearidade perceptiva e por apresentar uma distribuição da cor de pele mais compacta e uniforme.

Wang e Chang *apud* (1997) (GARCIA; TZIRITAS, 1999) executaram a classificação da cor da pele neste espaço, diretamente sobre o plano de crominância. Kawato e Ohya (2000), observando que a distribuição de pixels de pele no espaço de cores RGB predomina ao longo da diagonal principal do cubo (0,0,0)–(255,255,255), definiram um plano perpendicular a este eixo, onde as variações de crominância possam ser mais precisamente percebidas.

Martinkauppi, Soriano e Laaksonen (2001) apresentam a distribuição de cor da pele sob diferentes espaços de cores, utilizando câmeras diferentes. Observa-se que a nuvem de pontos cor da pele nestes espaços é compacta, alongada e ligeiramente curva, e não é suficientemente simétrica para ser descrita por uma distribuição simples monofuncional. Em espaços que separam a intensidade da cromaticidade, a distribuição no plano de cromaticidades é dependente da intensidade. Ainda, Martinkauppi, Soriano e Laaksonen (2001) e Yang, Lu e Waibel (1997) mostraram que os pixels associados à pele constituem aglomerados compactos na maioria dos espaços de cores.

Um resultado bastante incisivo refere-se à escolha do melhor espaço de cores para detecção de pele (SHIN; CHANG; TSAP, 2002). Através de medidas como a separabilidade entre grupos pele e não-pele, e análise de histogramas, concluiu-se que a melhora na separação é mínima na transformação entre espaços de cores e que a eliminação da componente intensidade não melhora significativamente a precisão na detecção da pele.

Como conclusão geral, os melhores espaços em separabilidade são o YCbCr e o RGB, e ainda, o espaço RGB ocupou sempre a primeira ou segunda posição em cinco das oito medidas de desempenho. Ou seja, “*a maioria das transformações de espaços de cores não auxiliaram na detecção da pele*”. Conclui-se que um esforço computacional extra na mudança do espaço RGB para outro qualquer é desnecessário, devido à variedade de tons de pele observados nestas bases de dados, e não melhora a acurácia da separação.

3.3.2 Faixas de Agrupamentos de Pele em Espaço de Cores

O objetivo final da modelagem de cor de pele é construir uma regra de decisão que discrimina os pixels de pele e os de não-pele. Isso é realizado geralmente introduzindo uma métrica, que calcula as medidas entre a cor do pixel e o tom da pele. O tipo desta métrica é definido pelo método de modelagem de cor de pele.

Métodos para se construir classificadores de cor de pele podem ser definidos empiricamente por regras de decisões sobre agrupamentos de cor de pele em algum espaço de cor. Por exemplo Peer, Kovac e Solina (2003), onde (R,G,B) é classificado como um pixel de cor de pele se a equação 3.18 for satisfeita:

$$\begin{aligned} & (R > 95) \text{ e } (G > 40) \text{ e } (B > 20) \text{ e} \\ & ((\max\{R,G,B\} - \min\{R,G,B\}) > 15) \text{ e } (|R-G| > 15) \text{ e} \\ & (R > G) \text{ e } (R > B) \end{aligned} \quad (3.18)$$

A simplicidade deste método tem atraído muitos pesquisadores (PEER; KOVAC; SOLINA, 2003), (AHLBERG, 1999), (FLECK; FORSYTH; BREGLER, 2002), (JORDAO *et al.*, 1999). A principal vantagem deste método é a simplicidade das regras de detecção de pele que conduz à construção de um classificador muito rápido. A principal dificuldade de alcançar elevada taxa de reconhecimento com este método é a necessidade encontrar de forma empírica um bom espaço de cor e também regras de decisão adequadas.

Em Kühl e Silva (2004) foram utilizadas faixas de RGB normalizado e a componente V de HSV. O RGB normalizado foi utilizado e para completá-lo utilizaram também o componente V do modelo HSV. Um pixel é transformado em preto se estiver nos intervalos aceitos pela equação 3.19:

$$(0.35 \leq r \leq 0.465) \text{ e } (0.28 \leq g \leq 0.363) \text{ e } (0.25 \leq V \leq 1), \quad (3.19)$$

onde r , g e V são obtidas através das equações 3.1, 3.2 e 3.20:

$$V = \frac{\max(R, G, B)}{255}. \quad (3.20)$$

Em Buhiyan, Ampornaramveth e Ueno (2003) a cor da pele é utilizada para determinar a região desejada. As cores relevantes e dominantes são extraídas da imagem em RGB. Em seguida a imagem é transformada para o espaço de cores YIQ descrito por

três atributos: luminância, matiz (*hue*) e saturação. Esse sistema de cores é obtido diretamente a partir de RGB pela equação 3.16 (descrito na seção 3.2.5 deste capítulo).

Uma vez que a cor da pele tende a se aglomerar numa região do espaço de cores, um limiar (*threshold*) é utilizado para detectar os pixels de pele. Nos experimentos de Buhiyan, Ampornaramveth e Ueno (2003) os limiares utilizados, determinados empiricamente, são dados pela equação 3.21:

$$(60 < Y < 200) \text{ e } (20 < I < 50). \quad (3.21)$$

Duan *et al.* (2000) desenvolveram um método de detecção de pele baseado na percepção humana das cores. Eles treinaram uma grande quantidade de imagens com pixels de pele e então converteram os pixels de RGB para YUV e YIQ respectivamente. Obtiveram, então, uma distribuição de pixels de cor de pele. Os valores RGB foram transformados usando as equações 3.16 e 3.17 (descritos na seção 3.2.5).

A informação de crominância está codificada nas componentes U e V . A saturação e a matiz são obtidas pelas equações 3.22 e 3.23:

$$Ch = \sqrt{|U|^2 + |V|^2}, \quad (3.22)$$

$$\theta = \tan^{-1}(|V| / |U|), \quad (3.23)$$

onde θ representa a matiz, que é definido como o angulo do vetor no espaço de cores YUV e Ch representa a saturação, que é definida como módulo de U e V . Os autores observaram que a faixa de matiz θ para a maioria de cores peles de pessoas variavam de 100° a 150° . No espaço de cores YIQ, I é o eixo vermelho-laranja, e Q é aproximadamente ortogonal a I . Menos I significa menos azul-verde e mais amarelo. Também observaram que a faixa de I para a maioria de tipos de peles variava de 20 a 90. Após algumas experiências descobriram que combinando os espaços de cores YUV e YIQ alcançavam maior robustez que cada um isolado. Assim se o pixel satisfaz as faixas determinadas pela equação 3.24 é possível que seja um pixel com cor de pele.

$$(20 \leq I \leq 90) \text{ e } (100 \leq \theta \leq 150) \quad (3.24)$$

Recentemente, foi proposto um método que usasse algoritmos de aprendizagem de máquina (*machine learning algorithms*) para encontrar o espaço de cor apropriado e uma regra de decisão simples que consigam taxas elevadas do reconhecimento (GOMEZ; MORALES, 2002). Os autores começam com um espaço normalizado do

RGB e então aplicam um algoritmo de indução construtiva (*constructive induction algorithm*), para criar um número de novos conjuntos de três atributos que são uma superposição de r , g , b e uma constante $1/3$, construídos por operações de aritmética básica. Uma regra de decisão, similar à descrita acima que consegue o melhor reconhecimento possível, é estimada para cada conjunto de atributos.

Os autores proíbem a construção de regras demasiadamente complexas, o que ajuda a evitar a criação excessiva de dados (*overfitting*). Isso é possível deixando de treinar conjuntos pouco representativos. Eles conseguiram resultados que superaram o rendimento do classificador do mapa de probabilidade de pele de Bayes no espaço de cor RGB para seu conjunto de dados.

Gomez, Sanchez e Sucar (2002) usaram a proposta de análise de dados para seleção de componentes de cores de diferentes espaços de cores visando à detecção de pele. Eles avaliaram cada componente de uma série de espaços de cor, como HSV, YIQ, RGB-Y, CIE XYZ. Imagens com e sem cor de pele de mais de 1000 pessoas e com diferentes tons de pele foram captadas com diferentes câmeras, em ambientes internos e externos com diferentes condições de luz. Essas imagens foram usadas para explorar o desempenho dos componentes de cor e descobrir componentes complementares usando por exemplo, PCA e χ^2 . Eles descobriram que o espaço híbrido 3D (H-GY-Wr) a partir de $H(\text{HSV})$, $GY(\text{RGB-Y})$ e Wr (equação 3.25) alcançam desempenho melhor, com 97% de pixels de cor de pele detectados e 5,16% de falso positivos.

$$Wr = \left(\frac{r}{r+g+b} - \frac{1}{3} \right)^2 + \left(\frac{2}{r+g+b} - \frac{1}{3} \right)^2 \quad (3.25)$$

Um pixel é considerado pertencente a um agrupamento de cor de pele no o espaço híbrido 3D (H-GY-Wr) se obedecer à equação 3.26:

$$-17.4545 < H < 26.666 \quad e \quad GY < -5.9216 \quad e \quad Wr < 0.0271 \quad (3.26)$$

4 SEGMENTAÇÃO USANDO GMM

4.1 Introdução

O objetivo deste capítulo é avaliar metodologias capazes de segmentar dinamicamente imagens de mãos visando a aplicação em IUC (Interação Usuário-Computador). Descrevem-se aqui diferentes abordagens de um algoritmo de segmentação de objetos de interesse através da estimação de um modelo robusto de cena de fundo utilizando uma mistura de distribuições de Gaussianas adaptativas para modelar cada pixel baseado no modelo proposto Stauffer e Grimson (1999) usando o espaço de cores RGB e comparando-o com diferentes abordagens e modificações realizadas por Power e Schoones (2002) e KadewTraKuPong e Bowden (2001). Também é utilizado um método de detecção de cor de pele, abordado por Peer, Kovac e Solina (2003), para os pixels considerados não pertencentes à cena de fundo, visando utilização em reconhecimento de gestos de mão.

4.2 Segmentação

O primeiro passo na análise de imagens é a segmentação que consiste em definir na imagem, recortes automáticos ao redor de objetos de interesse. A segmentação subdivide uma imagem em suas partes ou objetos constituintes. O nível até o qual essa subdivisão deve ser realizada, assim como a técnica utilizada, depende do problema que está sendo resolvido. Algoritmos de segmentação permitem achar diferenças entre dois ou mais objetos, e distinguir as regiões umas das outras e do fundo. Esta distinção permitirá interpretar pixels contíguos e agrupá-los em regiões. Os algoritmos de segmentação para imagens monocromáticas são geralmente baseados em uma das seguintes propriedades básicas de valores de níveis de cinza:

- **Descontinuidade:** Essa abordagem consiste em particionar a imagem baseando-se em mudanças bruscas nos níveis de cinza. As principais áreas de interesse são a detecção de pontos isolados, detecção de linhas e bordas na imagem.
- **Similaridade:** As principais abordagens baseiam-se em limiarização e crescimento de regiões.

Uma das principais dificuldades encontradas para realizar essa tarefa é a segmentação da mão em movimento com fundos complexos. Em diversas aplicações baseadas em técnicas de visão computacional, como sistemas de inspeção e monitoramento, reconhecimento de faces e gestos, a segmentação dos objetos de interesse numa cena dinâmica é uma etapa crítica e primordial para todo o processamento subsequente. Tradicionalmente, para a realização da segmentação, utiliza-se a subtração da cena de fundo da cena corrente. A grande dificuldade desta abordagem, entretanto, está na construção de um modelo robusto de cena de fundo, adaptável a variações de iluminação, a sombras e reflexões criadas pelos objetos de interesse e às diferentes texturas das regiões da própria cena. Este modelo deve ainda lidar com situações como cenas de fundo dinâmicas e alto tráfego de objetos de interesse na cena.

Com os avanços dos sistemas computacionais, muitas destas dificuldades têm sido suplantadas e diversas abordagens para o problema da estimativa de um modelo robusto de cena de fundo têm sido desenvolvidas. Contudo, algumas destas abordagens são muito custosas computacionalmente, chegando até a serem proibitivas para aplicações comerciais, que na maioria das vezes têm requisitos de funcionamento em tempo real.

Algoritmos avançados de segmentação, que podem lidar com isso, foram propostos e entre eles existe um em que se modela cada pixel como uma mistura de distribuições de Gaussianas (GMM) usando uma aproximação quadro a quadro para atualizar o modelo, idéia original de Stauffer e Grimson (1999). O método de segmentação que explora a mistura de Gaussianas consegue resultados bem satisfatórios mesmo quando a imagem de entrada não é de alta qualidade. Cada pixel é classificado baseado na distribuição de Gaussianas que o representa mais eficazmente como sendo

parte da cena de fundo. O método resulta em um rastreador em tempo real estável para uso em ambientes fechados ou externos que trata mudanças de luminosidade, movimentos repetitivos, desordenados e mudanças de cenas em longos períodos.

Entre métodos de segmentação de regiões de uma imagem com mãos, os que usam a cor da pele como indicador de detecção ganharam forte popularidade. A cor permite processamento rápido e é bastante robusto às variações geométricas de padrões de mão. Também, experiências sugerem que a pele humana tem uma cor característica, que é reconhecida facilmente por seres humanos (VEZHNEVETS *et al.*, 2003) (GOMEZ; MORALES, 2002). Mas a cor da pele varia extremamente entre indivíduos e sob alteração de iluminação. Além disso, a segmentação pela cor da pele pode ser confundida por objetos no fundo com uma cor similar. Apesar dos problemas, empregar a cor da pele para a segmentação de mão é uma idéia adotada pela facilidade e simplicidade do processo. Dentre esses métodos destacam-se os métodos de definição explícita de região de cor de pele (PEER; KOVAC; SOLINA, 2003).

4.3 Trabalhos Relacionados

Têm-se desenvolvido diversas abordagens para modelar cenas de fundo. Uma idéia básica e intuitiva do processo, baseada na subtração da cena de fundo, foi implementada por Heikkila e Silven (1999), em que a cada novo quadro realiza-se uma subtração das intensidades (cores) por uma imagem de referência, seguida de uma limiarização. Apesar do baixo custo computacional, essa técnica não lida com as diversas dificuldades na aplicação de interesse, além de necessitar de uma imagem de referência que não contivesse os objetos de interesse.

Wren *et al.* (1997), com o Pfinder, introduziram a idéia de utilizar um modelo estatístico baseado no atributo cor, utilizando uma única distribuição de Gaussianas por pixel, construída após um período de treinamento com a cena de fundo sem os objetos de interesse. O modelo do Pfinder, entretanto, não é robusto para cenas de ambientes externos, submetidas a todas as diferentes condições já citadas. Esse modelo do Pfinder foi estendido por Stauffer e Grimson (1999), que introduziram a utilização de uma mistura de distribuições de Gaussianas adaptativas para modelar cada pixel da cena de

fundo e de um algoritmo de componentes conectados para segmentar os objetos de interesse.

Gao *et al.* (2000) compararam a utilização de uma única distribuição de Gaussianas com uma mistura de distribuições de Gaussianas para a modelagem do atributo cor de uma cena de fundo e comprovaram que a última de fato é a melhor representação para a cena de fundo, mesmo em cenas estáticas. Com isso, o modelo de Stauffer e Grimson tornou-se bastante popular e utilizado por diversos autores em inúmeras aplicações que requerem estimativa da cena de fundo. Este modelo, contudo, não é totalmente adaptável a todas diferentes condições de uma cena de ambiente externo, apresentando falhas na ocorrência de sombras e reflexões, além de muitos erros quando implementado em cenas com objetos de interesse quase estáticos ou em alto tráfego.

Diversos autores têm proposto diversas modificações no algoritmo original de Stauffer e Grimson (1999) para contornar os referidos problemas, como, por exemplo, o uso conjunto da mistura de distribuições de Gaussianas utilizando as intensidades e a detecção de bordas ou detecção de sombras. Outra alteração interessante é a proposta por Harville, Gordon e Woodfill (2001), que além do atributo cor, a mistura de distribuições de Gaussianas seria formada por um novo atributo: a profundidade. Este novo atributo é extraído de um arranjo estéreo binocular e sua adição permite ao modelo incorporar as sombras e reflexões. Harville, Gordon e Woodfill (2001) também criaram um nível de atividade da cena para controlar a velocidade com que um objeto é incorporado ao modelo de cena de fundo, solucionando o problema dos objetos de interesse quase estáticos ou em alto tráfego.

KadewTraKuPong e Bowden (2001) propuseram alterações nas equações de atualizações do algoritmo padrão. Existem diferentes equações para diferentes fases, permitindo segundo os autores, acelerar o processo de aprendizado e com maior precisão, podendo adaptar-se com mais eficiência às mudanças no ambiente. Também foi introduzida uma detecção de sombras no algoritmo.

Power e Schoones (2002) também sugerem melhorias de desempenho através de aproximações das equações do algoritmo padrão que deixa de usar nas equações a função densidade de probabilidade da distribuição detectada como fundo e passa a

utilizar a relação taxa de aprendizagem sobre o peso que é a freqüência que a distribuição é identificada como fundo.

Outras abordagens também têm sido desenvolvidas, utilizando-se como alternativa aos modelos estatísticos, derivadas temporais, filtros de predição de Wiener (KOLLER *et al.*, 1999), filtros de Kalman (BROWN; HWANG, 1992), teoria de decisão bayesiana e análise de componentes principais (BIRK; MOESLUND; MADSEN, 1997), e Modelos Ocultos de Markov (STANER; WEAVER; PENTLAND, 1998). Mas muitas destas soluções apresentam custo computacional elevado, quando comparadas com o modelo da mistura de distribuições de Gaussianas adaptativas ou resultados piores na estimativa de um modelo robusto de cena de fundo.

4.4 Modelo de Cena de Fundo

A idéia básica do algoritmo GMM é definir uma região segmentada delimitando-se os pixels de interesse. Para isso, modela-se o atributo cor de cada pixel numa seqüência de quadros (processamento pontual) através de uma mistura de distribuições de Gaussianas adaptativas. O modelo é atualizado a cada nova observação obtida, diminuindo a influência das observações passadas, permitindo adaptar-se às variações graduais de iluminação da cena. Entretanto, estas distribuições de Gaussianas representam tanto o modelo de cena de fundo quanto o objeto de interesse (frente), sendo necessária a definição de um subconjunto destas para descrever o modelo de cena de fundo. A definição deste subconjunto é realizada a cada nova observação, de acordo com pesos associados a cada distribuição, os quais indicam a freqüência que aquela distribuição melhor representou o pixel.

Após o processamento pontual, os resultados são submetidos a uma etapa de segmentação por cor de pele somente nos pixels considerados como sendo pixels de frente. Com a máscara obtida realiza-se uma etapa de remoção de ruídos, na qual são aplicados um filtro gaussiano e operações de morfologia matemática com o objetivo de atenuar classificações errôneas.

4.5 Modelo de Misturas de Gaussianas – GMM

O valor de um pixel decorre da aquisição da energia refletida por uma certa superfície sob uma condição de iluminação particular, podendo, assim, ser modelado por uma distribuição de Gaussianas simples, considerando algum ruído que por ventura ocorra na aquisição da imagem. Se apenas houvesse variação na iluminação da cena, uma distribuição de Gaussianas simples adaptativa poderia perfeitamente resolver esse problema (YANG; AHUJA, 1999). Entretanto, em aplicações em que há variação das superfícies que compõem a cena, além da variação de iluminação, faz-se necessário a utilização de múltiplas distribuições de Gaussianas adaptativas (mistura de K Gaussianas) (BISHOP, 1995).

Define-se o histórico de um pixel como sendo uma série temporal dos valores deste pixel, que são vetores $\vec{X}_{i,t} = \{R_{i,t}, G_{i,t}, B_{i,t}\}$. A cada instante t , para cada pixel $i = \{x_0, y_0\}$, pode-se representar seu histórico como na equação 4.1:

$$\{\vec{X}_{i,1}, \dots, \vec{X}_{i,t-1}\} = \{\vec{I}(x_0, y_0, j) : 1 \leq j \leq t-1\}, \quad (4.1)$$

onde I é a seqüência de quadros. Um vetor $\vec{X}_{i,t}$, pode ser 1-dimensional (monocromático), 2-dimensional (espaço de cores normalizados), 3-dimensional (espaço de cor RGB) ou D -dimensional, generalizando (representada por colunas de vetores).

A probabilidade de um pixel atual possuir o valor ou intensidade $\vec{X}_{i,t}$ no instante t pode ser estimada com a equação 4.2:

$$P(\vec{X}_{i,t} | \vec{X}_{i,1}, \dots, \vec{X}_{i,t-1}) = \sum_{k=1}^K \omega_{i,t-1,k} * \eta(\vec{X}_{i,t}; \bar{\mu}_{i,t-1,k}, \Sigma_{i,t-1,k}), \quad (4.2)$$

onde $\omega_{i,t-1,k}$ é o peso associado à k -ésima distribuição de Gaussianas na mistura, no tempo $t-1$, os quais indicam as proporções relativas às observações passadas modeladas por cada distribuição de Gaussianas; $\bar{\mu}_{i,t-1,k}$ e $\Sigma_{i,t-1,k}$ são os valores do vetor-média e matriz de covariância dessa k -ésima distribuição de Gaussianas e η é a função densidade de probabilidade normal dada pela equação 4.3:

$$\eta(\vec{X}_{i,t}, \vec{\mu}_{i,t-1,k}, \Sigma_{i,t-1,k}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\sum_{i,t-1,k}|^{\frac{1}{2}}} * \exp\left\{-\frac{1}{2} (\vec{X}_{i,t} - \vec{\mu}_{i,t-1,k})^T \sum_{i,t-1,k} (\vec{X}_{i,t} - \vec{\mu}_{i,t-1,k})\right\}, \quad (4.3)$$

onde D é o D -dimensional do vetor $\vec{X}_{i,t}$, neste caso $D = 3$ porque o espaço de cores utilizado foi o RGB.

A idéia central por trás da estimativa de densidades de probabilidade utilizando misturas de funções normais, está na utilização de uma combinação de distribuições simples, no caso Gaussianas, para modelar distribuições de complexidade arbitrária. Assume-se que a matriz de covariância deva ser $\sum_{i,t-1,k} = \sigma_{i,t-1,k}^2 I$ por razões de custo computacional. Portanto convencionou-se utilizar a matriz de covariância $\Sigma = \text{diag} [\sigma_R^2 \ \sigma_G^2 \ \sigma_B^2]$, onde σ_R^2, σ_G^2 e σ_B^2 são, respectivamente, as variâncias das componentes RGB, assumindo que estas componentes são independentes e possuem o mesmo valor de variância (STAUFFER; GRIMSON, 1999).

Tradicionalmente a escolha do valor k é igual para todos os pixels, sendo tipicamente na faixa de 3 a 5. Quanto mais distribuições forem usadas, melhor será a representação das cenas pelo modelo, acarretando, entretanto, maior custo computacional.

A cada nova observação, a mistura de distribuições de Gaussianas utilizada para modelar a história de observações de cada pixel deve ser atualizada. Idealmente, a cada instante t , deve-se re-estimar todos os parâmetros da mistura de distribuições de Gaussianas de cada pixel aplicando-se um algoritmo de Maximização de Expectância (*Expectation Maximization*) (BILMES, 1998) (REDNER; WALKER, 1984) a alguma janela de recentes observações, já incluindo a última. Este procedimento, porém, é demasiadamente custoso, podendo-se empregar sem maiores prejuízos uma aproximação com o algoritmo K -médias (STAUFFER; GRIMSON, 1999). Essa aproximação pode ser vista como a busca pela distribuição que melhor representa o pixel atual a fim de atualizar os parâmetros de η_k utilizando a observação corrente. A busca é realizada através do cálculo da distância, normalmente euclidiana, entre a observação corrente e as k distribuições do modelo.

No caso em que duas ou mais distribuições tenham a mesma distância em relação à observação atual, escolhe-se a distribuição de Gaussianas com a maior relação

peso/variância. Este critério foi adotado porque se espera que uma distribuição de Gaussianas que represente a cena de fundo tenha grande peso (ocorre freqüentemente) e baixa variância (varia pouco no tempo).

A busca falha quando as distâncias calculadas entre as distribuições e observação corrente forem maiores do que um valor β (limiar de desvio padrão), geralmente da ordem de 2.5, a fim de englobar 95% da função densidade de probabilidade normal e devem satisfazer o critério $|X_t - \mu| \leq 2.5\sigma$ para todas componentes de RGB (STAUFFER; GRIMSON, 1999).

Se a busca falhar, a distribuição com a menor relação peso/variância é substituída por uma nova que representa a observação atual, sendo os seus parâmetros inicialmente ajustados para uma alta variância, um baixo peso e a média com o valor do pixel corrente.

No caso de sucesso da busca, a atualização dos parâmetros da distribuição de Gaussianas que melhor modela a observação corrente é dada pelas equações 4.4 a 4.7:

$$\vec{\mu}_{i,t,k} = (1 - \rho)\vec{\mu}_{i,t-1,k} + \rho X_{i,t}, \quad (4.4)$$

$$\sigma_{R,i,t,k}^2 = (1 - \rho)\sigma_{R,i,t-1,k}^2 + \rho(R_{i,t} - \mu_{R,i,t-1,k})^2, \quad (4.5)$$

$$\sigma_{G,i,t,k}^2 = (1 - \rho)\sigma_{G,i,t-1,k}^2 + \rho(G_{i,t} - \mu_{G,i,t-1,k})^2, \quad (4.6)$$

$$\sigma_{B,i,t,k}^2 = (1 - \rho)\sigma_{B,i,t-1,k}^2 + \rho(B_{i,t} - \mu_{B,i,t-1,k})^2, \quad (4.7)$$

onde ρ é calculado pela equação 4.8.

$$\rho = \alpha * \eta(\vec{X}_{i,t}; \vec{\mu}_{i,t-1,k}, \sigma_{i,t-1,k}), \quad (4.8)$$

e α ($0 < \alpha \leq 1$) é denominada taxa de aprendizagem e efetivamente, a constante de tempo $1/\alpha$, determina a velocidade que os parâmetros da distribuição mudam e incorporam a nova observação, podendo ser ajustada de acordo com o objetivo da implementação. Uma observação importante é que todas as variâncias não podem ser decrescidas abaixo de um valor mínimo, como forma de evitar instabilidade na busca em regiões da cena que permaneçam estáticas por um longo período de tempo.

Os parâmetros das outras distribuições de Gaussianas que não modelam a observação corrente permanecem inalterados. Outro importante parâmetro é o peso de cada distribuição na mistura, que devem ser ajustados como na equação 4.9:

$$\omega_{i,t,k} = (1 - \alpha) \omega_{i,t-1,k} + \alpha M_{i,t,k}, \quad (4.9)$$

onde é $M_{i,t,k}$ é 1 para a distribuição que melhor modela a observação corrente e 0 para as demais.

Tanto para o caso de sucesso na busca, quanto para o caso de falha, os pesos devem ser renormalizados. A cada instante, uma ou mais distribuições de Gaussianas de cada mistura são selecionadas como modelos de cena de fundo para um determinado pixel, enquanto as outras são classificadas como modelos de objetos de interesse. A escolha das distribuições que representam o modelo de cena de fundo é realizada primeiramente ordenando-se decrescentemente as K distribuições de Gaussianas pela relação peso/variância.

Em seguida, escolhe-se as B primeiras distribuições como modelo de cena de fundo de acordo com o critério: $B = \arg \min_b (\sum_{k=1}^b \omega_{i,k} > T)$, onde T é um limiar de cena de fundo previamente fixado. Se a distribuição que melhor modela a observação corrente, no caso de sucesso da busca, for uma dessas primeiras distribuições, o pixel atual é classificado como pertencente ao modelo de fundo. Se a busca falhar ou a distribuição não pertencer a estas B primeiras, o pixel é classificado como objeto de interesse.

O procedimento anterior, é extremamente dependente da taxa de aprendizagem α . Para valores elevados desta taxa, os objetos de interesse que por ventura permaneçam praticamente estáticos na cena, serão rapidamente incorporados ao modelo de fundo. Já no caso de valores baixos destas taxas, poderiam resultar na classificação errônea de determinadas partes da cena de fundo como objetos de interesse por um longo período de tempo. Para contornar estas dificuldades, existe a possibilidade de se utilizar uma taxa de aprendizagem adaptativa baseada no nível de atividade da cena A . Especificamente, computa-se uma medida de atividade da cena A para cada pixel e reduz-se a taxa de aprendizagem α por um fator ε nos pixels onde A excede um limiar H . Com isso, variações bruscas na luminosidade de uma determinada região seriam mais rapidamente incorporadas ao modelo de cena de fundo e objetos de interesse que

não estejam completamente estáticos seriam incorporados mais lentamente ao modelo de cena de fundo. Inicialmente, para cada pixel, o nível de atividade da cena é ajustado em zero e gradativamente é computado recursivamente do nível anterior de atividade e da diferença de luminância entre o quadro atual e o passado, mostrado na equação 4.10 e adotada por Harville, Gordon e Woodfill (2001):

$$A_{i,t,k} = (1 - \lambda) A_{i,t-1,k} + \lambda |Y_{i,t} - Y_{i,t-1}|, \quad (4.10)$$

onde $Y_{i,t}$ é o valor da luminância no instante t e λ ($0 < \lambda \leq 1$) é a taxa de aprendizagem da medida de atividade, cuja função é suavizar ruídos decorrentes da luminância. É importante destacar que o valor de α não é reduzido a zero, permitindo que o modelo de cena de fundo seja sempre ajustado corretamente.

A figura 4.1a mostra um gráfico, onde os dados são plotados e representados pelos círculos azuis e as distribuições por círculos vermelhos onde se encontram três agrupamentos de elementos diferentes. A figura 4.1b mostra superfícies que representam as densidades de probabilidade.

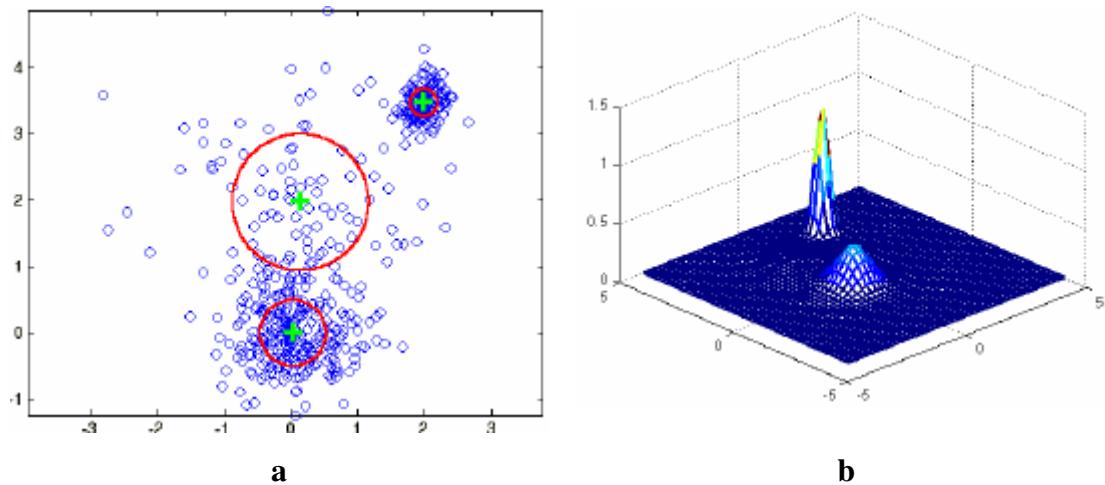


Figura 4.1: Exemplo de agrupamentos de densidades criadas usando-se GMM.

4.6 Limiar de Cor de Pele

Após o processamento pontual, os resultados são submetidos a uma etapa de detecção por cor de pele somente nos pixels considerados como sendo objetos de interesse. O método utilizado para construir este classificador de pele foi o abordado por Peer, Kovac e Solina (2003), que define empiricamente o conjunto de regras que

limitam cor de pele no espaço de cor RGB, na qual um pixel é classificado como cor de pele se as regras, mostradas na equação 3.15 (descrito no capítulo 3, seção 3.3.2), forem satisfeitas. Um exemplo de gráfico representando um agrupamento de pixels de cor de pele no espaço de cor RGB é usado na figura 4.2.

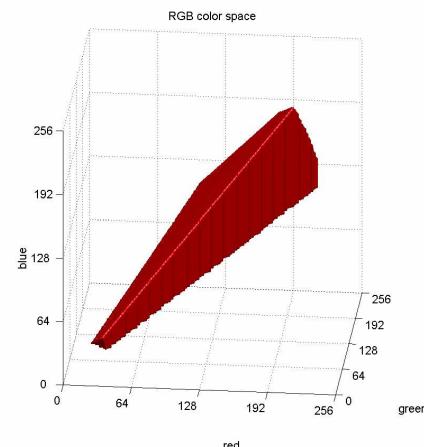


Figura 4.2: Agrupamento de pixels de cor de pele em RGB.

5 PÓS-PROCESSAMENTO

5.1 Introdução

Neste capítulo são descritas as técnicas utilizadas para recuperar as falhas resultantes da segmentação e ruídos na sua morfologia. A segmentação é realizada pelo algoritmo GMM e pelo filtro de cor de pele, conforme mostrado no capítulo 4.

Como primeiro passo são considerados os filtros morfológicos, a detecção do contorno e coleta e representação dos pontos que definem o contorno da mão.

Também será mostrado como obter o contorno mais representativo da região segmentada e que tenha a forma da mão. Finalizado o pós-processamento, podem-se iniciar as etapas de posicionamento da mão e remoção de antebraço, que serão abordadas nas seções seguintes.

Na próxima seção é apresentada uma motivação geral da detecção de contorno considerando-se a sua importância no plano deste trabalho.

5.2 Importância do Contorno

A capacidade de se reconhecer objetos em uma imagem depende muito da quantidade de informações que se conhece de cada objeto. Logo, extrair características dos objetos é uma etapa e tarefa fundamental para alcançar os objetivos no processo de reconhecimento. A extração de características depende fortemente de como os objetos são representados computacionalmente. Por este motivo, é necessário um cuidado especial na escolha da representação dos objetos de tal maneira que o processo de detecção das características possa se dar da maneira mais natural possível.

Para que o processo de reconhecimento possa ser realizado é necessário que os objetos de interesse sejam identificados e representados adequadamente.

Uma forma bastante comum de representação dos objetos e que foi utilizada neste trabalho é a representação baseada no contorno. O contorno é uma representação concisa e suficiente para capturar a morfologia do objeto.

Particularmente no caso da mão em movimento, muitas informações podem ser obtidas extraindo-se contorno a partir das imagens geradas pela segmentação da mão. Se a cada momento pode-se conhecer a contorno da mão, então pode-se saber as diferentes formas que a mão está descrevendo ao longo do tempo. Isto ajuda muito e é uma grande vantagem, principalmente para objetos que mudam a sua morfologia ao longo do tempo. Interfaces baseadas em gestos, no caso da mão, poderiam facilmente ser suportadas com uma modelagem de contorno.

A detecção de contorno é o processo que determina quais são os pontos da imagem que fazem parte da borda de uma região preenchida da imagem. Neste processo, são coletados pontos de forma a capturar a morfologia do objeto e finalmente estruturá-los de modo que possam servir como dado de entrada para as etapas posteriores do reconhecimento.

No processo de detecção de contorno são considerados três passos fundamentais. Em um primeiro passo, é feita uma abordagem de melhoria da segmentação; aqui são considerados filtros morfológicos para tentar corrigir e diminuir ao máximo os erros de segmentação e ruídos na morfologia do objeto.

No segundo passo considera-se uma abordagem para a detecção das bordas dos objetos, que está diretamente influenciada pelo passo anterior. No último passo são feitas a coleta e a estruturação dos pontos que fazem parte do contorno da mão. É fundamental que este passo seja capaz de lidar com os ruídos e erros ainda presentes após os dois passos anteriores.

5.3 Pós-processamento

Com o resultado da segmentação, tem-se uma imagem binária contendo os pixels relativos aos objetos de interesse com valor 1 (um) e os pixels relativos ao modelo de cena de fundo estimado com valor 0 (zero). Esta imagem binária resultante possui erros de segmentação e ruídos na morfologia do objeto.

Portanto, o objetivo principal nesta etapa é o refinamento da segmentação, isto é, corrigir ou pelo menos diminuir os erros e ruídos do processo de segmentação. Para isto, é utilizada uma abordagem de processamento de imagens baseada em filtros de suavização e em filtros morfológicos que visam corrigir a imagem do objeto segmentado completando pequenos buracos e eliminando regiões isoladas de poucos pixels.

Primeiramente se utiliza um filtro de suavização na imagem segmentada. Esse filtro utilizado é o Gaussiano com máscara de 5x5. A função *cvSmooth* da biblioteca de visão computacional OpenCV (INTEL, 2005) foi empregada para executar essa etapa do processo.

Os filtros morfológicos utilizados foram: o filtro de dilatação e o filtro de erosão (GONZALEZ; WOODS, 2000). Também foram empregadas as funções *cvDilate* e *cvErode* da biblioteca de visão computacional OpenCV.

A dilatação, em geral, faz com que os objetos se dilatem ou aumentem de tamanho, enquanto que a erosão faz com que eles encolham. Ambos filtros atuam nas bordas internas e externas dos objetos. A quantidade e a forma como os objetos se dilatam ou encolhem depende fortemente da escolha de uma máscara. As máscaras mais comuns são a de vizinhança 4 e de vizinhança 8.

Na aplicação dos filtros morfológicos nas imagens provenientes da segmentação, foi utilizada a máscara vizinhança 8. A seqüência de aplicação dos filtros foi: primeiro aplica-se a dilatação e depois, na imagem dilatada, aplica-se o filtro de erosão.

Com a aplicação destes filtros na seqüência indicada, procura-se em princípio com a dilatação expandir o objeto através das suas bordas internas e externas. Assim os buracos tendem a ser preenchidos e as bordas a serem expandidas uniformemente.

Depois, aplicando a erosão na imagem dilatada, procura-se retornar ao objeto original; apenas as bordas serão afetadas, e os buracos totalmente preenchidos na etapa de dilatação serão mantidos. A aplicação destes filtros nessa seqüência é conhecida também como “fechamento”.

Os resultados obtidos da aplicação desta seqüência de filtros morfológicos nas imagens de entrada são do tipo ilustrado na figura 5.1.

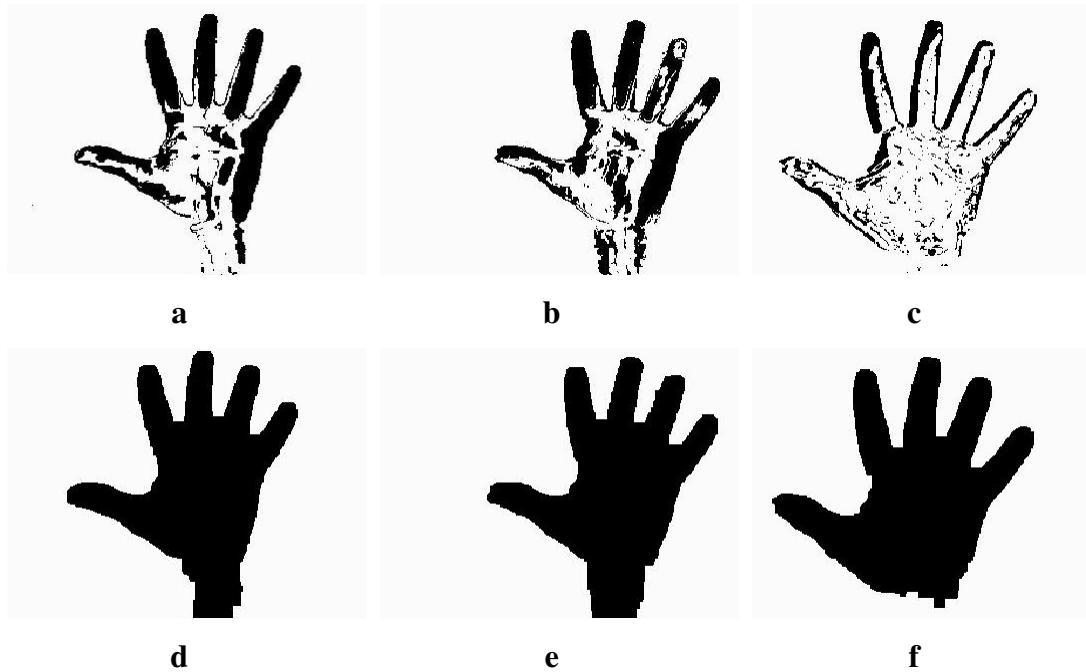


Figura 5.1: Três exemplos de imagens segmentadas com falhas e buracos (a,b,c) e depois ao pós-processamento (d,e,f respectivamente).

Na figura 5.1, nas colunas a, b, c, existem três imagens resultantes da segmentação pelo método GMM e filtro de cor de pele a partir de três quadros diferentes de uma seqüência de vídeo. Na linha inferior da figura 5.1 a coluna d, e, f, contém três imagens resultantes do pós-processamento de a, b, c respectivamente. Os buracos foram totalmente preenchidos na etapa de pós-processamento, onde foram executados filtros de suavização, dilatação, erosão e remoção de regiões segmentadas com pequena área.

5.4 Detecção do Contorno

Depois da execução do pós-processamento sobre a imagem binária da segmentação da mão, é necessário obter o contorno dessa forma de mão. Para isso, foi utilizada a função *cvFindContours* da biblioteca OpenCV que a partir de uma imagem binária cria uma lista de todos os contornos da imagem por aproximação de polígonos.

Essa lista é percorrida e somente o polígono de maior área é escolhido. Além disso, somente contornos com áreas maiores que 5% da área total da imagem são considerados

como relevantes. Assim, se não existirem contornos maiores que esse limite, o processo de reconhecimento não continua, e o sistema tenta captar o próximo quadro.

Esses vértices são armazenados para uso na etapa de remoção de antebraço do contorno da mão e no cálculo de momentos do contorno que serão mostrados nas seções seguintes. Na figura 5.2, em (a) pode-se observar a máscara de entrada da mão no processo de detecção de contornos. Em (b) todos os contornos gerados pelo detector de contornos. Em (c) apenas o maior contorno detectado da lista de contornos criada.

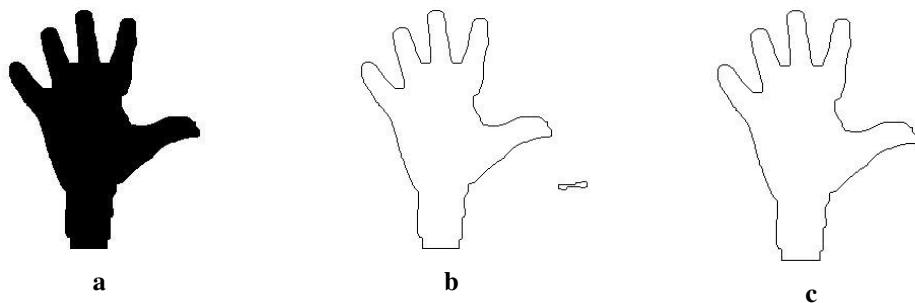


Figura 5.2: Em (a) é mostrada a imagem segmentada pós-processada. Em (b) todos os contornos gerados pelo detector de contornos. Em (c) o maior contorno somente.

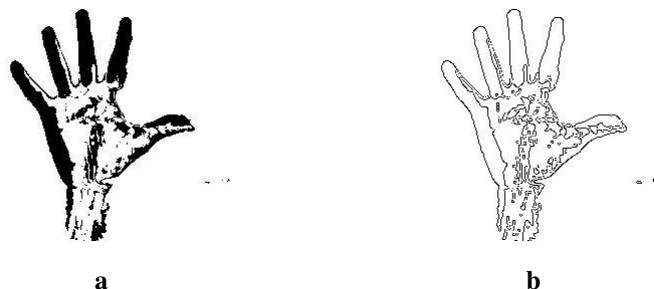


Figura 5.3: Em (a) a imagem segmentada sem pós-processamento. Em (b) a imagem com contornos da imagem segmentada sem pós-processamento.

Na figura 5.3, pode-se notar como a imagem segmentada e sem o pós-processamento (a) provoca a criação de uma quantidade de contornos muito grande, mostrados em (b), o que impossibilitaria um reconhecimento de gestos viável.

5.5 Posição e Orientação da Mão

Após corrigir as falhas presentes na segmentação da mão, já é possível utilizar algum método para calcular a posição da mesma. Considera-se como posição da mão a

posição de um ponto que se encontra no centro da palma da mão. Descobrir este ponto pode não ser uma tarefa fácil, uma vez que a mão é um objeto articulado e muda sua forma de acordo com sua orientação e a configuração dos dedos. Para identificar a posição da mão foi adotada a técnica da transformada da distância euclidiana (TDE) sobre a imagem binária da imagem segmentada, conforme será visto a seguir.

5.6 Transformada da Distância

A técnica avaliada baseia-se na Transformada da Distância Euclidiana (TDE), utilizada por Morris (2004) e Deimel (1998) para localizar o centro da palma da mão. A transformada da distância é normalmente aplicada sobre uma imagem binária, cujo resultado é outra imagem, denominada imagem de distâncias. O valor da intensidade dos pixels desta imagem é o valor da distância ao limite mais próximo do mesmo, sendo que o limite considerado pode ser o fundo da imagem ou o contorno de um objeto. Um exemplo de transformada da distância aplicada a uma imagem binária (figura 5.4a) pode ser visto na figura 5.4b, onde é possível perceber que quanto mais internos são os pixels dentro do objeto, maior é o valor de suas intensidades, ou seja, maior o valor da distância em relação ao fundo da imagem.

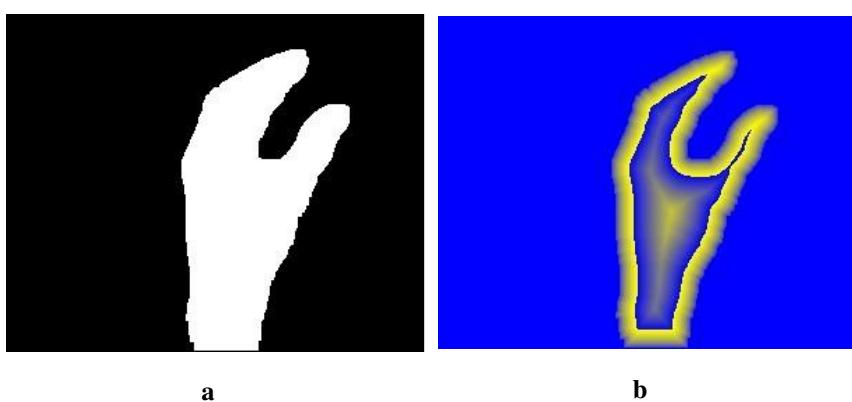


Figura 5.4: (a) Imagem da mão segmentada. (b) Imagem gerada pela aplicação da transformada da distância na imagem segmentada.

Para determinar o centro da mão, foi implementado um método que utiliza a imagem segmentada da mão e calcula a transformada da distância dos pontos internos à mão com relação ao fundo. Para essa tarefa, foi usada a função *cvDistTransform* da

biblioteca de visão computacional OpenCV (INTEL, 2005). Essa função retorna a imagem de distâncias de intensidade dos pixels e assim se obtém a distância máxima e as coordenadas desse pixel, que determina o centro da palma da mão (figura 5.5).

A circunferência na palma da mão mostrada na figura 5.5 representa a maior distância detectada pela transformada da distância. Essa circunferência tem o centro no pixel com maior distância da imagem em relação ao fundo e o raio possui o valor dessa distância. Essa circunferência determina aproximadamente a região compreendida pela palma da mão. Essa informação também será usada na remoção do pulso do restante do contorno da mão para extração de características mais precisas do que mantendo o pulso.



Figura 5.5: Posição da mão
determinada pelo centro da palma e
calculado com TDE .

5.7 Influência do Antebraço na Posição da Mão

Foi verificado em alguns testes que a presença do antebraço ou pulso na cena prejudica o cálculo da orientação da palma da mão, pois a orientação do eixo principal da elipse é influenciada pela orientação do antebraço, uma vez que a área desse também é utilizada no cálculo dos momentos de imagem.

Na figura 5.6 pode-se notar a influência do antebraço no cálculo da orientação, através da observação da orientação do maior semi-eixo da elipse, que não corresponde à orientação da palma da mão. Devido a este fenômeno, adotou-se a condição de que somente a mão deva aparecer na cena.

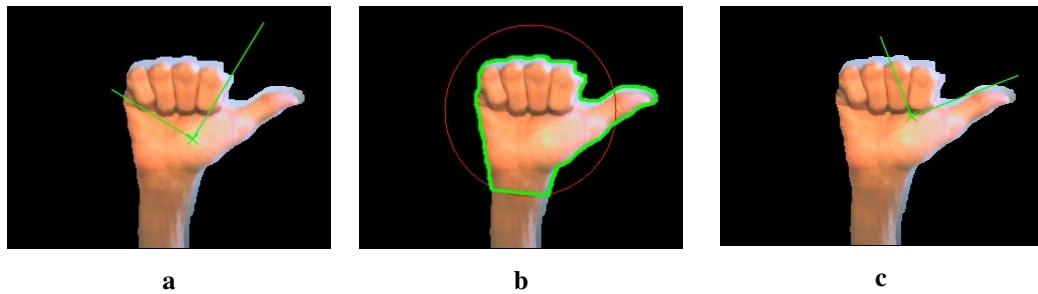


Figura 5.6: (a) Influência do antebraço no cálculo da orientação dos semi-eixos da elipse. (b) Contorno mostrando que o pulso foi removido. (c) Nova orientação dos semi-eixos da elipse após a remoção do pulso.

Em seguida na figura 5.7 tem-se mais um exemplo que o antebraço influencia também no comprimento do maior semi-eixo da elipse. Note que sem a remoção do antebraço o comprimento do maior semi-eixo (a) é maior do que após a remoção (c). Em (b) é mostrado o novo contorno da mão após a remoção do antebraço.

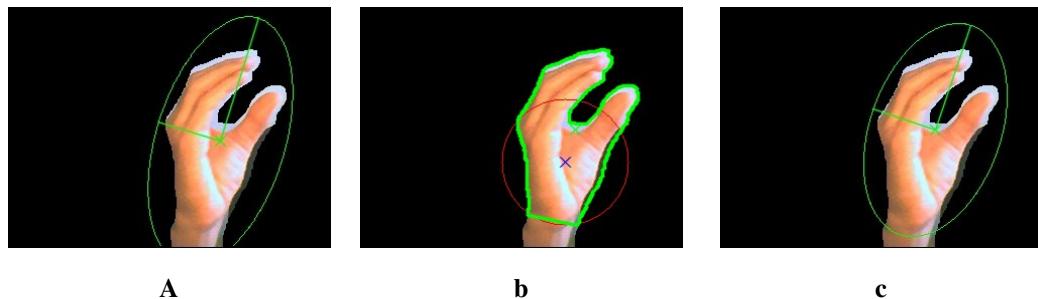


Figura 5.7: (a) Orientação e comprimento dos semi-eixos da elipse calculada pelos momentos do contorno. (b) Contorno mostrando que o pulso foi removido. (c) Orientação e comprimento dos semi-eixos da elipse modificados após a remoção do pulso.

5.8 Remoção do Antebraço

Baseado no artigo de Deimel e Schröter (1999) uma solução de remoção de antebraço foi implementada com simplificações e modificações do método original. Os autores usam a Transformada da Distância Euclidiana que é calculada sobre a imagem binarizada da segmentação da mão. O algoritmo adotado é detalhado no capítulo de Metodologia (capítulo 7, seção 7.2.3.2). A figura 5.8a mostra o contorno da mão sem a

remoção do antebraço e com o círculo da palma da mão e seu *offset* que determina a região a ser retirada. Já a figura 5.8b exibe o mesmo contorno sem o antebraço.

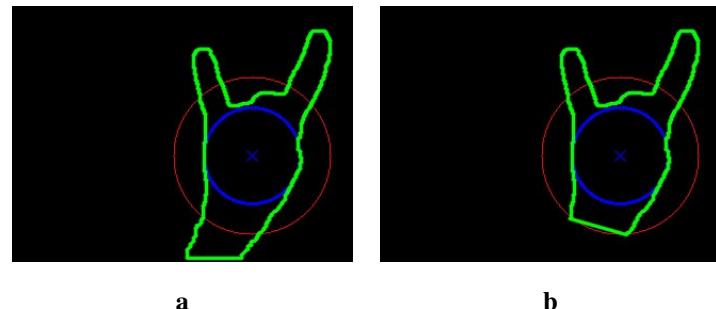


Figura 5.8: Em (a) contorno sem a remoção do antebraço e com as circunferências da palma(azul) e de *offset* (vermelho). Em (b) Contorno mostrando que o pulso foi removido exatamente nas intersecções com circunferência *offset*.

6 RECONHECIMENTO DE GESTOS

6.1 Introdução

A partir do momento em que se obtém o contorno através da aplicação da segmentação e do pós-processamento da imagem, surge a necessidade de efetuar-se o reconhecimento da pose da mão determinada por este contorno. No decorrer deste capítulo é proposto um modelo para o reconhecimento de gestos de mão baseado no seu contorno. Além dos gestos, são detectadas a posição, a orientação da mão e uma série de características relevantes.

Para tal, foi desenvolvido um subsistema capaz de realizar comparações entre um contorno de mão qualquer, definida por um conjunto de pontos (polígono que representa o contorno da mão), e padrões previamente estabelecidos e finalmente alguns gestos pré-estabelecidos são reconhecidos. A saída deste módulo é o padrão que melhor se ajusta ao contorno dado.

6.2 Casamento de Padrões

As técnicas computacionais para realização de casamento de padrões variam consideravelmente, desde o uso de classificadores baseados em menor distância ao emprego de sofisticadas redes neurais. Neste trabalho foi utilizada uma técnica de reconhecimento de padrões baseada em momentos.

A teoria dos momentos data-se anterior a Newton, mas sua aplicação a casamentos de padrões não era conhecida até a década de setenta, quando o estudante de graduação Sigfried descobriu, durante suas pesquisas para a tese de mestrado, que este método poderia ser aplicado ao reconhecimento de padrões, dada a sua expressividade em relação à representação de características de objetos (TOUSSAINT, 1994).

No presente trabalho, a partir do contorno dado, é montado um vetor onde cada componente é um momento do contorno. Estes momentos têm por objetivo representar

o contorno em um certo espaço de dimensão menor e são usados como parâmetros dos vetores de características para classificar o padrão de entrada entre um conjunto de classes. Estas classes estão, por sua vez, também caracterizadas através de vetores de características.

Uma das dificuldades da utilização desta técnica é a determinação daqueles momentos mais adequados ao problema, já que existe um número infinito deles que podem ser computados. Como em todo problema de reconhecimento de padrões, é importante determinar um conjunto reduzido de características que, simultaneamente, possuam as seguintes propriedades (TOUSSAINT, 1994):

- Para dois objetos diferentes, o casamento entre eles seja quantitativamente pobre.
- Para dois objetos similares, o casamento entre eles seja quantitativamente bom.
- Que o cálculo dos padrões seja computacionalmente atraente, isto é, relativamente barato e estável perante ruídos, rotações, mudanças de escala e outras possíveis fontes de perturbação.

6.3 Momentos de Imagem

Os momentos de imagem permitem calcular determinadas propriedades geométricas de objetos presentes em uma imagem como posição, tamanho e orientação. Tais propriedades são interessantes na implementação de propostas de reconhecimento de gestos, rastreamento de objetos e extração de características para reconhecimento de padrões (HECKENBERG, 1999) (CHAUMETTE, 2004) (LIU; LOVELL, 2001), onde há a necessidade de encontrar descritores de objetos que sejam invariantes com relação à translação, rotação e escala (GONZALEZ, 1977) (LIAO, 1993).

Para uma função bidimensional $f(x,y)$ os momentos de ordem $(p+q)$ são dados pela equação 6.1:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy , \quad (6.1)$$

desde que a integral exista.

Aplicando-se a equação 6.1 sobre um objeto binário numa imagem digital, pode-se reapresentá-la pela equação 6.2:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y). \quad (6.2)$$

Para um objeto binário de imagem digital, a função $f(x,y)$ será igual a 1 para todos os pixels que pertencem ao objeto, portanto pode-se simplificar a equação 6.2 conforme a equação 6.3:

$$m_{pq} = \sum_A x^p y^q, \quad (6.3)$$

onde A é a área do objeto, ou simplesmente o número de pixels que o compõe, e corresponde ao momento de ordem zero M_{00} ($(p,q) = (0,0)$), que pode ser calculada pela equação 6.4.

$$m_{00} = \sum_x \sum_y f(x, y). \quad (6.4)$$

Os momentos de primeira ordem ($(p,q) = (1,0)$ ou $(0,1)$) são calculados pelas equações 6.5 e 6.6 e contém informações sobre o centro de gravidade do objeto que são determinados pelas equações 6.11 e 6.12.

$$m_{10} = \sum_x \sum_y x f(x, y). \quad (6.5)$$

$$m_{01} = \sum_x \sum_y y f(x, y). \quad (6.6)$$

Os momentos de segunda ordem ($(p,q) = (2,0)$ ou $(0,2)$ ou $(1,1)$) são calculados pelas equações de 6.7 a 6.9 e contém informações sobre os momentos de inércia em relação ao eixo vertical e em relação ao eixo horizontal e aos dois.

$$m_{20} = \sum_x \sum_y x^2 f(x, y). \quad (6.7)$$

$$m_{02} = \sum_x \sum_y y^2 f(x, y). \quad (6.8)$$

$$m_{11} = \sum_x \sum_y x^2 y^2 f(x, y). \quad (6.9)$$

Como $f(x,y)$ é o valor da intensidade da luminância (nível de cinza) de um pixel na imagem, na localização (x,y) , a medida m_{pq} varia se f sofre uma translação. Para tornar m_{pq} invariante em relação à translação de f , usam-se os momentos centrais de ordem $(p+q)$ da imagem f , sendo a equação 6.2 reescrita como a equação 6.10:

$$u_{pq} = \sum_x \sum_y (x - x')^p (y - y')^q f(x, y), \quad (6.10)$$

onde x' e y' são as coordenadas do centróide encontradas através das equações 6.11 e 6.12 respectivamente, que em última análise correspondem às equações do centro de massa.

$$x' = \frac{\sum_x \sum_y x f(x, y)}{\sum_x \sum_y f(x, y)} = \frac{m_{10}}{m_{00}} \quad (6.11)$$

$$y' = \frac{\sum_x \sum_y y f(x, y)}{\sum_x \sum_y f(x, y)} = \frac{m_{01}}{m_{00}} \quad (6.12)$$

Estes momentos centrais ainda variam com relação à rotação. Para que os momentos de segunda e terceira ordem sejam também invariantes à rotação, além da translação, são então definidos pelas equações de 6.13 a 6.19:

$$a_1 = u_{20} + u_{02} \quad (6.13)$$

$$a_2 = (u_{20} - u_{02})^2 + 4u_{11}^2 \quad (6.14)$$

$$a_3 = (u_{30} - 3u_{12})^2 + (3u_{21} - u_{03})^2 \quad (6.15)$$

$$a_4 = (u_{30} + u_{12})^2 + (u_{21} + u_{03})^2 \quad (6.16)$$

$$a_5 = (u_{30} - 3u_{12})(u_{30} + u_{12})[(u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2] \\ + 3(u_{21} - u_{03})(u_{21} + u_{03})[3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2] \quad (6.17)$$

$$a_6 = (u_{20} - u_{02})[(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2] \\ + 4u_{11}(u_{30} + u_{12})(u_{21} + u_{03}) \quad (6.18)$$

$$a_7 = (3u_{21} - u_{03})(u_{30} + u_{12})[(u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2] \\ + (u_{30} - 3u_{12})(u_{21} + u_{03})[3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2] \quad (6.19)$$

Para ilustrar melhor o papel dos momentos na classificação de padrões, são descritos os significados de alguns deles:

- m_{00} : Representa o numero de elementos do objeto sobre a grade digital.
- m_{20} : Determina o momento de inércia em relação ao eixo vertical.
- m_{02} : Determina o momento de inércia em relação ao eixo horizontal.
- m_{03} : Estabelece o grau de simetria do objeto em torno do eixo horizontal.

6.4 Gestos Estáticos e Dinâmicos

Nos dois capítulos anteriores houve o enfoque em localizar as regiões de interesse nas imagens analisadas através da segmentação. A próxima etapa é observar mais de perto essas regiões e encontrar algumas características que permitam extrair informações de relevância relacionadas à posição da mão e do gesto que está sendo mostrado.

Existem dois tipos de gestos de mão que podem ser reconhecidos:

- Os gestos estáticos são aqueles que mostram apenas uma determinada postura da mão e o significado se traduz na forma ou postura que apresentam. Para os gestos estáticos é importante apenas saber qual é o gesto da mão em cada quadro, independentemente do tempo ou do número de quadros.
- Os gestos dinâmicos são aqueles em que seu significado depende da postura e do movimento que a mão descreve. Para os gestos dinâmicos é importante analisar o comportamento da mão ao longo do tempo.

Os gestos dinâmicos podem ser reconhecidos a partir de um conjunto de gestos estáticos. Como exemplo, a detecção de movimento poderia acontecer a partir de um conjunto de gestos estáticos ou a partir do comportamento dos gestos estáticos em um certo período de tempo.

6.5 Gestos Usados Como Padrão

Para esse trabalho, foi definido que o reconhecimento dos gestos deve ser feito baseado na forma que um contorno da mão apresenta em um determinado instante. Para isso, é necessário determinar com clareza a forma presente e também suas características, tais como momentos invariantes de *Hu* (HU, 1962), sua posição, orientação, os pontos que representam o contorno e várias características calculadas a

partir dessas propriedades. O objetivo desta fase é construir um módulo para o reconhecimento de alguns gestos básicos. As aplicações podem utilizar esses gestos básicos de forma particular e de acordo com suas necessidades.

São 10 (dez) o número de gestos que podem ser construídos e reconhecidos neste trabalho (figura 6.1). Esses gestos devem ser estáticos e bidimensionais (2D) e mais básicos possíveis, para que possam servir na construção ou reconhecimento de tipos de gestos mais complexos por outros módulos do sistema ou mesmo por outras aplicações.

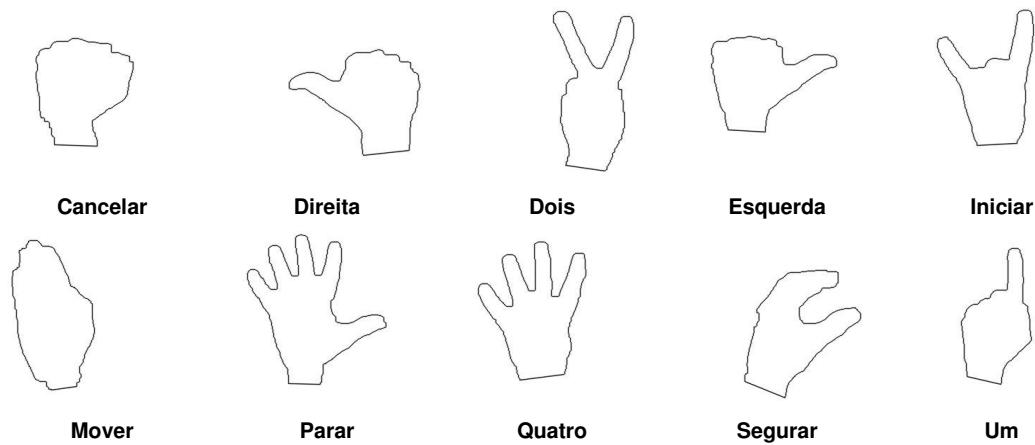


Figura 6.1: Imagens contendo os 10 contornos de gestos de mão armazenados para casamento de padrões.

6.6 Extração de Características

Como foi apresentado no segundo capítulo não existe uma única abordagem para reconhecer gestos de mão em imagens. Algumas técnicas utilizam forma, outras forma e cor, e outras utilizam ainda transformadas para o reconhecimento de gestos da mão. Nesse presente trabalho faz-se uso de casamento de padrões utilizando-se vetores de características formados com momentos invariantes de *Hu* (HU, 1962) e alguns parâmetros determinados a partir dos momentos da imagem.

Os momentos são calculados empregando-se o centro de gravidade do objeto como a origem dos eixos de coordenadas. Dessa forma, elimina-se o problema de posicionamento da curva em relação aos eixos. Calculados os momentos do contorno pode-se determinar as características para a montagem do vetor. Nossa abordagem

utilizará alguns parâmetros relevantes da imagem para criação do vetor de características. Portanto, as imagens de contornos são representadas por esses vetores e são utilizados no classificador para o casamento de padrões (TEAGUE, 1980) (COSTA; CESAR, 2001). O vetor de características é formado por 14 parâmetros: os momentos invariantes de *Hu* (HU, 1962), o ângulo de orientação do semi-eixo menor da elipse, a circularidade, a excentricidade, o raio de giro, a dispersão e o máximo e mínimo momento de inércia.

6.6.1 Momentos Geométricos ou Invariantes

Os momentos são denominados por momentos geométricos ou invariantes, devido às suas propriedades de insensibilidade a transformações geométricas como a translação, rotação, mudança de escala e ponto inicial (HU, 1962). Os parâmetros de forma baseados nos momentos invariantes têm como objetivo representar o contorno fechado de um objeto simples ou região através das suas propriedades estatísticas. Normalmente, os momentos invariantes calculados são sete e baseiam-se nos momentos centrais de segunda e terceira ordem. A razão pela qual são calculados estes sete momentos invariantes, e nem mais nem menos, deve-se ao fato de a partir de certa ordem não se obter mais detalhe do que aquele que é necessário e distingível pelo ser humano.

Dessa forma, os sete momentos invariantes são calculados do seguinte modo:

- Normalização dos momentos centrais u_{pq} , gerando-se momentos centrais normalizados M_{pq} através das equações de 6.20 a 6.22:

$$M_{pq} = \frac{u_{pq}}{u_{00}^{\gamma}} \text{ em que } \gamma = \frac{1}{2}(p+q)+1 \text{ e } (p+q) = 2,3,\dots \quad (6.20)$$

$$M_{11} = \frac{u_{11}}{u_{00}^2}; M_{20} = \frac{u_{20}}{u_{00}^2}; M_{02} = \frac{u_{02}}{u_{00}^2}; \quad (6.21)$$

$$M_{21} = \frac{u_{21}}{u_{00}^{3/2}}; M_{12} = \frac{u_{12}}{u_{00}^{3/2}}; M_{30} = \frac{u_{30}}{u_{00}^{3/2}}; M_{03} = \frac{u_{03}}{u_{00}^{3/2}} \quad (6.22)$$

- Cálculo dos momentos invariantes, $A1\dots A7$, através das equações de 6.23 a 6.29, usando os valores M_{pq} obtidos das equações de 6.20 a 6.22:

$$A_1 = M_{20} + M_{02} \quad (6.23)$$

$$A_2 = (M_{20} - M_{02})^2 + 4M_{11}^2 \quad (6.24)$$

$$A_3 = (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2 \quad (6.25)$$

$$A_4 = (M_{30} + M_{12})^2 + (M_{21} + M_{03})^2 \quad (6.26)$$

$$A_5 = (M_{30} - 3M_{12})(M_{30} + M_{12}) \left[(M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2 \right] + \\ 3(M_{21} - M_{03})(M_{21} + M_{03}) \left[3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2 \right] \quad (6.27)$$

$$A_6 = (M_{20} - m_{02}) \left[(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2 \right] + \\ 4M_{11}(M_{30} + M_{12})(M_{21} + M_{03}) \quad (6.28)$$

$$A_7 = (3M_{21} - M_{03})(M_{30} + M_{12}) \left[(M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2 \right] + \\ (M_{30} - 3M_{12})(M_{21} + M_{03}) \left[3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2 \right] \quad (6.29)$$

6.6.2 Orientação

Para facilitar a análise de objetos que possui um formato complexo, pode-se utilizar uma elipse equivalente que melhor descreve sua forma, como ilustrado nas figuras 6.2 e 6.3 (COSTA; CESAR, 2001). A elipse equivalente permite determinar algumas das propriedades do objeto em análise, como sua posição de centro. O centro da elipse corresponde ao centróide do objeto. Além da posição, a orientação do objeto pode ser obtida através da orientação da elipse. O centróide é determinado por $x_c = m_{10}/m_{00}$ e $y_c = m_{01}/m_{00}$.

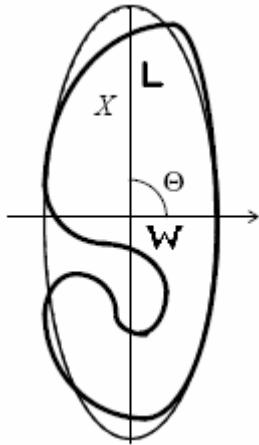


Figura 6.2: Elipse equivalente que descreve a posição e a orientação de um contorno, mostrando o semi-eixo maior L , o semi-eixo menor W e a orientação θ .

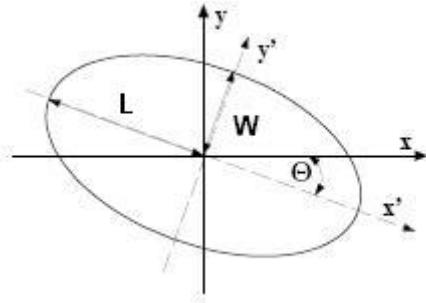


Figura 6.3: Elipse da imagem mostrando o semi-eixo maior L , o semi-eixo menor W e a orientação θ .

Para determinar-se a orientação de um objeto, considera-se a orientação do maior eixo principal da elipse equivalente. A orientação deste eixo pode ser calculada pela equação 6.30 (PROKOP; REEVES, 1992).

$$\theta = \frac{\arctan\left(\frac{b}{(a - c)}\right)}{2}, \quad (6.30)$$

onde θ corresponde ao menor ângulo entre o maior eixo e a horizontal e, a , b e c são computados pelas equações 6.31 a 6.33:

$$a = \frac{m_{20}}{m_{00}} - y_c^2. \quad (6.31)$$

$$b = 2\left(\frac{m_{11}}{m_{00}} - x_c y_c\right). \quad (6.32)$$

$$c = \frac{m_{02}}{m_{00}} - y_c^2. \quad (6.33)$$

6.6.3 Comprimentos dos Semi-Eixos da Elipse

Os comprimentos dos semi-eixos principais da elipse podem ser calculados através das seguintes equações 6.34 e 6.35 (COSTA; CESAR, 2001):

$$W = \sqrt{6(a + c - \sqrt{b^2 + (a - c)^2})}, \quad (6.34)$$

$$L = \sqrt{6(a + c + \sqrt{b^2 + (a - c)^2})}, \quad (6.35)$$

onde W corresponde ao menor semi-eixo e L corresponde ao maior semi-eixo.

6.6.4 Circularidade

A circularidade (*circularity*) de um objeto simples ou de uma região traduz a sua semelhança em relação a uma circunferência (DAVIS, 1986). Um objeto simples ou região é tão mais circular quanto mais próximo da unidade for o valor da sua circularidade normalizada a 1. A equação 6.36 define a circularidade:

$$Circ = \frac{4\pi A}{P^2}, \quad (6.36)$$

onde P é o perímetro da forma a descrever e A é a área do objeto simples ou região. Este parâmetro tem como vantagens a insensibilidade a rotações, translações, mudanças de escala e ponto inicial. A forma da figura 6.4a tem como circunferência equivalente à sua circularidade, isto é, circunferência com a mesma área da forma, a circunferência em preto na figura 6.4b.



Figura 6.4: a) Forma do objeto. b) Representação gráfica da circularidade do objeto.

6.6.5 Excentricidade

A excentricidade (*eccentricity*) de um objeto simples ou região é definida como a relação existente entre o seu raio máximo, $L=R_{max}$, e o seu raio mínimo, $W=R_{min}$. Os vários raios são definidos como a distância entre o centróide do objeto e qualquer ponto da menor elipse envolvente do objeto e centrada no centróide do objeto. A excentricidade também pode ser definida em termos dos parâmetros a e b definidos

anteriormente na seção 6.6.2. A equação 6.37 define a excentricidade (BALLARD; BROWN, 1982):

$$\epsilon = \frac{R_{\max}}{R_{\min}} = \frac{L}{W} = \sqrt{\frac{a^2 - b^2}{a^2}}. \quad (6.37)$$

Este parâmetro tem como vantagens a insensibilidade a rotações, translações, mudanças de escala e ponto inicial. A forma da figura 6.5a tem como raio máximo e mínimo as linhas traçadas em preto na figura 6.5b.

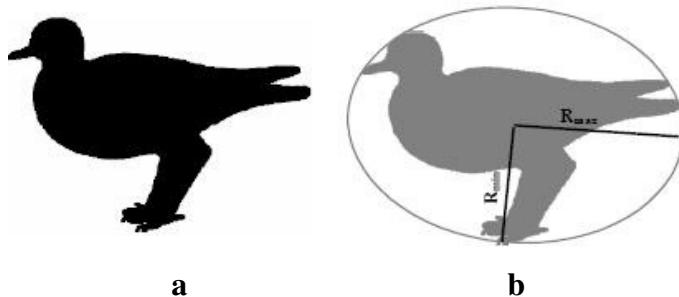


Figura 6.5: a) Forma do objeto. b) Representação gráfica dos raios máximo e mínimo do objeto

6.6.6 Raio de Giro

O raio de giro (*gyration radius*) é definido como o raio de um círculo onde toda a massa de um objeto estaria concentrada sem alterar o momento de inércia do seu centro de massa. Pode ser calculado a partir dos momentos centrais de ordens zero e de ordem dois, como mostrado na equação 6.38 (GROSBERG; KHOKHLOV, 1994).

$$R = \sqrt{\frac{\mu_{20} - \mu_{02}}{\mu_{00}}}. \quad (6.38)$$

6.6.7 Dispersão

Dispersão de um objeto simples ou região (*spreadness*) é definida com a medida da forma que mostra o quanto um objeto está espalhado (disperso) na imagem. Pode ser calculado em termos de momentos centrais pela equação 6.39 (BOESCH, 1993).

$$S = \frac{\mu_{20} + \mu_{02}}{\mu_{00}^2}. \quad (6.39)$$

6.6.8 Maior e Menor Momentos de Inércia

O momento de inércia de um corpo com relação aos eixos do sistema de referência é a soma dos produtos da massa de cada componente do corpo pela distância deste componente ao eixo de rotação. Momento inercial quantifica a resistência de um objeto em relação à aceleração angular. Em termos de momentos centrais o maior momento de inércia pode ser definido pela equação 6.40 e o menor momento de inércia pela equação 6.41. Juntos são chamados de principais momentos de inércia (*principal moments of inertia*) (COSTA; CESAR, 2001) (PROKOP; REEVES, 1992).

$$MaiorI = (\mu_{20} + \mu_{02}) + \sqrt{(\mu_{20} - \mu_{02}) + 4 * \mu_{11}^2} . \quad (6.40)$$

$$MenorI = (\mu_{20} + \mu_{02}) - \sqrt{(\mu_{20} - \mu_{02}) + 4 * \mu_{11}^2} . \quad (6.41)$$

6.7 Classificação

Para realizar a classificação do padrão é utilizada uma métrica baseada na discrepância entre os vetores, onde as diferenças entre estes vetores de características são calculadas. A distância euclidiana é usada como métrica entre o vetor do quadro corrente e os vetores padrão.

Para diminuir a influência da variação dos valores de vetores de características no processo de casamento de padrões, os valores de vetores precisam ser normalizados e só então a diferença total é calculada. Como é desejado que cada valor tenha o mesmo peso ou importância no cálculo das distâncias entre os objetos, necessita-se normalizar cada variável para que cada uma delas tenha aproximadamente o mesmo valor em escala.

Utilizando-se dos exemplos mostrado na tabela 6.1, sejam i e k o contorno e os padrões de referência, respectivamente, e $1 < k < N$. O contorno e cada padrão são descritos pelos vetores $vi = \{x_{i1}, x_{i2}, \dots, x_{ip}\}$ e $vk = \{x_{k1}, x_{k2}, \dots, x_{kp}\}$, respectivamente. A diferença entre o contorno i e o padrão k , é computada pela distância euclidiana d_{ik} , dada pela equação 6.46.

Para que os momentos de mais alta ordem não dominem o processo de classificação, estas diferenças são padronizadas e só então a diferença total é calculada. A normalização de cada valor é feita subtraindo esse pela média dos valores do vetor e

dividindo pelo desvio padrão desses mesmos valores. Especificamente, cada valor real x_{ij} do vetor de características deve ser transformado no valor normalizado z_{ij} usando a equação 6.43 (YANG; TREWN, 2004):

Contornos	Características				
	1	2	3	...	p
1	x_{11}	x_{12}	x_{13}	...	x_{1p}
2	x_{21}	x_{22}	x_{23}	...	x_{2p}
:	:	:	:	...	:
N	x_{N1}	x_{N2}	x_{N3}	...	x_{Np}

Tabela 6.1: Exemplo de vetores de características.

$$d_{ik} = \sqrt{\sum_{j=1}^p (x_{ij} - \bar{x}_j)^2} \quad (6.42)$$

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}, \quad (6.43)$$

onde a média \bar{x}_j é determinada pela equação 6.44 e o desvio padrão s_j pela equação 6.45.

$$\bar{x}_j = \frac{\sum_{k=1}^N x_{kj}}{N} \quad (6.44)$$

$$s_j = \sqrt{\frac{\sum_{k=1}^N (x_{kj} - \bar{x}_j)^2}{N-1}} \quad (6.45)$$

Após a normalização a distância euclidiana será calculada sobre os valores normalizados dos vetores, como na equação 6.46 (YANG; TREWN, 2004).

$$d_{ik} = \sqrt{\sum_{j=1}^p (z_{ij} - z_{kj})^2} \quad (6.46)$$

6.8 Algoritmo de Classificação

O algoritmo de classificação recebe como entrada um contorno de consulta com classificação desconhecida e calcula a sua classe. Seu pseudocódigo é descrito a seguir:

Pseudocódigo do Algoritmo:

1. $D_{min} = \text{INFINITO}$
2. $\text{Classe} = \text{NENHUMA}$
3. $D_{limite1} = \text{LIMITE1}$ e $D_{limite2} = \text{LIMITE2}$
4. Para cada vetor de característica padrão de classe, c , dado um vetor de características x desconhecido:
 5. $D_{hu} = \text{DistânciaEuclidiana}(x[1..7])$ (para os 7 momentos invariantes de Hu)
 6. Se $D_{hu} < D_{limite1}$ então
 7. Se $(\text{Orientação} - \text{OrientaçãoPadrão}) \leq 5^\circ$ então
 8. $D = \text{DistânciaEuclidiana}(x)$ (para todos os parâmetros de x)
 9. Se $D < D_{limite2}$ e $(D + D_{hu}) < D_{min}$ então
 10. $D_{min} = D;$
 11. $\text{Classe} = c$
 12. Final de 9.
 13. Final de 7.
 14. Final de 6.
 15. Final de 4.
 16. $\text{Classificação} = \text{Classe}$

Depois de calculados as 14 características do vetor do contorno, faz-se uma checagem, em um primeiro estágio, onde são utilizados somente os 7 momentos invariantes de Hu . O vetor é pré-classificado quando possuir a distância euclidiana menor que um limiar de distância. Checa-se também a orientação do maior semi-eixo que não deve variar mais que 5 graus. Se o vetor corrente for aceito no primeiro estágio e ficou dentro da faixa de inclinação, todos os parâmetros do vetor são utilizados no cálculo da distância euclidiana para determinar se o vetor é menor que um limiar de erro da distância. O vetor escolhido terá a menor distância euclidiana entre os vetores classificados.

7 METODOLOGIA, RESULTADOS E CONCLUSÕES

7.1 Introdução

No presente trabalho é proposto um sistema base para o reconhecimento de alguns gestos simples da mão que possam ser utilizados para reconhecer gestos mais complexos em ambientes de trabalho convencionais. Este sistema se apresenta como sendo uma camada de baixo nível que garante o reconhecimento de um conjunto de gestos básicos para outras aplicações, as quais podem utilizar esses gestos de acordo com suas necessidades. O sistema proposto neste trabalho possui as características dos sistemas baseados na detecção de pose de mãos cujo objetivo principal é a detecção e rastreamento da mão.

Com base na mão detectada e em suas informações (vetor de características) pode-se reconhecer também um conjunto básico de gestos. Assim, na interação, tanto os gestos reconhecidos quanto as informações da mão são utilizados. Para detectar-se a mão emprega-se uma abordagem baseada em contornos fechados sendo necessária a extração do contorno da mão para determinar-se a pose.

A detecção do contorno é baseada numa abordagem de segmentação que considera as características da cor e da iluminação do ambiente onde a mão está se movimentando. Nessa etapa de segmentação é introduzida uma abordagem de segmentação a partir de subtração do fundo que, além de segmentar o objeto de interesse, procura diminuir as restrições do ambiente e a influência da iluminação na modelagem do fundo. O processo deve ser feito a partir de imagens de arquivos de vídeo ou diretamente do sinal de vídeo de uma *webcam* para que ocorra a segmentação de uma só mão humana visando à aplicação em IUC.

7.2 Metodologia

As imagens de entrada do sistema são obtidas a partir de uma câmera (*webcam*). Com a seqüência de imagens em vídeo, têm-se as imagens quadro a quadro que serão trabalhadas e analisadas para o devido pré-processamento.

Em cada quadro aplicar-se um filtro Gaussiano de máscara 5x5, para suavizar a imagem. Para a segmentação da mão, são utilizados os processos de subtração do fundo com mistura de Gaussianas (STAUFFER; GRIMSON, 1999) e detecção de cor de pele com métodos classificadores definidos por regras empíricas (PEER; KOVAC; SOLINA, 2003).

Após a segmentação, é realizado um pós-processamento para se obter melhor qualidade na região segmentada através de técnicas para recuperar as falhas resultantes da segmentação e ruídos na sua morfologia. Nessa etapa também ocorre a detecção do contorno, coleta e representação dos pontos que definem o contorno da mão. A maior área de contorno será considerada como região da mão e promove-se a remoção do punho do restante do contorno da mão devido à interferência que o punho causa nos cálculos de características desse contorno.

As regiões segmentadas são analisadas para determinar a posição e a orientação da mão. Essa posição e outros atributos das mãos (vetor de características) são rastreados quadro a quadro (*tracking*). Isso é feito para distinguir um movimento da mão em relação ao fundo e de outros objetos em movimento, e para extrair a informação do movimento para o reconhecimento de gestos. Baseados na posição coletada são calculados o movimento e indícios de postura de um gesto significativo. Essas informações do vetor de características são casadas com 10 vetores de características padrão para detecção ou não de uma dessas classes de contorno. Caso alguma seqüência de gestos seja encontrada em uma ordem pré-definida no sistema, pode ser considerado um gesto dinâmico a partir dessa seqüência de gestos estáticos.

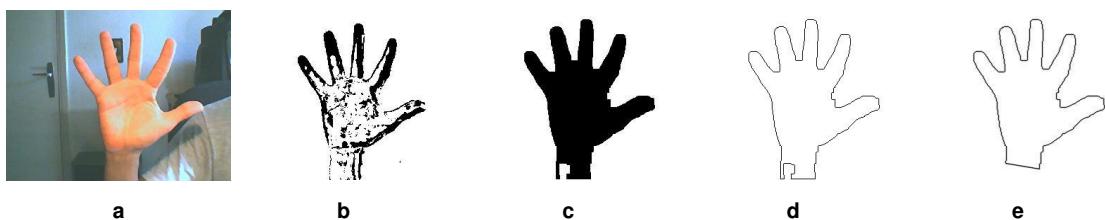


Figura 7.1: (a) Imagem de vídeo original. (b) Segmentação da imagem original usando GMM e filtro de cor de pele. (c) Pós-processamento. (d) Extração do contorno da mão. (e) Remoção do punho da mão.

Na figura 7.1, mostra-se a ordem do processamento da imagem no modelo proposto: captura da imagem, segmentação, pós-processamento, extração de contorno e remoção do pulso. Na figura 7.2, é apresentada a etapa de reconhecimento do contorno obtido no casamento com as classes de contornos padrão.

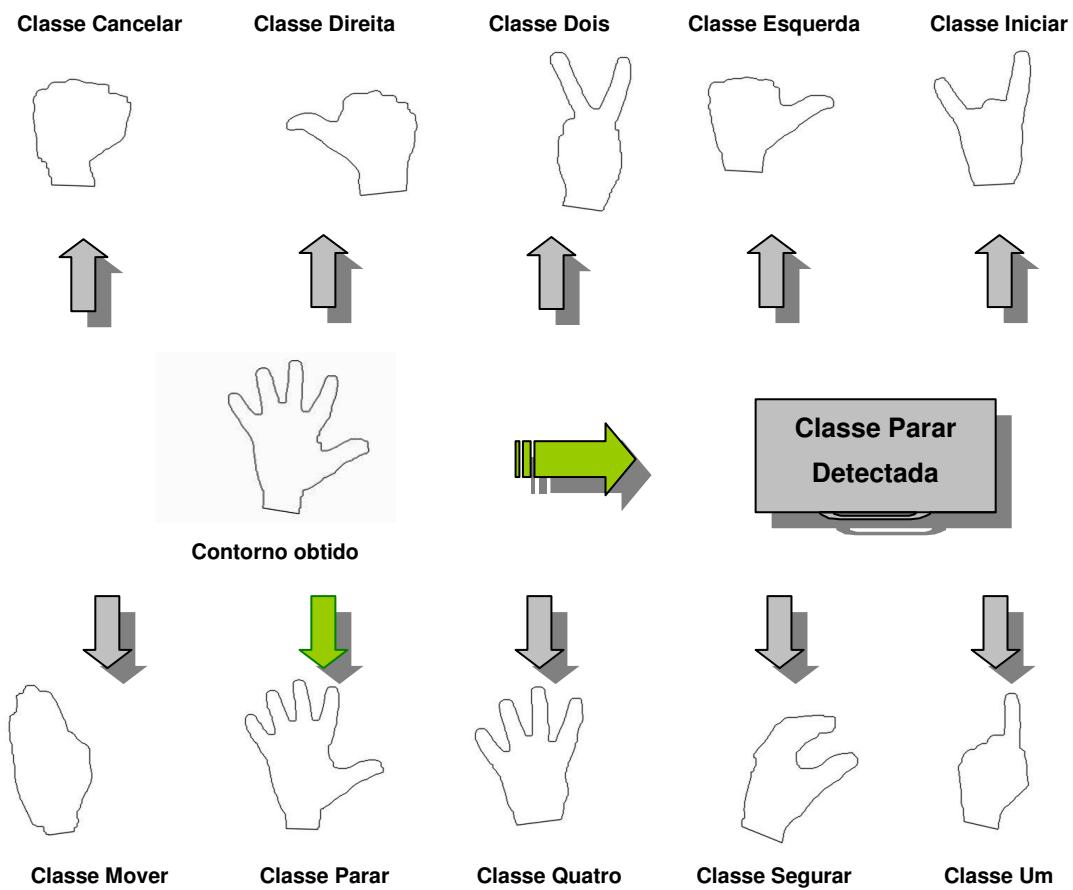


Figura 7.2: Representação do casamento de padrões entre os 10 contornos de classes de gestos de mão armazenados e o contorno obtido, detectando-se a classe Parar.

Nas próximas seções serão descritas cada etapa da metodologia mais detalhadamente.

7.2.1 Aquisição da Seqüência de Imagens

Na implementação, a aquisição das imagens pode ser feita por uma *webcam* via canal USB ou por um arquivo de vídeo já criado, no padrão AVI. Para a análise dos resultados considere a aquisição via *webcam* como sendo a forma de tratamento em tempo real, que é objetivo do trabalho. Os arquivos de vídeos são usados para gerarem dados para exemplificarem resultados e conclusões. O pré-processamento será feito quadro a quadro na seqüência de imagem.

7.2.2 Segmentação da Imagem

Após a captura de um quadro da seqüência de imagens, foi utilizado um filtro Gaussiano com máscara 5x5 para suavizar a imagem. Para a segmentação da mão, é utilizado o método de segmentação pela subtração do fundo usando mistura de Gaussianas propostos por Stauffer e Grimson (1999) e o espaço de cores RGB. Foi realizada uma análise comparando o algoritmo original (STAUFFER; GRIMSON, 1999) com diferentes abordagens e modificações realizadas por Power e Schoones (2002) e KadewTraKuPong e Bowden (2001).

Com os resultados obtidos foi decidido o uso da abordagem de Power e Schoones (2002). Essa análise e os resultados são mostrados na seção de Resultados e Discussões (seção 7.4). Os valores de parâmetros utilizados para inicializar o algoritmo GMM são indicados na tabela 7.1 e descrições na tabela 7.2.

7.2.2.1 Passos do Algoritmo GMM

- a) **Inicializando as K Gaussianas por pixel:** Cada nova K Gaussiana é criada com o valor médio do pixel, peso inicial baixo e alto desvio padrão inicial, o que determina que o quadro inicial da seqüência seja considerado com todos pixels de fundo.
- b) **Checando o Limiar de Desvio Padrão:** Para cada pixel do quadro capturado, checa-se o valor do pixel X_t está dentro do limite de desvio padrão para todas as K Gaussianas, ou seja, a busca pela distribuição mais provável de pertencer à cena de fundo. Para isso, deve-se calcular o desvio padrão σ e a média μ de cada Gaussiana e checar o critério limiar de desvio padrão. De acordo com cada abordagem existe um critério diferente: Stauffer e Grimson (1999) utilizam a

equação 7.1. Power e Schoones (2002) e KadewTraKuPong e Bowden (2001) utilizam a equação 7.2:

$$|R - \mu_R| \leq 2.5\sigma \quad e \quad |G - \mu_G| \leq 2.5\sigma \quad e \quad |B - \mu_B| \leq 2.5\sigma \quad (7.1)$$

$$\left(\frac{(R - \mu_R)}{\sigma_R} \right)^2 + \left(\frac{(G - \mu_G)}{\sigma_G} \right)^2 + \left(\frac{(B - \mu_B)}{\sigma_B} \right)^2 \leq (2.5)^2 \quad (7.2)$$

- c) **Falhando a Busca (pixel é frente):** Quando a busca não encontra uma distribuição entre as K existentes é necessário determinar a distribuição menos provável que é a de menor relação peso/variância. Essa distribuição menos provável é substituída com a nova distribuição do pixel atual com seu valor de média, peso inicial e alta variância.
- d) **Sucesso na Busca (pixel é fundo):** Quando a busca encontra uma Gaussiana os parâmetros devem ser ajustados. Os pesos ω de todas as distribuições devem ser ajustados. A média μ e desvio padrão σ são atualizados somente para a distribuição encontrada na busca enquanto nas outras Gaussianas não ocorre alteração. Os pesos ω , médias μ e variâncias σ são atualizados pelas equações 4.4 a 4.9 (capítulo 4, seção 4.5) . Onde ρ é calculado pela equação 4.8 na abordagem de Stauffer e Grimson (1999), pela equação 7.3 na abordagem de Power e Schoones e pela equação 7.4 para KadewTraKuPong e Bowden (2001).

$$\rho = \alpha / \omega_{i,t,k} . \quad (7.3)$$

$$\rho = \alpha = \max(1/(N+1), 1/L) . \quad (7.4)$$

Sendo N a quantidade de distribuições já consideradas como cena de fundo e L a quantidade de distribuições limite para se considerar o uso de N . A partir de L distribuições as equações consideram α sendo $1/L$, e anteriormente à L as equações consideram α sendo $1/(N+1)$.

- e) **Escolhendo a Distribuição de Fundo:** Para encontrar a distribuição que melhor representa o fundo é necessário: após a atualização dos parâmetros, ordenar decrescentemente as distribuições pela relação ω/σ . Escolher as B primeiras distribuições como sendo cena de fundo de acordo com o critério:

$B = \arg \min_b (\sum_{k=1}^b \omega_{i,k} > T)$, ou seja, se a soma de seus pesos for maior que T (limiar de cena de fundo). Se a distribuição escolhida na busca com sucesso for uma dessas B distribuições, o pixel atual é classificado como fundo. Se a busca falhar ou a distribuição não pertencer a estas B distribuições, o pixel atual é classificado como frente.

Métodos	Parâmetros						
	K	α	σ	Ω	T	β	L
Stauffer/Grimson	3	0.005	25	0.05	0.79	2.5	_
Power/ Schoones	3	0.005	30	0.05	0.7	2.5	_
TraKuPong/Bowden	4	_	12	0.05	0.7	2.5	60

Tabela 7.1: Parâmetros usados no algoritmo GMM nas diferentes abordagens.

Parâmetro	Descrição
K	Número de distribuições de Gaussianas
α	Taxa de aprendizagem
σ	Desvio padrão inicial
ω	Peso inicial
T	Limiar de fundo
β	Fator limite de desvio padrão
L	Limite de quadros considerados fundo

Tabela 7.2: Descrições dos parâmetros de GMM.

7.2.2.2 Limiar de Cor de Pele

Após o processamento pontual na tarefa realizada pelo algoritmo GMM, os resultados são submetidos a uma etapa de detecção por cor de pele somente nos pixels considerados como sendo objetos de interesse. O método utilizado para construir este classificador de pele foi o abordado por Peer, Kovac e Solina (2003) em que um pixel é classificado como cor de pele se a equação 7.5, for satisfeita:

$$(R > 95) \text{ e } (G > 40) \text{ e } (B > 20) \text{ e } ((\max\{R,G,B\} - \min\{R,G,B\}) > 15) \text{ e } (|R-G| > 15) \text{ e } (R > G) \text{ e } (R > B), \quad (7.5)$$

O motivo da escolha por esse método e do espaço de cores RGB foi simplesmente pelo fato que geralmente uma *webcam* tem como padrão a aquisição da imagem em RGB diretamente. Para se usar outros filtros de cor de pele em outro espaço de cor, que mostram maior eficiência nessa finalidade (por exemplo, YCbCr), seria necessário converter cada quadro nesse novo espaço de cor, determinando um grande

esforço computacional e diminuindo em muito a performance de todo o processo. Usando RGB e esse limiar (PEER; KOVAC; SOLINA, 2003) foi obtido um bom resultado na segmentação, não havendo necessidade de uma maior sofisticação nessa etapa.

Na figura 7.3 são mostradas três imagens representando um quadro de uma seqüência de vídeo que passou pela segmentação com regras empíricas de cor de pele em RGB. Na figura 7.3a tem-se o quadro original capturado da seqüência, na figura 7.3b tem-se o resultado da segmentação por cor de pele da imagem original e na figura 7.3c tem-se o contorno extraído.

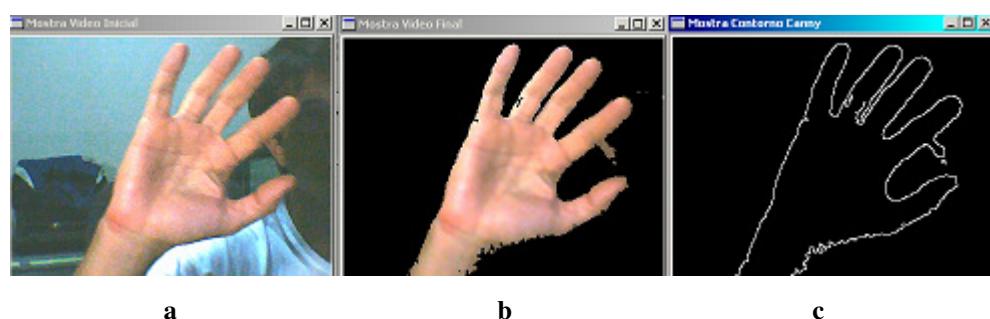


Figura 7.3: Em (a) tem-se a imagem original, em (b) a imagem segmentada por cor de pele, e em (c) a imagem com contorno resultante da imagem segmentada.

7.2.3 Pós-processamento

Com o resultado da segmentação, tem-se uma imagem binária contendo os pixels relativos aos objetos de interesse. Esta imagem binária resultante possui erros de segmentação e ruídos na morfologia do objeto. Portanto, é necessário corrigir ou pelo menos diminuir os erros e ruídos do processo de segmentação. Para isto, primeiramente se utiliza um filtro de suavização na imagem segmentada. Esse filtro utilizado é o Gaussiano com máscara de 5x5. A função *cvSmooth* da biblioteca de visão computacional OpenCV (INTEL, 2005) foi empregada para executar essa etapa do processo. Na seqüência, filtros morfológicos foram utilizados, um filtro de dilatação e um de erosão (GONZALEZ; WOODS, 2000). Também foram empregadas as funções *cvDilate* e *cvErode* da biblioteca de visão computacional OpenCV.

A figura 7.4 exibe uma imagem resultante da segmentação (figura 7.4a) e uma imagem com a mesma imagem segmentada após o tratamento com filtros de suavização e morfológicos (figura 7.4b).



Figura 7.4: Exemplo de uma imagem segmentada com falhas e buracos (a) e depois do pós-processamento (b).

7.2.3.1 Extração do Contorno

Sobre a imagem binária resultante do pós-processamento, obtém-se o contorno da forma de mão. Para isso, foi utilizada a função *cvFindContours* da biblioteca OpenCV que a partir de uma imagem binária cria uma lista de todos os contornos da imagem por aproximação de polígonos. Essa lista é percorrida e somente o polígono de maior área é escolhido. Além disso, somente contornos com áreas maiores que 5% da área total da imagem são considerados como relevantes. Assim, se não existirem contornos maiores que esse limite, o processo de reconhecimento não continua, e o sistema tenta captar o próximo quadro. Um exemplo de resultado desse contorno pode ser visto na figura 5.2, já descrita no capítulo 5, seção 5.4:

7.2.3.2 Remoção do Antebraço

A remoção de antebraço foi implementada com simplificações e modificações do método do artigo de Deimel e Sven Schröter (1999). O algoritmo adotado de remoção é explicado a seguir:

- Através da Transformada da Distância Euclidiana (TDE) achar o centro da palma da mão e a distância máxima.
- Traçar a circunferência cujo centro é o centro da palma da mão e o raio é a distância máxima calculada na TDE multiplicado pelo fator $k = 1.65$. (circunferência *offset*)
- O contorno da mão encontrado deve ser transformado em um polígono aproximado para existir o acesso aos vértices e se encontrar pontos de intersecção com o *offset* da circunferência da palma.

- Sobre a circunferência *offset* verificar os pontos de intersecção do polígono do contorno da mão.
- Os dois pontos consecutivos de maior distância e mais extremos à direita e a esquerda devem representar os pontos de intersecção do punho com a circunferência.
- Eliminar os pontos do polígono do contorno que distam mais do que o raio da circunferência do centro da palma e que estejam próximos dos dois pontos de intersecção calculados anteriormente.

São assumidas algumas restrições para facilitar a remoção. Caso alguma das restrições não for cumprida, possivelmente o algoritmo irá falhar e o antebraço não será removido.

- A posição da câmera deve estar sempre de frente para o usuário, para que a mão sempre esteja na parte superior da imagem e o antebraço na parte inferior.
- A imagem binarizada da mão não pode possuir muitas falhas e buracos na região da palma, pois uma distância máxima menor que a real e um centro de palma deslocado, podem ser encontrados.
- O raio da circunferência da palma deve ser maior que um sexto da largura do retângulo que compreende área do contorno (*Blob*): Raio \geq (Largura do *Blob*) / 6;

Na figura 7.5 existe um exemplo de um contorno extraído sem a remoção do antebraço (figura 7.5a) e com a remoção do antebraço (figura 7.5b).

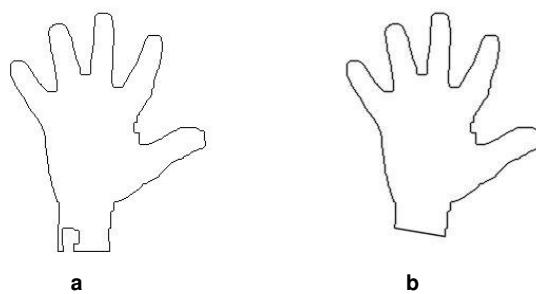


Figura 7.5: (a) Contorno da mão. (b) Contorno com remoção do punho da mão.

7.2.4 Reconhecimento

Para essa etapa, foi definido que o reconhecimento dos gestos deve ser feito baseado na forma que um contorno da mão apresenta em um determinado instante. O objetivo então, é o reconhecimento de alguns gestos básicos. Projetos futuros poderão utilizar esses gestos básicos de forma particular e de acordo com suas necessidades.

7.2.4.1 Determinação de Gestos de Mão

A criação dos contornos de gestos de mão seguirá os seguintes procedimentos:

- Foram gravados, com uma *webcam*, vários vídeos com várias posições (poses) de mão diferentes e 10 diferentes poses foram escolhidas para serem os gestos padrão, ou seja, gestos a serem reconhecidos.
- Executou-se a aplicação com cada vídeo selecionando-se a opção para exportação de cada quadro com o respectivo vetor de características e seqüência de vértices do polígono aproximado do contorno.
- As imagens exportadas foram selecionadas manualmente usando critérios visuais como contorno fechado, gesto bem definido, ocupando pelo menos 30% da imagem e antebraço removido da mão.
- Para cada quadro escolhido manualmente, foram também captados os respectivos arquivos de vetor de características e seqüências de vértices do polígono aproximado do contorno.
- Cada imagem do contorno e correspondentes vetores e polígonos, foram armazenados numa base de dados.
- Isso é feito para que toda vez em que a aplicação de testes é executada, essas informações sejam usadas no casamento de padrões e reconhecimento de gestos.

7.2.4.2 Vetor de Características

São calculados os 14 parâmetros obtidos do contorno do quadro corrente visando realizar o casamento de padrões com os vetores de características. Os parâmetros são descritos no capítulo 6, seção 6.6.

O vetor de características com os 14 parâmetros pode ser representado pelo *vetor* $= (p1, p2, p3, p4, p5, p6, p7, \dots, p14)$ onde :

- $[p1, \dots, p7]$: Representam os sete momentos invariantes de *Hu* (HU, 1962) do contorno.
- $[p8]$: Contém o ângulo de orientação do maior semi-eixo da elipse.
- $[p9]$: Mostra a circularidade do contorno (*circularity*).
- $[p10]$: Representa a excentricidade do contorno (*eccentricity*).
- $[p11]$: Tem atribuição de raio de giro do contorno (*gyration radius*).
- $[p12]$: Determina a dispersão do contorno (*spreadness*)
- $[p13, p14]$: Representam o máximo e mínimo momento de inércia do contorno.

7.2.4.3 Classificação

Na classificação utiliza-se a distância euclidiana entre o vetor do quadro corrente e os vetores padrão após uma normalização de cada parâmetro dos vetores. O algoritmo de classificação recebe como entrada um contorno de consulta com classificação desconhecida e calcula a sua classe. Esse algoritmo é descrito a seguir:

- Dado um contorno de consulta com classificação desconhecida, calcular as 14 características que formam seu vetor de características. Denota-se o vetor de características obtido como x .
- No primeiro estágio do casamento de padrões, são utilizados apenas os 7 momentos invariantes de *Hu* que fazem parte do vetor x . O vetor x será pré-classificado como pertencendo a uma classe quando for obtida uma distância euclidiana entre as características do vetor x e as características do vetor de classe padrão e que possua um limite mínimo referencial de distância.
- Caso o vetor x seja classificado no primeiro estágio, um segundo estágio de classificação irá ocorrer. A orientação do maior semi-eixo da elipse não deve variar mais que 5 graus do vetor da classe padrão pré-classificado.
- Se o segundo estágio também for aceito, todos os parâmetros do vetor x serão utilizados no cálculo da distância euclidiana para detectar se está no limite mínimo referencial de distância. Sua distância será armazenada e comparada com a menor distância calculada entre todos os vetores padrão.

- O vetor x será classificado como pertencendo a uma classe se for o vetor que tiver a menor distância euclidiana entre os vetores das classes padrão selecionados no primeiro e segundo estágio.

7.2.4.4 Exemplo de Classificação

A partir dos vetores de contornos da figura 7.6 considerados como padrões para o casamento, foram criadas as tabelas 7.3 e 7.4 em que são mostrados os valores dos parâmetros dos vetores de características calculados sobre esses contornos padrão.

$$d_{ik} = \sqrt{\sum_{j=1}^p (z_{ij} - z_{kj})^2} \quad (7.6)$$

São 10 classes representativas de contornos e ao todo, o sistema utiliza 14 parâmetros no cômputo da distância euclidiana entre vetores (equação 7.6).

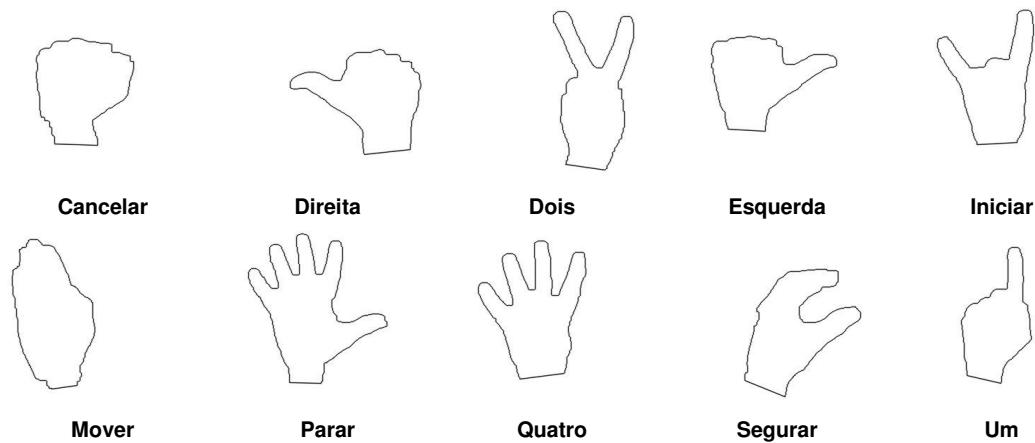


Figura 7.6: Os 10 contornos de gestos de mão usados para casamento de padrões.

A tabela 7.5 representa o vetor de características calculado para o contorno de entrada correspondente à figura 7.7. Na coluna do meio estão os valores reais calculados para todos os 14 parâmetros do vetor. E na coluna da direita estão os valores normalizados para todos os 14 valores reais do vetor.

Sobre os vetores, é realizada a normalização com relação ao conjunto de valores correspondentes a cada parâmetro. Assim, todos os valores estarão em uma distribuição espacial proporcional, impedindo que valores de parâmetros de mais alta ordem, como o parâmetro $p14$ da tabela 7.5, dominem completamente o processo. Isto poderia ocorrer porque os valores de parâmetros deste tipo possuem um valor absoluto significativamente maior do que outros parâmetros de ordem mais baixa.

Parâmetro	Classes de Vetores de Características Padrão				
	Cancelar	Direita	Dois	Esquerda	Iniciar
p_1	0.0295275	0.0292868	0.0261641	0.028101	0.0249103
p_2	2.48787e-05	6.30243e-05	0.000216642	5.56768e-05	4.06761e-05
p_3	1.82703e-06	5.80515e-06	2.63392e-06	6.20182e-06	4.90033e-06
p_4	3.44467e-08	2.45978e-08	7.93953e-07	4.27854e-09	6.31754e-08
p_5	-7.55732e-15	-8.19139e-15	1.13944e-12	-6.75423e-16	3.3327e-14
p_6	1.70421e-10	-1.90985e-10	1.13797e-08	-2.47186e-11	2.30979e-10
p_7	-4.19097e-15	4.39301e-15	1.41083e-13	-1.71887e-16	-1.11754e-14
p_8	0.0295275	0.0292868	0.0261641	0.028101	0.0249103
p_9	0.0285348	0.0734791	0.316469	0.0705067	0.0655512
p_{10}	70.8832	74.2792	80.0626	70.565	72.0124
p_{11}	2.3224	2.38201	2.72024	2.7748	3.31364
p_{12}	1.219	5.83062	1.53787	0.342149	1.42356
p_{13}	9.99384e+08	1.32119e+09	2.45386e+09	1.11663e+09	1.35597e+09
p_{14}	7.10539e+08	7.57674e+08	6.86968e+08	6.4805e+08	8.03165e+08

Tabela 7.3: Tabela mostrando 5 classes (cancelar, direita, dois, esquerda e iniciar) de vetores de características padrão e os valores de seus respectivos parâmetros.

Parâmetro	Classes de Vetores de Características Padrão				
	Mover	Parar	Quatro	Segurar	Um
p_1	0.0347818	0.020751	0.017924	0.0245277	0.0275144
p_2	0.000330285	3.18621e-05	1.94192e-05	0.000112248	0.000285646
p_3	4.58202e-07	1.87861e-07	9.91624e-07	1.01353e-06	5.04812e-07
p_4	5.40887e-09	5.44683e-07	5.04237e-07	4.82372e-07	3.99241e-08
p_5	-1.66825e-16	4.86539e-14	3.5467e-13	3.33158e-13	-3.6696e-15
p_6	-5.1391e-11	2.93054e-09	2.05137e-09	5.09127e-09	6.01774e-10
p_7	-2.11366e-16	-1.67303e-13	3.66076e-14	5.25769e-14	-4.31953e-15
p_8	0.0347818	0.020751	0.017924	0.0245277	0.0275144
p_9	0.273015	0.0739936	0.0604454	0.186581	0.377318
p_{10}	79.4736	79.7949	71.1989	72.8768	67.3978
p_{11}	2.26139	3.46958	3.7433	2.49411	1.97473
p_{12}	4.94024	5.07843	1.48178	0.933088	1.27048
p_{13}	1.74622e+09	2.48516e+09	1.78619e+09	1.64675e+09	1.21059e+09
p_{14}	5.47654e+08	1.42227e+09	1.08122e+09	6.53261e+08	2.89278e+08

Tabela 7.4: Tabela mostrando 5 classes (mover, parar, quatro, separar e um) de vetores de características padrão e os valores de seus respectivos parâmetros.



**Figura 7.7: Contorno de
entrada para reconhecimento.**

Parâmetro	Valor Real	Valor Normalizado
<i>p1</i>	0.0353233	-0,329809772406525
<i>p2</i>	0.000346112	-0,329809772472803
<i>p3</i>	4.52735e-07	-0,329809772473458
<i>p4</i>	1.691e-08	-0,329809772473459
<i>p5</i>	-7.74774e-16	0,329809772473459
<i>p6</i>	-1.63825e-10	0,329809772473459
<i>p7</i>	-1.26051e-15	0,329809772473459
<i>p8</i>	0.0353233	-0,329809772406525
<i>p9</i>	0.277393	-0,329809771947824
<i>p10</i>	82.4042	-0,329809616324643
<i>p11</i>	2.21964	-0,329809768267433
<i>p12</i>	4.79068	-0,329809763395531
<i>p13</i>	1.99289e+09	3,44654669833777
<i>p14</i>	6.17859e+08	0,840980172669177

Tabela 7.5: Tabela exemplo de um contorno de entrada com seus valores do vetor de características e os mesmos normalizados.

Com os valores normalizados do vetor de entrada e dos vetores padrão, realiza-se a comparação entre os vetores obtendo os valores das distâncias mostradas na tabela 7.6. Como a classe padrão *Mover* apresenta a menor distância em relação ao contorno da figura 7.7, esta é a classe de padrão em que o contorno foi classificado.

Classe de Vetor Padrão	Distância Euclidiana	Classificação
Cancelar	0,340403734235595	Nenhuma
Direita	0,26582797576535	Nenhuma
Dois	0,652565867886914	Nenhuma
Esquerda	0,153441392497596	Nenhuma
Iniciar	0,518177800655831	Nenhuma
Mover	0,0365392734307135	Escolhido
Parar	0,572867468785657	Nenhuma
Quatro	0,737297956748547	Nenhuma
Segurar	0,653888386521543	Nenhuma
Um	0,278599609605867	Nenhuma

Tabela 7.6: Tabela com diferenças euclidianas entre o vetor de entrada e 10 dos vetores padrão. A menor distância encontrada foi a da classe Mover.

7.2.4.5 Definição dos Contornos Padrão

Os testes foram divididos em duas etapas:

- Na primeira etapa, dividiu-se em 10 blocos de 10 vídeos cada bloco. Esses vídeos correspondem a um vídeo de um minuto de cada gesto escolhido para ser de uma classe padrão. Assim, cada bloco possui 10 vídeos de 10 gestos diferentes. No total foram gravadas 100 vídeos de 1 minuto, sendo 10 vídeos para cada gesto padrão. Procurou-se alterar o fundo e iluminação em cada bloco de testes. Todos os testes foram realizados em ambientes internos, com luz natural e uma fonte de luz artificial colocada junto à câmera. Todos os vídeos foram gravados com a *webcam* fixa usada no sistema de reconhecimento. Em cada vídeo eram realizados os movimentos da mão para cada gesto, sendo essa pose repetida durante um minuto.
- Na segunda etapa, os 10 blocos de vídeos foram submetidos ao sistema de reconhecimento. Em cada seqüência de vídeo, para cada quadro que era detectado como de algum gesto padrão, o usuário tinha que checar se o gesto detectado estava correto para o quadro em questão. Em caso de reconhecimento correto o usuário simplesmente confirmava o sucesso da averiguação. Em caso negativo, o usuário devia selecionar qual deveria ser o padrão a ser reconhecido

como correto e confirmar a operação. Também existia a possibilidade do sistema reconhecer algum gesto de um quadro que não estivesse relacionado a nenhum dos gestos padrão, assim seria necessário que o usuário selecionasse a opção de erro de detecção. Essa etapa dependente do usuário é necessária para que as informações tenham persistência para a tarefa de observação dos resultados e conclusões.

7.2.5 Material

O algoritmo foi desenvolvido e implementado para operar com o sistema operacional Microsoft Windows. Foi utilizada programação em C++ no compilador Borland C++Builder 6.0. A biblioteca de visão computacional OpenCV (INTEL, 2005) foi utilizada na manipulação e visualização das imagens, conversões de espaços de cores e na extração de contornos. Essa biblioteca possui implementações de ótimo desempenho para aplicações em tempo real. OpenCV é uma biblioteca gratuita desenvolvida pela Intel e significa Open Source Computer Vision Library (INTEL, 2005). É uma biblioteca com funções em linguagem de programação C e classes em C++ que implementam os algoritmos mais populares de processamento de imagens e visão computacional. É uma biblioteca livre para utilização não comercial e comercial.

São exemplos de funções da biblioteca OpenCV:

1. Funções Básicas:

- Conversões de Vetores (*arrays*), transformações, operações básicas (adição, subtração, multiplicação, etc.).
- Operações com matrizes, álgebra linear e funções matemáticas (transposições, SVD, multiplicação, etc.).

2. Funções de Processamento de Imagens:

- Funções de desenho (linha, retângulo, textos, etc.).
- Detectores de contornos e cantos.
- Operações morfológicas.
- Filtros.
- Ferramentas de conexão de componentes.
- Histogramas.

3. Funções Análise de movimento e rastreio de objetos:

- Cálculo de fluxo óptico.
 - Avaliadores (Kalman, condensação).
 - Estatística de cena de fundo (média corrente)
4. Funções de Reconhecimento de Objetos:
- Funções de Eigen.
 - Modelos Ocultos de Markov.

As imagens de entrada são obtidas a partir de uma câmera (*webcam*) com resolução de 320x240 pixels e capaz de captar 15 quadros por segundo. O computador utilizado para os experimentos e desenvolvimento foi um PC AMD Athlon 64 Processor, model 3000, com 1 GB de memória RAM e disco rígido de 80GB, uma saída USB para conexão da *webcam*.

7.3 Resultados e Discussões

Nesta seção serão mostrados os resultados relacionados aos algoritmos usados pelas diferentes abordagens de GMM e dos algoritmos utilizados para a etapa de reconhecimento de gestos.

7.3.1 Escolha do Algoritmo GMM

Primeiramente foram feitos testes utilizando-se os valores de parâmetros ajustados para cada abordagem do algoritmo de distribuições de Gaussianas, mencionados na tabela 7.7. Os resultados desses testes, sem limiar de cor de pele, são mostrados na figura 7.8 e com limiar de cor de pele, na figura 7.9.

Na coluna (a) da figura 7.8, são exibidos os quadros originais da seqüência. Nas colunas (b,c) são exibidas a segmentação por GMM baseada no modelo de Stauffer e Grimson (1999) e extração de contorno. Nas colunas (d,e,f,g) são mostradas a segmentação por GMM e extração de contorno relativas à abordagem realizada por Power e Schoones (2002) e KadewTraKuPong e Bowden (2001), respectivamente.

A primeira coluna da figura 7.9 apresenta os quadros originais da seqüência, as outras colunas apresentam a segmentação por GMM com limiar de cor de pele,

respectivamente para os autores Stauffer e Grimson (1999) (coluna b), Power e Schoones (2002) (coluna c) e por KadewTraKuPong e Bowden (2001) (coluna d).



Figura 7.8: Testes de um vídeo mostrando imagens originais na coluna (a). Nas colunas (b,d,f) mostram imagens segmentadas por GMM sem limiar de cor de pele. Nas colunas (c,e,g) são mostrados os contornos. Em (b,c) são resultados baseados no modelo de Stauffer e Grimson (1999). Em (d,e) baseado em Power e Schoones (2002) e em (f,g) baseado em KadewTraKuPong e Bowden (2001).

Utilizando os valores de parâmetros mais adequados para cada abordagem, o algoritmo GMM de Stauffer e Grimson (1999) conseguiu detectar mais pontos de fundo, embora tenha detectado maior quantidade de pixels de objetos de interesse como

sendo pixels de fundo. Este efeito ocasionou mais buracos e contornos mais deformados do que com as outras abordagens. Power e Schoones (2002) conseguiram formas mais definidas para os pixels de frente, mas também detectaram mais pontos de fundo como sendo de frente em determinadas situações.

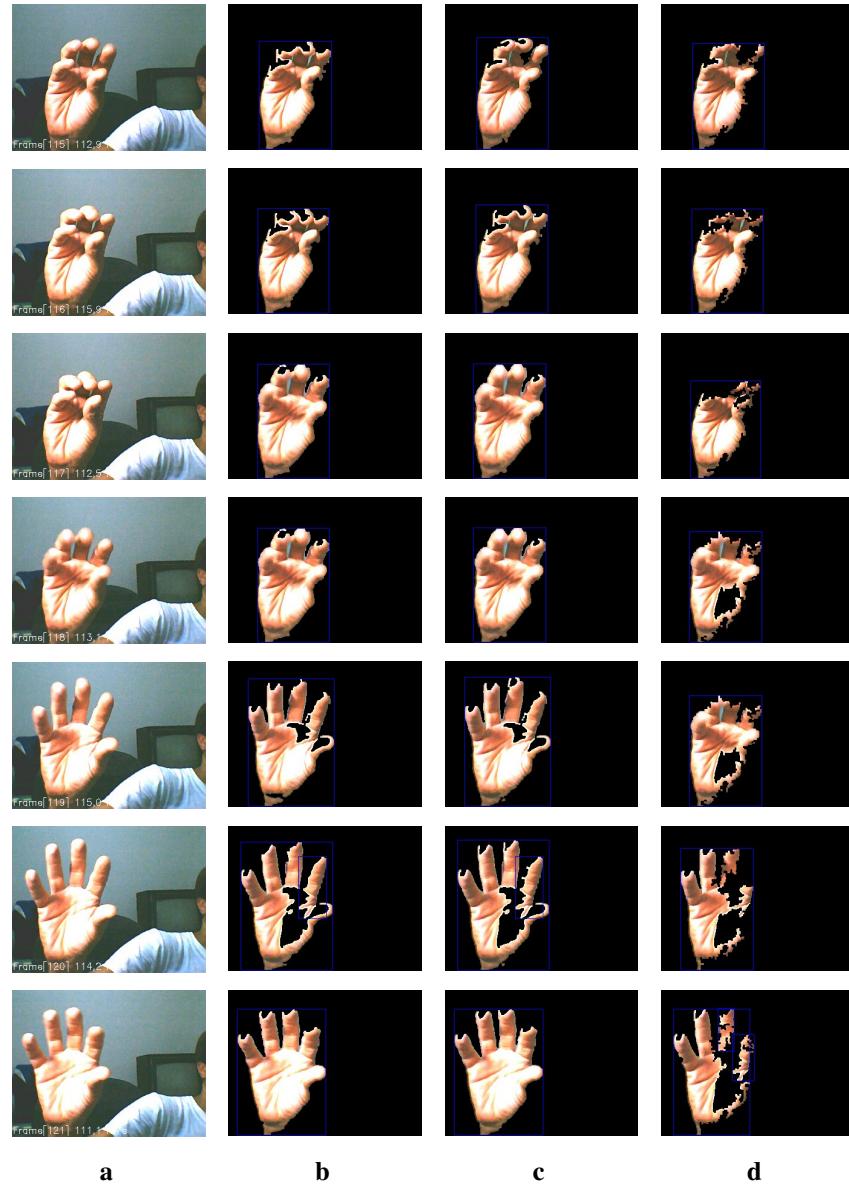


Figura 7.9: Imagens com limiar de cor de pele, baseado no modelo de Stauffer e Grimson (1999) (b) e suas simplificações realizadas por Power e Schoones (2002) (c) e KadewTraKuPong e Bowden (2001) (d).

Objetos de interesse, com pouco ou quase nenhum movimento, demoram a se incorporarem ao modelo de fundo, já na abordagem de Power e Schoones (2002), os objetos são rapidamente incorporados ao modelo de fundo, pois as simplificações

aceleram o processo adaptativo do algoritmo. Na abordagem baseada em KadewTraKuPong e Bowden (2001), os resultados deixaram a desejar no quesito continuidade da segmentação; muitas regiões da mão foram consideradas como fundo, ocasionando dificuldade de obtenção de um contorno de mão satisfatório.



Figura 7.10: Quadros com subtração de fundo por GMM com parâmetros iguais para as três abordagens. Em (a) são mostrados quadros obtidos usando Stauffer e Grimson (1999), em (b) usando Power e Schoones (2002) e em (c) usando KadewTraKuPong e Bowden (2001).

Ocorreram falhas também após o limite de quadros considerados como de aprendizado do algoritmo (L é o Limite de quadros considerados como fundo). Após esse limite a segmentação quase não ocorria, pois são necessários muitos quadros para esse aprendizado. Segundo os autores são necessários por volta de 500 quadros, mas como a aplicação desejada, segmentação de mãos em tempo real, não pode esperar tantos quadros para serem analisados, foi utilizado um limite de apenas 60 quadros, o que provocou o excesso de falhas na segmentação.

Os resultados com o limiar de cor de pele, figura 7.9 ficam muito próximos entre as três abordagens, com ligeiro melhor desempenho para a abordagem de Power e Schoones (2002) com áreas mais cheias que os demais, obtendo-se um melhor contorno.

Esse limiar corrige muitas das falhas de detecção de pixels de objetos de interesse nas três abordagens. A checagem por cor de pele somente é realizada nos pixels já classificados como objetos de interesse, o que diminui o esforço computacional, já que a quantidade de pixels é bem inferior que a imagem toda. Novamente com Power e Schoones (2002) obtiveram melhores resultados com valores de parâmetros iguais para as três abordagens, como é mostrado na seqüência de quadros da figura 7.10. Com Stauffer e Grimson (1999) grandes áreas dos dedos deixaram de ser segmentadas e em KadewTraKuPong e Bowden (2001) segmentaram muitas áreas de fundo como sendo de objetos de interesse.

Resultados	Pós-processamento			Pós-process. e Pele			Sem pós-process.		
	a)Stauffer	b)Power	c)Bowden	d)Stauffer	e)Power	f)Bowden	g)Stauffer	h)Power	i)Bowden
Duração-min	18.77	19.43	18.06	18.61	19.29	17.97	16.48	16.72	15.60
Média -fps	8.83	8.53	9.17	8.91	8.59	9.22	10.05	9.91	10.63
Melhor-fps	9.17	8.91	9.49	9.22	8.98	9.52	10.42	10.35	10.98
Pior -fps	8.18	7.38	8.21	8.29	7.92	8.24	9.30	9.22	9.24

Tabela 7.7: Taxas de quadros/segundo usando os métodos GMM. Os valores mostrados são com pós-processamento, pós-processamento e detecção de pele e sem pós-processamento.

A partir de um vídeo com 9944 quadros, os métodos de GMM de Stauffer e Grimson (1999), Power e Schoones (2002) e KadewTraKuPong e Bowden (2001), foram testados visando a comparação em termos de tempo de processamento utilizando-se os mesmos parâmetros mostrados na tabela 7.8 e $K=3$ modelos. Primeiramente foram usados os algoritmos de GMM sem pós-processamento e sem detecção de pele. Esses resultados são vistos nas três últimas colunas da tabela 7.7 (g,h,i) em referência às três abordagens estudadas. Os valores apresentados nas três primeiras colunas (a,b,c) apresentam os resultados quando usados os algoritmos de GMM com pós-processamento e sem detecção de pele. E nas colunas (d,e,f) são mostrados os resultados com pós-processamento e com detecção de pele.

Parâmetros	K	α	σ	ω	T	β	L
Valores	3	.005	12	0.05	0.79	2.5	60

Tabela 7.8: Valores de parâmetros iniciais usados para todos as abordagens de GMM.

Nos resultados com pós-processamento, mostrados nas colunas (a,b,c) da tabela 7.7, as taxas de quadro por segundo alcançam 8.83 fps em média e tem valores na faixa entre 9.17 a 8.18 fps usando Stauffer e Grimson (1999), 8.53 fps de média e faixa de

8.91 a 7.38 fps para Power e Schoones (2002) e 9.17 fps de média e faixa de 9.49 a 8.21 fps para KadewTraKuPong e Bowden (2001). No teste com pós-processamento e detecção de pele, colunas (d,e,f), as taxas de quadro por segundo alcançam 8.91 fps em média e tem valores na faixa entre 9.22 a 8.29 fps usando Stauffer e Grimson (1999), 8.59 fps de média e faixa de 8.98 a 7.92 fps para Power e Schoones (2002) e 9.22 fps de média e faixa de 9.52 a 8.24 fps para KadewTraKuPong e Bowden (2001).

Os resultados nas colunas (g,h,i) da tabela 7.7, sem pós-processamento, mostram que as taxas de quadro por segundo alcançam 10.05 fps em média e tem valores na faixa entre 10.42 a 9.30 fps usando Stauffer e Grimson (1999), 9.91 fps de média e faixa de 10.35 a 9.22 fps para Power e Schoones (2002) e 10.63 fps de média e faixa de 10.98 a 9.24 fps para KadewTraKuPong e Bowden (2001). Os resultados alcançados por KadewTraKuPong e Bowden (2001) foram os de melhor performance em termos de tempo de processamento, seguidos por Stauffer e Grimson (1999) e Power e Schoones (2002). Os três testes mostram que KadewTraKuPong e Bowden (2001) possuem a melhor média.

As figuras 7.11, 7.12 e 7.13 contém os gráficos de quantidade de quadros por segundo (fps) indicando as performances das três técnicas: Sem pós-processamento, com pós-processamento apenas e com pós-processamento agregando a detecção de pele. Os teste processaram 9944 quadros de um mesmo vídeo para as três propostas de GMM. A linha vermelha representa os valores obtidos usando-se a proposta de Stauffer e Grimson (1999), a linha azul representa os valores obtidos usando-se a proposta de Power e Schoones (2002) e a linha verde representa os resultados de KadewTraKuPong e Bowden (2001). Esses gráficos confirmam que KadewTraKuPong e Bowden (2001) possuem a melhor performance em termos de processamento de tempo. Utiliza maior tempo no início do processamento, mas após uma determinada quantidade de quadros torna-se mais rápido que as outras duas abordagens. O fato é que o algoritmo de KadewTraKuPong e Bowden (2001) gasta uma quantidade de quadros para o modelo aprender com maior acurácia. Em nosso teste, essa fase de aprendizado (60 quadros) é mais lenta que os dois outros métodos (POWER; SCHOONES, 2002) (STAUFFER; GRIMSON, 1999), diferentemente da descrição dos autores KadewTraKuPong e Bowden (2001), afirmando que essa fase seria mais rápida.

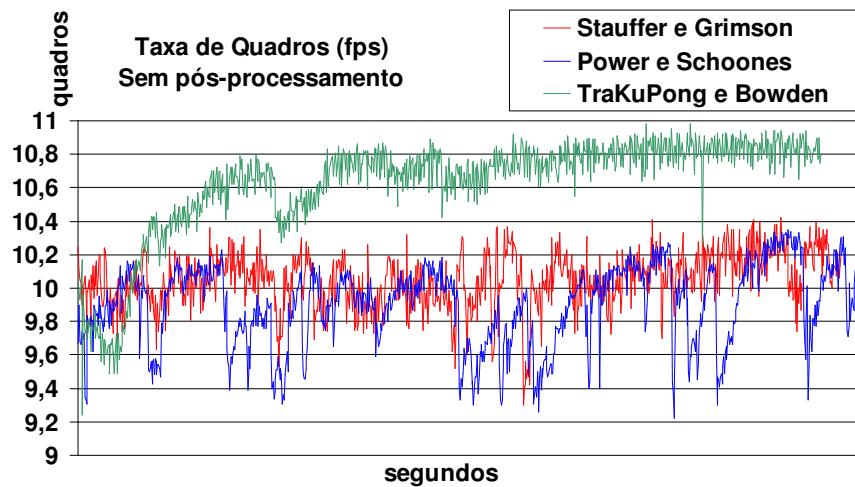


Figura 7.11: Taxa de quadros/segundo das abordagens de GMM sem pós-processamento.

Na maioria do tempo, Stauffer e Grimson (1999) é mais rápido que Power e Schoones (2002). Isso acontece porque nos cálculos propostos por eles (STAUFFER; GRIMSON, 1999), as componentes do pixel vermelho, verde e azul são independentes e tem a mesma variância, diminuindo assim o esforço computacional. Em Power e Schoones (2002), a variância é calculada para cada componente do pixel.

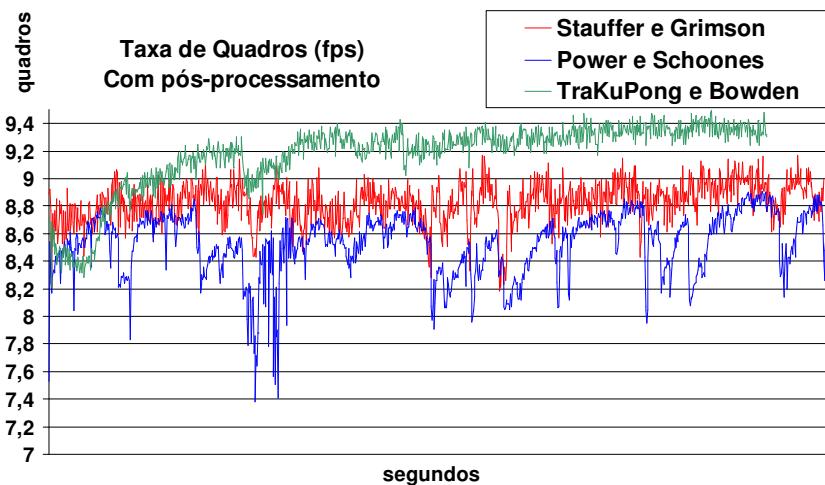


Figura 7.12: Taxa de número de quadros por segundo para o algoritmo GMM em suas três abordagens com pós-processamento.

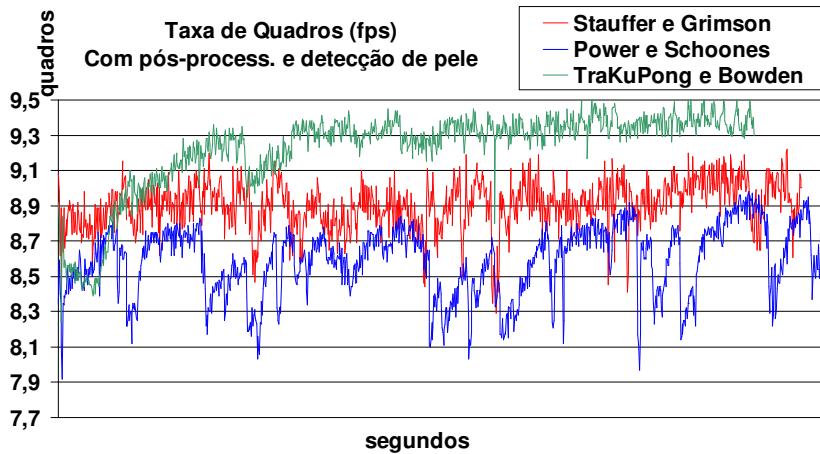


Figura 7.13: : Taxa de número de quadros por segundo para o algoritmo GMM em suas três abordagens com pós-processamento.

Deve-se salientar as diferenças entre o algoritmo GMM, com e sem o tratamento de remoção de ruídos. O processamento pontual gera alguns erros de classificação de pixels, originados por sombras, reflexões ou mesmo objetos de interesse estáticos. Muitas vezes são corrigidos ou atenuados pela fase de remoção de ruídos e com o limiar de cor de pele.

7.3.2 Resultados da Classificação

A tabela 7.9 resume os resultados de classificação de testes de gestos de mãos. Em sua diagonal estão as quantidades totais de acertos em numero de quadros para cada vídeo de um gesto de classe diferente. As 10 tabelas com cada teste realizado contendo 10 vídeos de gestos distintos podem ser checados no apêndice B.

Nessa tabela o elemento (i,j) (linha i , coluna j) corresponde ao número de gestos de mão rotulados como i e classificados pelo usuário como j . As 10 classes de gestos são nomeadas de A a J, como pode ser visto nas duas primeiras colunas de tabela 7.10.

Por exemplo, a terceira linha expressa que existem 481 contornos de consulta da classe C (gesto dois). Sendo que 437 deles foram corretamente classificados pelo algoritmo, um foi classificado como da classe G (gesto parar), oito foram classificados como da classe H (gesto quatro) e 35 foram classificados como da classe I (gesto segurar). Analisando a tabela 7.9 observa-se que:

- A classe I foi a que obteve a maior quantidade de acertos com um total de 1095 acertos e com apenas 9 falsos positivos, 8 para a classe J e 1 para a classe E.

- A classe D foi a que obteve o menor número de acertos com um total de 70 acertos, 1 falso positivo para a classe A e 1 para a classe I.
- A classe que recebeu o maior número de erros (falsos positivos), foi a classe H com um total de 50 resultados errados considerados como se fosse da classe C.

Gestos	A	B	C	D	E	F	G	H	I	J	Total
Total	209	246	531	70	1009	166	600	530	1141	642	5144
A	208	0	0	0	2	0	0	1	4	1	216
B	0	246	0	0	0	0	0	0	0	0	246
C	0	0	437	0	0	0	1	8	35	0	481
D	1	0	0	70	0	0	0	0	1	0	72
E	0	0	3	0	1000	0	2	3	0	1	1009
F	0	0	11	0	2	166	5	8	0	4	196
G	0	0	9	0	1	0	583	2	0	0	595
H	0	0	50	0	1	0	8	505	0	0	564
I	0	0	1	0	0	0	0	0	1095	8	1104
J	0	0	20	0	3	0	1	3	6	628	661

Tabela 7.9: Tabela com valores totais dos 10 testes realizados com os 100 vídeos de gestos.

		Total					
Classes	Gestos	Quadros/60s	Acertos	Erros	Acertos (%)	Erros (%)	
A	Cancelar	3748	208	8	92,71	7,29	
B	Direita	3727	246	0	100	0	
C	Dois	3642	437	44	92,48	7,52	
D	Esquerda	3701	70	2	95,83	4,17	
E	Inicial	3701	1000	9	98,63	1,37	
F	Mover	3720	166	30	83,45	16,54	
G	Parar	3697	583	12	98,24	1,76	
H	Quatro	3624	505	59	89,79	9,64	
I	Segurar	3740	1095	9	98,92	1,08	
J	Um	3741	628	33	94,2	5,8	

Tabela 7.10: Resultado dos testes feitos para a avaliação do reconhecimento dos gestos predefinidos. Para cada um dos 10 gestos é apresentada uma estatística indicando o número e a porcentagem de acertos e erros ao longo do teste.

Todos Gestos	Quadros/60s	Acertos	Erros	Acertos (%)	Erros (%)
Média	3704,1	493,8	20,6	94,43	5,52

Tabela 7.11: Tabela de médias de todos os testes realizados e contabilizado por todos os gestos.

A tabela 7.10 contém informações relativas a todos os testes realizados com vídeos de um só gesto, para as 10 classes diferentes de gestos. O gesto que conseguiu maior taxa de reconhecimento foi a classe B (gesto direita) com 100% de acertos e nenhum falso positivo. Obteve 246 acertos e nenhum reconhecimento errado.

O gesto que conseguiu maior número de acertos em reconhecimento foi a classe I (gesto segurar) que contabilizou 1095 em número de acertos e 9 em número de erros, perfazendo 98.92% de acertos e 1.08 de falsos positivos. O gesto com menor percentual de acertos foi o de classe F (gesto mover) que conseguiu 83.45% em acertos com um número total de 166 acertos e 30 erros.

Considerando todos os testes, o sistema alcançou, em média, 94,43% de acertos e 5,52% de falsos positivos, como é mostrado pela tabela 7.11.

7.4 Conclusões e Trabalhos Futuros

Para a implementação deste trabalho foram realizadas, pesquisas e análises bibliográficas dos trabalhos relativos a algoritmos de reconhecimento de gestos, segmentações de imagens e da biblioteca de visão computacional OpenCV. Neste trabalho foi proposto um sistema em tempo real para a detecção e o rastreamento de gestos da mão em ambientes de trabalho convencionais. Esse sistema foi construído sobre a base de algoritmos de subtração de fundo e detecção de contornos.

Outro propósito do trabalho foi mostrar que sistemas de interação baseados em visão computacional podem ser implementados em computadores convencionais utilizando dispositivos de captura e interação do tipo câmeras de prateleira facilmente acessíveis. Além disso, foi demonstrado que é possível construir o sistema de interação baseado em gestos da mão sem precisar de equipamentos e marcadores colados na mão.

O uso da segmentação de pele justifica-se pela necessidade de separar a mão do ambiente de trabalho e pelas vantagens deste método que permite localizar a mão mesmo que ela mude de forma de uma cena para outra. Apesar do processo de segmentação de pele apresentar algumas falhas de classificação, estas foram superadas através da utilização de filtros de suavização e de morfologia matemática. Esses erros de segmentação muitas vezes são ocasionados pelo brilho e reflexão de alguns objetos e foram atenuados nas etapas posteriores à segmentação. Ainda que alguns erros tenham

persistido, os modelos de segmentação e detecção do contorno utilizados neste trabalho tiveram resultados satisfatórios.

No que se refere aos ambientes de trabalho convencionais, as condições de iluminação e a presença e intensidade das sombras nas imagens definem os parâmetros da segmentação. Uma dificuldade foi estabelecer mecanismos de ajuste automáticos ou ao menos semi-automáticos que considerem essas características. O mecanismo manual utilizado neste trabalho está sujeito a erros, e apesar disso, mostrou nas aplicações que pode funcionar de forma aceitável.

O algoritmo GMM analisado com as três abordagens pode ser utilizado em qualquer aplicação baseada em visão computacional que necessite realizar segmentação de mãos, como, por exemplo, sistemas de interação usuário-máquina. Os modelos de cena de fundo gerados pelos algoritmos de mistura de distribuições de Gaussianas mostraram-se bastante robustos às cenas de fundo complexas como as constituídas por diferentes formas geométricas, texturas e cores, a variações graduais de iluminação e movimentos bruscos ou lentos.

Embora o modelo tenha sido robusto para estas diferentes situações, não apresentaram bons resultados para os casos de variação brusca de iluminação, sombras que são consideradas como objetos de interesse ou quando o objeto em movimento fica praticamente estático, provocando a detecção desse objeto como cena de fundo e seu desaparecimento da cena como consequência.

Esse resultado de desaparecimento da cena é mais evidente quando, aumenta-se a taxa de aprendizagem do processamento pontual, podendo gerar problemas nos casos de objetos de interesse quase estáticos, considerando-se a abordagem de Stauffer e Grinsom (1999). Dependendo da aplicação, esta relação pode ser corretamente balanceada, sem prejuízos maiores ao modelo de fundo.

Como o objetivo desse trabalho é segmentar áreas de mãos para posterior reconhecimento de gestos, a abordagem utilizada por KadewTraKuPong e Bowden (2001) não foi considerada razoável para sua utilização nesse tipo de aplicação devido aos problemas com a segmentação em tempo real usando os *L-recent* quadros. A aplicação não pode esperar tantos quadros na fase de aprendizado por causa da proposta de tempos real, a despeito dos gráficos (figuras 6,7,8) mostrando que o método de KadewTraKuPong e Bowden (2001) tem a melhor performance em termos de tempo de

processamento. Um interessante fato é que os autores (KADEWTRAKUPONG; BOWDEN, 2001) descrevem que a fase de aprendizado deveria ser a mais rápida.

Nos testes com parâmetros específicos para cada abordagem, a comparação entre as abordagens do modelo de distribuições de Gaussianas, Power e Schoones (2002) obtiveram um melhor desempenho principalmente em termos de segmentação do que pelos tempos de processamento, já que teve a pior performance nesse quesito. Na subtração de fundo determinou uma imagem mais homogênea com menos buracos, permitindo-se gerar contornos mais contínuos. Nos testes com limiar de cor de pele para cada abordagem, a diferença de desempenho tanto em qualidade de segmentação quanto em tempo foram bem próximas entre as três abordagens.

Nos testes de tempo de processamento, Stauffer e Grimson (1999) é mais rápido que Power e Schoones (2002), mas visualmente a segmentação não é tão eficiente quanto a segunda. Isso acontece por causa da proposta dos autores (STAUFFER; GRIMSON, 1999) em que as componentes do pixel vermelho, verde e azul são independentes e tem a mesma variância, diminuindo assim o esforço computacional, mas também proporcionando mais erros de segmentação. Em Power e Schoones (2002), a variância é calculada para cada componente do pixel, produzindo uma segmentação mais acurada.

Na etapa de detecção de cor de pele com definições de regras empíricas, foi observada a falta de flexibilidade a grandes alterações de luminosidades e fundos com cores muito próximas da cor da pele. A despeito desses problemas, foi adotado o método de regras empíricas e o espaço de cores RGB simplesmente pelo fato que a *webcam* usada neste trabalho, tem como padrão a aquisição da imagem em RGB diretamente. Para se usar outros filtros de cor de pele em outro espaço de cor (por exemplo, YCbCr), que poderia ser mais eficiente para essa finalidade, haveria a necessidade de converter cada quadro nesse novo espaço de cor, determinando um grande esforço computacional e diminuindo muito a performance de todo o processo. Mesmo usando RGB e esse limiar (PEER; KOVAC; SOLINA, 2003) foi obtido um bom resultado na segmentação, não havendo necessidade de uma maior computação para a detecção de pele.

Na fase de pós-processamento, a Transformada da Distância para obtenção da posição da mão foi utilizada. Neste projeto a TDE foi escolhida como estratégia de

obtenção da posição da mão por fornecer respostas mais precisas mesmo com a presença do antebraço na cena e com diversas configurações da mão. Após a remoção do antebraço, recomenda-se o uso do centro de massa para detectar a posição, caso se deseje trabalhar somente com a mão. Além disso, este método é simples de implementar e rápido em termos de desempenho.

Os momentos de imagens foram utilizados para calcular a orientação da mão a partir de seu contorno, ao contrário de muitos trabalhos que utilizam a área do objeto de interesse (CHAUMETTE, 2004) (HECKENBERG, 1999). Dessa forma o número de pontos utilizados pelo algoritmo cai drasticamente, culminando no aumento significativo de desempenho do mesmo.

Nesta dissertação, foram usados conjuntos de características em sua maioria invariantes à translação, rotação e escala, respectivamente, para o problema de reconhecimento de contornos de mão. Um conjunto de características é extraído do contorno da imagem da mão e um classificador de distância mínima é usado para o reconhecimento do objeto. A metodologia desenvolvida é aplicada para o reconhecimento de gestos estáticos de mão e pode também ser usada no reconhecimento de seqüências de gestos determinando um gesto dinâmico. A alta taxa de reconhecimento obtido permite aplicações práticas e em tempo real.

Os resultados experimentais mostraram que as características escolhidas para o sistema são capazes de discriminar corretamente formas muitas vezes deformadas.

Um aspecto muito importante do problema de reconhecimento de formas reside na parametrização do contorno da forma. Esta parametrização não deve ser arbitrária, ela deve ser invariante em relação a várias transformações desse contorno, como por exemplo, modificação de rotação, escala e translação, para obter-se um bom desempenho neste tipo de problema.

Muitas aplicações interessantes podem ser construídas utilizando gestos da mão. A inclusão de câmeras em muitos dos aparelhos eletrônicos incrementa o potencial de aplicação e justifica as pesquisas em interfaces digitais que não requerem dispositivos intermediários de interação. Apesar das dificuldades encontradas ao longo das fases deste trabalho, os objetivos foram alcançados, isto é, implementa um sistema de reconhecimento de gestos para aplicações em ambientes de trabalho convencionais. As

taxas de erro estiveram dentro de intervalos que não comprometeram a usabilidade do sistema.

Explorar os gestos em outro tipo de aplicações, por exemplo, análise de movimento, permitiria reconhecer gestos dinâmicos na interação. A utilização de técnicas de baixo custo computacional devido ao compromisso adotado com os requisitos de tempo real e sua viabilidade em aplicações práticas de reconhecimento de gestos, foi considerada na escolha do algoritmo e suas abordagens.

Novas pesquisas devem ser desenvolvidas para a melhoria do problema da variação brusca de iluminação sem o prejuízo dos objetos de interesse quase estáticos nas cenas, criando-se uma forma de realimentação do processamento pontual, de forma que se tenha uma classificação supervisionada, com a capacidade de detectar erros de classificação, reajustando o modelo de mistura de distribuições de Gaussianas. Também pode-se utilizar o espaço de cores YCbCr, equacionado diretamente a partir do RGB. Esse espaço de cores separa a luminância das componentes cromáticas e apresenta vantagem por ser uma forma de representação de cores atrativa para modelagem de cor de pele. Esse espaço de cores tem mostrado também melhores resultados no algoritmo de mistura de distribuição de Gaussianas do que com o espaço de cores RGB, mesmo o normalizado (PHUNG; BOUZERDOUM; CHAI, 2002) (BOAZ; ZARIT; QUEK, 1999) (ABDEL-MOTTALEB; HSU; JAIN, 2002).

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDEL-MOTTALEB, M.; HSU, R. L.; JAIN, A. K. Face detection in color images. In: **IEEE Trans. on Pattern Analysis and Machine Intelligence**, v.24, n.5, p.696–706, 2002.
- AHLBERG, J. A system for face localization and facial feature extraction. In: **Tech. Rep. LiTH-ISY-R-2172**, Linkoping University, 1999. Disponível em: <<http://www.citeseer.ist.psu.edu/ahlberg99system.html>>. Acesso em: mar. 2006.
- BALLAD, D. H.; BROWN, Ch. M. **Computer Vision**. Prentice-Hall Inc., Englewood Cliffs (New Jersey), 1982, 523 p.
- BÉRARD, F. **Computer Vision for the Strongly Coupled Human-Computer Interaction**. Doctoral Thesis, Université Joseph Fourier, Grenoble, 1999. Disponível em: <<http://tel.ccsd.cnrs.fr/tel-00004804>>. Acesso em: abr. 2006.
- BIRK, H.; MOESLUND, T. B.; MADSEN, C. B. Realtime recognition of hand alphabet gestures using principal component analysis. In: **10th Scandinavian Conference on Image Analysis**, Lappeenranta, Finland, 1997. Disponível em: <<http://www.citeseer.ist.psu.edu/henrik97realtime.htm>>. Acesso em: jan. 2006.
- BILMES, J. A. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. **Technical Report ICSI-TR-97-021**, University of Berkeley, 1998.
- BISHOP, C. **Neural Networks for Pattern Recognition**, Oxford University Press, 1995.
- BOAZ, J.; ZARIT, S. B. D.; QUEK, F. K. H. Comparison of five color models in skin pixel classification. In: **ICCV'99 International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems**, RATFG-RTS'99, p.58-63, set. 1999.
- BOBICK, A. F.; WILSON, A. D. State-Based Approach to the Representation and Recognition of Gesture, In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.19, p.1325-1337, 1997.
- BOESCH, R. Modellierung von Niederschlagsverlagerungen mit Radardaten; Geo-Processing Reihe, 21, Geographisches Institut der Universität Zürich, 1993, 143p.
- BROWN, R. G.; HWANG, P. Y. C. **Introduction to Random Signals and Applied Kalman Filtering**, 2nd Edition, John Wiley & Sons, Inc, 1992.
- BUHIYAN, M.A.A.; AMPORNARAMVETH, V.; UENO, S.Y.H. Face Detection and Facial Feature Localization for Human-machine Interface, **NII Journal** n.5, 2003.

Referências Bibliográficas

Disponível em: <<http://www.nii.ac.jp/hrd/HTML/Journal/pdf/05/05-03.pdf>>. Acesso em: fev. 2006.

CASSEL J. A quadrowork for gesture generation and interpretation. In: **R. Cipolla and A. Pentland, editors, Computer Vision in Human-Machine Interaction**, Cambridge University Press, New York, p.191–215, 1998.

CHAUMETTE, F. Image Moments: A General and Useful Set of Features for Visual Servoing. In: **IEEE Transactions on Robotics**, v.20, n.4, agost. 2004. Disponível em: <http://www.irisa.fr/lagadic/pdf/2004_itro_chaumette.pdf>. Acesso em: jan. 2005.

CHI, Z.; YAN, H.; PHAM, T. Fuzzy algorithms: with application to image processing and pattern recognition. In: **Advances in Fuzzy Systems – Application and Theory**, World Scientific Publ. Co. Pte. Ltd., Singapore, v.10, 1996.

COSTA, L. F.; CESAR JR., R. M. Shape Analysis and Classification: Theory and Practice. Boca Raton: CRC Press, 2001.

DARRELL, T.; PENTLAND A. Space-time gestures. In: **Proceedings of Computer Vision and Pattern Recognition Conference**, p.335-340, 1993.

DARRELL, T.; ESSA, A.; PENTLAND A. Task-Specific Gesture Analysis in Real-Time Using Interpolated Views, In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.18, p.1236-1242, 1996.

DAVIS, J. C. **Statistics and Data Analysis in Geology**. 2nd Edition. John Wiley & Sons, USA; 646 p.

DAVIS, J.; SHAH, M. Recognizing Hand Gestures, In: Computer Vision - ECCV'94, v.1, p.331-340, 1994. Disponível em: <<http://www.citeseer.ist.psu.edu/article/davis94recognizing.html>>. Acesso em: jan. 2006.

DEIMEL, B. **Development of a Stable Method to Remove the Forearm from Hand in Video Images**. Dissertação (Mestrado), Universität Dortmund, 1998.

DEIMEL, B.; SCHROTER, S. Improving Hand-Gesture Recognition via Video Based Methods for the Separation of the forearm from the Human Hand. In: **Gesture Workshop'99**, Orsay, France, março 1999. Disponível em: <<http://www.citeseer.ist.psu.edu/deime1999improving.html>>. Acesso em: abr. 2006.

DUAN, L.; CUI, G.; GAO, W.; ZHANG, H. Adult Image detection Method Based-On Skin Color Model And Support Vector Machine. In: **Asian Conference on Computer Vision**, Melbourne, Australia, p.797-800, 2002. Disponível em: <<http://www.jdl.ac.cn/doc/2002/Adult%20Image%20Detection%20Method%20Base-On%20Skin%20Color%20Model%20And%20Support%20Vector%20Machine.pdf>>. Acesso em: abr. 2006.

- FILLBRANDT, H.; AKYOL, S.; KRAISS, K. F. Extraction of 3D Hand Shape and Posture from Image Sequences for Sign Language Recognition. In: **International Workshop on Analysis and Modeling of Faces and Gestures**, Nice, France, p.181-186, 2003.
- FITZMAURICE, G.; ISHII, H.; BUXTON, W. Bricks: Laying the Foundations of Graspable User Interfaces. In: **ACM conference on Computer-Human Interaction**, p.442-449, 1995.
- FLECK, M.; FORSYTH, D. A.; AND BREGLER, C. Finding naked people. In: **Proc. of the ECCV**, v.2, p.592–602, 2002.
- FOLEY, J. D.; VAN DAM, A.; FEINER, S.K.; HUGHES, J. F. **Computer graphics: principles and practice**. Reading, MA: Addison-Wesley, 1990, p.1176.
- FREEMAN, W. T.; ROTH, M. Orientation Histogram for Hand Gesture Recognition, In: **Int'l Workshop on Automatic Face and Gesture-Recognition**, p.296-301, 1995. Disponível em: <<http://www.citeseer.ist.psu.edu/freeman94orientation.html>>. Acesso em: dez. 2005.
- GARCIA C.; TZIRITAS, G. Face detection using quantized skin color regions merging and wavelet packet analysis. In: **IEEE Trans. on Multimedia**, v.1, n.3, p.264-277, 1999.
- GAO, X.; BOULT, T. E.; COETZEE, F.; RAMESH, V. Error analysis of background adaption. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, CVPR, v.1, p.503-510, 2000.
- GROSBERG, A.Y.; KHOKHLOV, A. R. **Statistical Physics of Macromolecules**, AIP Press. [ISBN 156396710](#), 1994.
- GOMEZ, G. On selecting colour components for skin detection. In: **Int. Conf. On Pattern Recognition**, Quebec, Canada, v.2, p.961-964, 2002.
- GOMEZ, G.; SANCHEZ, M.; SUCAR, L. On selecting an appropriate colour space for skin detection. In: **Mexican Int. Conf. on Artificial Intelligence**, Merida, Yucatan, Mexico, v.2313, p.70-79, 2002.
- GOMEZ, G.; MORALES, E. Automatic feature construction and a simple rule induction algorithm for skin detection. In: **Proc. of the ICML Workshop on Machine Learning in Computer Vision**, p.31–38, 2002.
- GONZALEZ, R. C.; WINTZ, P. **Digital Image Processing**. Addison-Wesley Publishing Company Inc, 1977.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de Imagens**, traduzido por Roberto M. C. Júnior e Luciano da F. Costa. Editora Edgard Blucher Ltda., São Paulo, 2000.

HANDEMBERG, C. **Fingertracking and Handposture Recognition for Real-Time Human-Computer Interaction**, Dissertação (Mestrado), Fachbereich Elektrotechnik und Informatik der Technischen Universität Berlin, 2001. Disponível em: <http://iihm.imag.fr/pubs/2001/Hardenberg01_FingerTracking.pdf> Acesso em: mar. 2006.

HARVILLE, M.; GORDON, G.; WOODFILL, J. Foreground segmentation using adaptive mixture models in color and depth. In: **Proceedings of the IEEE Workshop on Detection and Recognition of Events in Video**, p.3, 2001.

HECKENBERG, D. **MIME: A Video Based Hand Gesture Interface**, Dissertação (Mestrado), Department of Computer Science and Electrical Engineering, University of Queensland, 1999. Disponível em: <<http://innovexpo.itee.uq.edu.au/1999/thesis/heckendr/thesis.pdf>>. Acesso em: mar. 2006.

HEIKKILA, J.; SILVEN, O. A real-time system for monitoring of cyclists and pedestrians. In: **Proceedings of the Second IEEE Workshop on Visual Surveillance**, p.74-81, 1999.

HU, M. K. **Visual Pattern Recognition by Moment Invariants**, IRE Trans. Info. Theory, v. IT-8, p.179-187, 1962.

INTEL. OpenCV Open Source Computer Vision Library, 2005. Disponível em: <<http://www.intel.com/research/mrl/research/opencv/>>. Acesso em: 10 set. 2005.

JORDAO, L.; PERRONE, M.; COSTEIRA, J.; SANTOS-VICTOR, J. Active face and feature tracking. In: **Proceedings of the 10th International Conference on Image Analysis and Processing**, p.572–577, 1999.

KADEWTRAKUPONG, P.; BOWDEN, R. An improved adaptive background mixture model for real-time tracking with shadow detection. In: **Proc. 2nd European Workshp on Advanced Video-Based Surveillance Systems**, 2001. Disponível em: <<http://www.ee.surrey.ac.uk/Personal/R.Bowden/publications/avbs01/avbs01.pdf>>. Acesso em: jan. 2005.

KAWATO, S.; OHYA, J. Automatic skin-color distribution extraction for face detection and tracking. In: **The 5th Int. Conf. on Signal Processing**, Beijing, China, v.2, p.21-25, 2000.

KENDON, A. Current issues in the study of gesture, **The Biological Foundations of Gestures: Motor and Semiotic Aspects**, Lawrence Erlbaum Assoc, 1986, p.23-47.

- KÜHL, M. G.; SILVA, M. S. Nossa Hair: Sistema para simulação de coloração em tingimento de cabelos, 2004. Disponível em: <<http://www.javasoft.com.br/academic/ArtigoMarceloMoacir.pdf>>. Acesso em: 01 mar. 2005.
- KOLLER, D; WEBER, J.; HUANG, T.; MALIK, J.; OGASAWARA, G.; RAO, B.; RUSSEL, S. Towards robust automatic traffic scene analysis in realtime. In: **Proceedings of the 33rd IEEE Conference on Decision and Control** (Cat. No.94CH34603). IEEE. Part v.4, 1994. Disponível em: <<http://www.citeseer.ist.psu.edu/koller94towards.html>>. Acesso em: 01 mar. 2005.
- LEE, H. K.; KIM, J. H. An HMM-based Threshold Model Approach for Gesture Recognition. In: **Transactions on Pattern Analysis and Machine Intelligence**, v.21, n.10, p.961–972, 1999.
- LIAO, S. **Image Analysis by Moments**. Tese (Doutorado), Faculty of Graduate Studies, The Department of Electrical and Computer Engineering The University of Manitoba, Winnipeg, Manitoba, Canadá, 1993. Disponível em: <http://www.zernike.uwinnipeg.ca/~s_liao/pdf/thesis.pdf>. Acesso em: mar. 2006.
- LIU, N.; LOVELL, B. C. MMX-Accelerated Real-Time Hand Tracking System. In: **Proceedings of the Image and Vision Computing**, New Zealand, p.381-385, 2001.
- MCKENNA, S.; GONG, S.; RAJA, Y. Modelling facial colour and identity with gaussian mixtures. In: **Pattern Recognition**, v.31, n.12, 1998, p.1883–1892.
- MARTINKAUPPI, J.B.; SORIANO, M.N.; LAAKSONEN, M.H. Behavior of skin color under varying illumination seen by different cameras at different color spaces. In: **SPIE 4301 Machine Vision in Industrial Inspection**, v.9, p.102-113, 2001.
- MORRIS, T.; ELSHEHRY, S. O. Hand Segmentation from Live Video. In: **Proceedings of the International Conference on Image Science, Systems, and Technology (CISST'02)**, junho 2002. Disponível em: <<http://www.co.umist.ac.uk/~dtm/cisst2002.pdf>>. Acesso em: mar. 2006.
- NIELSEN, M.; MOESLUND, T., STORRING, M.; GRANUM, E. A procedure for developing intuitive and ergonomic gesture interfaces for HCI. In: **The 5th Int. Workshop on Gesture and Sign Language based Human-Computer Interaction**, Genova, Italy, p.15-17, 2003.
- OKA, K.; SATO, Y.; KOIKE H. Real-Time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems. In: **International Conference on Automatic Face and Gesture Recognition**, Washington D.C., USA, p.429-434, 2002. Disponível em: <<http://www.hci.iis.u-tokyo.ac.jp/~oka/pdf/fg2002.pdf>>. Acesso em: mar. 2005.

PEER, P.; KOVAC, J.; SOLINA, F. Human skin colour clustering for face detection. In: **submitted to EUROCON 2003 – International Conference on Computer as a Tool**, v.2, p.144-148, 2003.

PHUNG, S. L.; BOUZERDOUM, A; CHAI, D. A novel skin color model in ycbcr color space and its application to human face detection. In: **IEEE International Conference on Image Processing (ICIPŠ2002)**, v.1, p.289–292, 2002.

POWER, P.W.; SCHOONES, J.A. Understanding Background Mixture Models for Foreground Segmentation. In: **Proceedings Imaging and Vision Computing New Zealand**, Auckland, New Zealand, p.267-271, 2002. Disponível em: <<http://www.is.irl.cri.nz/pubdoc/2002/JSPP2002.pdf>>. Acesso em: mar. 2005.

PROKOP, R. J.; REEVES, A. P. A Survey of Moment – Based Techniques for Unoccluded Object Representation and Recognition. In: **Graphical Models and Image Processing**, v.54, n.5, p.438-460, 1992.

QUEK, F. Toward a vision-based hand gesture interface. In: **Virtual Reality Software and Technology Conference**, p.17 –31, 1994.

REDNER, R.A.; WALKER, H.F. **Mixture densities, maximum likelihood and the EM algorithm**, SIAM Review, n.26, n.2, p.195-239, 1984.

SCHLENZIG, J.; HUNTER, E.; JAIN, R. Vision Based Hand Gesture Interpretation Using Recursive Estimation. In: **Proceedings of the 28th Asilomar Conference on Signals, Systems & Computers**, v.2, p.1267 – 1271, 1994.

SCHUMEYER, R.; BARNER, K. A color-based classifier for region identification in video. In: **Visual Communications and Image Processing 1998**, SPIE, v.3309, p.189–200, 1998.

SHIN, M. C.; CHANG, K. I.; TSAP, L. V. Does colorspace transformation make any difference on skin detection? In: **IEEE Workshop on Applications of Computer Vision**, Orlando, FL, USA, p.275- 279, 2002.

SIGAL, L.; SCLAROFF, S.; ATHITSOS, V. Skin color-based video segmentation under time-varying illumination. In: **Pattern Analysis and Machine Intelligence, IEEE Transactions**, v.26, p.862–877, 2004.

STAFFORD-FRASER, J. **Video-Augmented Environments**, Tese (PhD), Gonville & Caius College, University of Cambridge, 1996. Disponível em: <<http://www.qandr.org/quentin/research/thesis.pdf>>. Acesso em: jan. 2006.

STARK, M. K. Video Based Gesture Recognition for Human Computer Interaction, University of Dortmund, p.593, 1995.

- STARNER, T.; PENTLAND, A. Real-Time American Sign Language recognition from video using hidden Markov models. In: **International Symposium on Computer Vision**, Coral Gables, USA, p.265, 1995. Disponível em: <http://www-static.cc.gatech.edu/~thad/p/031_10_SL/real-time-asl-recognition-from%20video-using-hmm-disab96.pdf>. Acesso em: nov. 2005.
- STARNER, T.; WEAVER, J.; PENTLAND, A. Real-time american sign language recognition using desk and wearable computer based video. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.20, n.12, p.1371– 1375, 1998.
- STAUFFER, C.; GRIMSOM, W. Adaptive background mixture models for real-time tracking. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR**, v.2, p.252, 1999. Disponível em: <http://www.ai.mit.edu/projects/vsam/Publications/stauffer_cvpr98_track.pdf>. Acesso em: nov. 2004.
- STERN, H.; EFROS, B. Adaptive color space switching for face tracking in multi-colored lighting environments. In: **Proc. of the International Conference on Automatic Face and Gesture Recognition**, p.249-255, 2002.
- SWETS, D. L.; WENG J. Using Discriminant Eigenfeatures for Image Retrieval. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.18, p.831-836, 1996.
- TEAGUE, M.R. Image Analysis Via the General Theory of Moments. In: **Journal of the Optical Society of America**, p.920-930, 1980.
- TEH, C. H.; CHIN, R. T. On Image Analysis by the Methods of Moments. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.10, p.496-513, 1988.
- TOUSSAINT, G. **Pattern Recognition Lecture Notes**, chapter 11, 1994. Disponível em: <<http://www.citeseer.nj.nec.com/79250.html>>. Acesso em: jan. 2005.
- TURK, M.; PENTLAND, A. Eigen Faces for Recognition. In: **Cognitive Neuro-Science**, v.3, n.1, p.77-86 ,1991.
- ULHAAS, K. D.; SCHMALSTIEG, D. Finger Tracking for Interaction in Augmented Environments. In: **International Symposium on Augmented Reality**, New York, New York, p.29-30, 2001.
- VEZHNEVETS, V.; SAZONOV, V.; ANDREEVA, A. A Survey on Pixel-Based Skin Color Detection Techniques. In: **Proc. Graphicon-2003**, Moscow, Russia, p.85-92, 2003.
- WREN, C. R.; AZARBAYEJANI , A.; DARRELL, T.; PENTLAND, A. P. Pfinder: Realtime tracking of the human body. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.19, n.7, p.780–785, 1997.

WANG, H.; CHANG, S. F. A highly efficient system for automatic face region detection in mpeg video. In: **IEEE Transactions on circuits and system for video technology**, v.7, p.615 – 628, 1997.

WONG, K. Y.; SPETSAKIS, M. E. Motion segmentation and tracking. In: **15th International Conference on Vision Interface**, Calgary, Canada, p.80–88, 2002.

WU, Y.; LIU, Q.; HUANG, T.S. An adaptive self-organizing color segmentation algorithm with application to robust human hand localization. In: **Proc. Asian Conf. on Comp Vision**, Taiwan, 2000. Disponível em: <<http://ifp.uiuc.edu/~yingwu/papers/ACCV00.pdf>>. Acessado em: mar. 2006.

YANG, M.-H.; AHUJA, N. Gaussian Mixture Modeling of Human Skin Color and Its Applications in Image and Video Databases. In: **the 1999 SPIE/EI&T Storage and Retrieval for Image and Video Databases**, San Jose, p. 458-466, 1999.

YANG, K.; TREWN, J. **Multivariate Statistical Methods in Quality Management**. McGraw-Hill Companies, Inc., New York, p.181-185, 2004.

YANG, J.; WAIBEL, A. Real-Time Face Tracker. In: **Proceedings of the IEEE Workshop on Applications of Computer Vision**, p.142, 1996.

YANG, J.; LU, W.; WAIBEL, A. Skin-color modeling and adaptation. In: **Proc. Of ACCV'98**. Hong-Kong, v.11, p.687-694, 1997.

APÊNDICE A - Programa REGEM

A.1 Reconhecimento de Gestos de Mão – REGEM

A aplicação REGEM foi desenvolvida para comprovar a eficácia em tempo real no uso das técnicas propostas. Tem função de plataforma base para testes e para utilização em outros módulos de reconhecimento de gestos dinâmicos de mãos. O programa foi feito para operar com o sistema operacional Microsoft Windows utilizando programação em C++ no compilador Borland C++ Builder 6.0.

A biblioteca de visão computacional OpenCV (INTEL, 2005) foi utilizada no processo. As imagens de entrada do sistema são obtidas a partir de uma câmera (webcam) com resolução de 320x240 pixels e capaz de captar 15 quadros por segundo. O computador utilizado para os experimentos e desenvolvimento foi um PC AMD Athlon 64 Processor, model 3000, com 1 GB de memória RAM e disco rígido de 80GB, uma saída USB para conexão da *webcam*.

A.2 Interface Gráfica

A Interface principal da aplicação possui três áreas de interesse:

- Opções de configuração do processo.
- Seqüência de imagens.
- Saída do programa.

A.2.1 Opções de Configuração

Essas opções permitem o usuário escolher todo os parâmetros e alternativas possíveis de todas as etapas do processamento e geração de dados no disco rígido e na tela para comprovação dos resultados e performance. Na figura A.1 é mostrada a tela inicial da aplicação com as opções de menus para seleção de algoritmo de segmentação usando GMM, parâmetros iniciais para esse algoritmo e os modos de captura de seqüência imagens. Existem 5 opções de configuração:

- GMM
- Pós-processamento

- Janelas
- Exportação
- Figuras

A.2.2 Opção GMM

Nessa opção é possível a seleção do tipo da técnica de segmentação que será executada através de diferentes propostas do algoritmo de misturas de distribuições de Gaussianas. Também se pode alterar os parâmetros iniciais utilizados no algoritmo GMM.

A.2.2.1 Algoritmo de Segmentação usando GMM

Nesse menu pode-se escolher qual algoritmo de segmentação que será utilizado quando for carregado um arquivo de vídeo ou acionada a câmera, as opções possíveis são mostradas na figura A.2. Pode-se escolher entre os algoritmos:

- Stauffer e Grimson (1999) com modelo de cores RGB e única variância.
- Power e Schoonees (2002) com modelo de cores RGB.
- KadewTraKuPong e Bowden (2001) com modelo de cores RGB.
- Harville, Gordon e Woodfill (2001) com modelo de cores YCbCr.
- Stauffer e Grimson (1999) com modelo de cores RGB e variâncias por componente.
- E somente filtro de cor de pele com modelo de cores HSV.

A.2.2.2 Parâmetros Iniciais Para GMM

Nesse menu de opções, mostrada na figura A.3, podem-se escolher quais são os parâmetros iniciais quando o algoritmo de segmentação utilizado for o de GMM. Esses parâmetros só são válidos e ficam ativados quando no menu de tipo de segmentação estiver selecionado com o item GMM.

Os parâmetros iniciais são os descritos no algoritmo de Stauffer e Grimson (1999) e possuem os valores iniciais de default descritos abaixo, embora possam ser alterados sempre antes de se iniciar uma nova sequência de imagens, seja via arquivo de vídeo ou webcam. A única exceção é a constante K (fixada em 3), que determina o número de distribuições Gaussianas e não está acessível para o usuário.

Os parâmetros são descritos a seguir:

- Número de distribuições de Gaussianas: $K=3$
- Taxa de aprendizagem: $\alpha=0.005$
- Limiar de fundo. $T=0.79$
- Peso Inicial: $\omega_{init} = 0.02$
- Desvio Padrão: $\sigma_{init} = 12$
- Limite de Desvio Padrão: $\beta = 2,5$

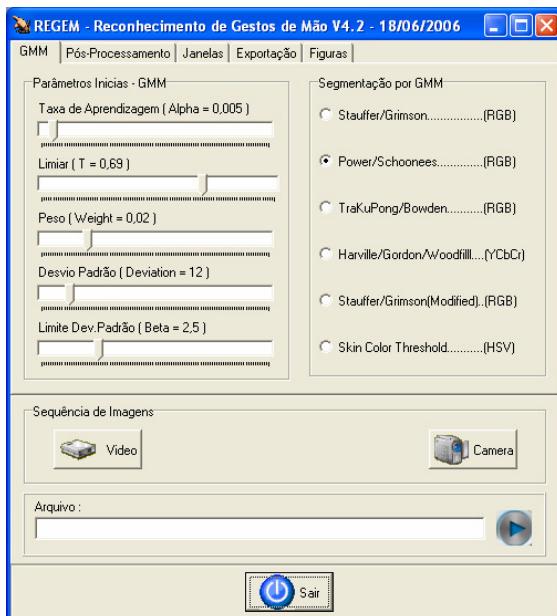


Figura A.1: Figura com a tela inicial do programa REGEM.

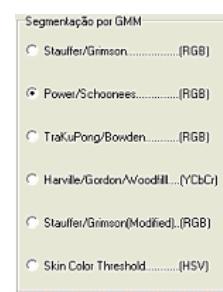


Figura A.2:
Propostas do
algoritmo GMM.

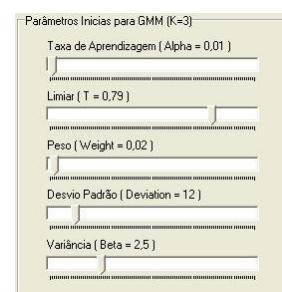


Figura A.3: Menu de
parâmetros iniciais com
valores default.

A.2.2.3 Seqüência de Imagens

Nesse menu existem dois botões para carregar um arquivo em vídeo de uma seqüência de imagens desejada (figura A.4). Uma vez carregado um vídeo, o nome desse é exibido no campo arquivo e caso seja necessário rodar o algoritmo novamente sobre o mesmo vídeo, basta selecionar o botão com uma seta à direita desse campo.

Os botões são descritos a seguir:

- Botão Vídeo: Seleção de um arquivo de vídeo.
- Botão Câmera: Conecta-se com a *webcam* corrente numa porta USB do micro.

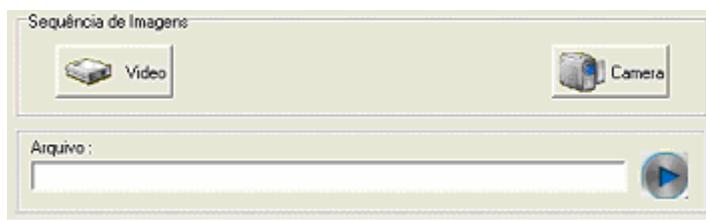


Figura A.4: Menu de seleção do modo de entrada de seqüência de imagens.

São botões para carregar um arquivo de vídeo ou captar o sinal de uma *webcam*.

A.2.3 Opção Pós-processamento

Nessa tela (figura A.5), podem-se escolher todas as etapas de pós-processamento e de refinamento do contorno após a etapa de segmentação usando-se GMM.

Existem as seguintes etapas:

- Filtro de cor de pele.
- Pós-processamento propriamente dito (filtros de suavização e morfológicos, detecção de maior contorno e rastreamento).
- A remoção de punho a partir do contorno da mão.
- Casamento de padrões de gestos estáticos.
- Reconhecimento de um gesto dinâmico.

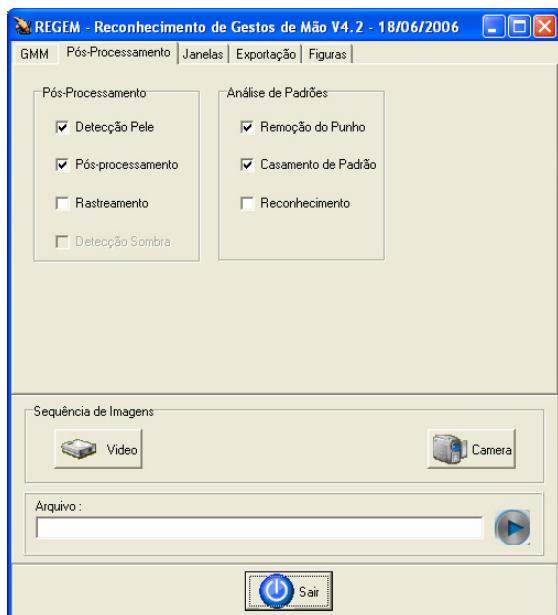


Figura A.5: Opções de pós-processamento, remoção de punho e casamento de padrões.

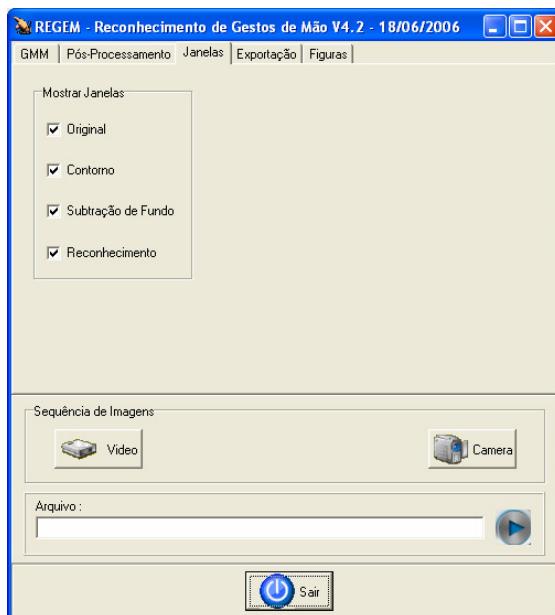


Figura A.6: Opções de janelas exibidas durante o processo.



Figura A.7: Janelas possíveis de serem exibidas pelo programa mostrando a imagem original, subtração de fundo usando GMM e filtro de cor de pele e contorno da mão.

A.2.4 Opção de Janelas

Nessa tela (figura A.6) existem a opções que determinam ou não a exibição das janelas com várias etapas do processamento do programa. As janelas que podem aparecer durante o processamento são e podem ser vistas na figura A.7:

- Original: Exibe ou não o quadro original captado do vídeo ou da webcam.
- Contorno: Mostra ou não a imagem do contorno da mão detectada.
- Subtração de Fundo: Janela que demonstra a região segmentada da mão por subtração de fundo usando GMM e filtro de cor de pele.
- Reconhecimento: Exibe a janela de casamento de padrões para que o usuário aceite ou não o gesto detectado como acerto ou erro. Na figura A.14 existe um exemplo de contorno capturado e detectado corretamente com o padrão de classe *Segurar*. Na figura A.15 é mostrado um exemplo de contorno do gesto de classe *Segurar* capturado e detectado erroneamente como padrão de gesto de classe *Um*.

O usuário deve clicar no botão Sim caso o casamento de padrões tenha detectado corretamente o contorno como na figura A.14. Caso essa detecção seja errada, como na figura A.15, o usuário deve selecionar qual o verdadeiro gesto padrão que corresponde ao contorno de entrada encontrando e selecionando o contorno correto, da lista de gestos padrão, e o botão Sim. Se o contorno capturado não corresponde a nenhum dos contornos de gestos padrão, o usuário deve selecionar o botão Não. Isso é necessário para que as informações de acertos e erros serem arquivadas corretamente.

A.2.5 Opção Exportação

Nessa tela (figura A.8) são exibidas as opções que permitem exportar informações geradas durante o processamento de várias fases do programa. Os tipos de exportação são mostrados a seguir:

- Cálculo de FPS: Valores de quadros por segundo de processamento de GMM.
- Quadro do Vídeo: Imagens originais e geradas podem ser gravadas (imagem original, segmentação e contorno).
- Vetor de Características: Os valores dos parâmetros do vetor de características de cada quadro podem ser gravados em arquivos.
- Polígono do Contorno: As coordenadas dos vértices do polígono correspondente de cada contorno podem ser gravadas em um arquivo.
- Casamento de Padrões: Os resultados de acertos e erros para cada gesto estático detectado no reconhecimento de padrões podem ser gravados em arquivos.
- Reconhecimento de Gesto: Os resultados de acertos e erros para cada gesto dinâmico detectado no reconhecimento podem ser gravados em arquivos.

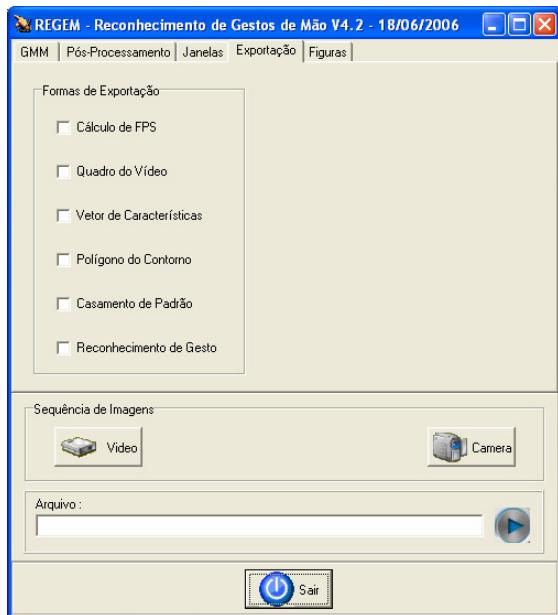


Figura A.8: Opções para exportar os resultados do processamento.

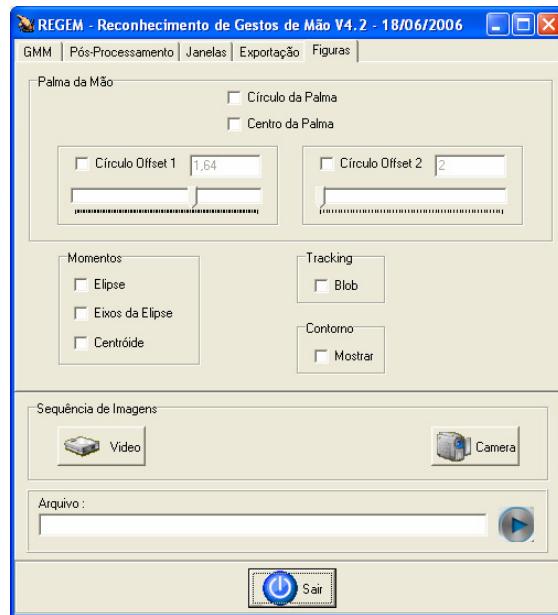


Figura A.9: Opções de figuras de mostram os resultados de etapas de processamento.

A.2.6 Opção Figuras

Nessa tela (figura A.9) existem opções que desenham curvas e retas nas imagens de saída e que mostram os resultados de várias etapas de processamento. Essas figuras são sempre exibidas na janela de segmentação e podem ser as seguintes:

- Centro e círculo que enquadra a palma da mão cujo raio é a máxima distância do centro da palma até sua extremidade (figura A.10).
- Círculos de offset da palma da mão. São usados como referências para detecção de intersecções delas com o contorno da mão para a remoção do punho do restante do contorno (figura A.10).
- Elipse e seus eixos calculados a partir dos momentos da imagem do contorno (figura A.11).
- Contorno da mão obtido com ou sem remoção do punho (figura A.12).
- Retângulo que representa o maior região de *Blob* durante o rastreamento da mão (figura A.13).

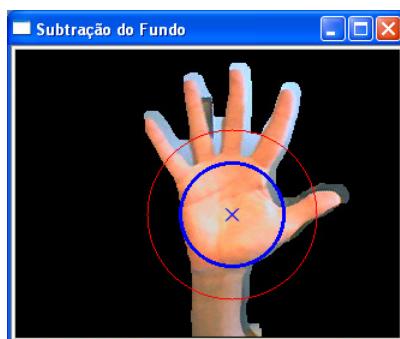


Figura A.10: Círculo da palma da mão em azul. Círculo offset da palma em vermelho.

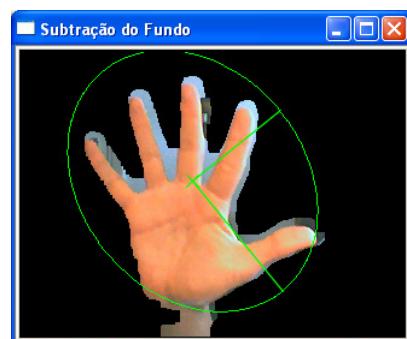


Figura A.11: Elipse e semi-eixos da mão em verde calculados por momentos da imagem.

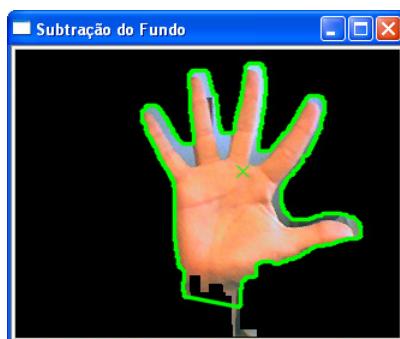


Figura A.12: O contorno sem o punho em verde.



Figura A.13: Retângulo em azul que representa a região de maior *blob* da imagem.

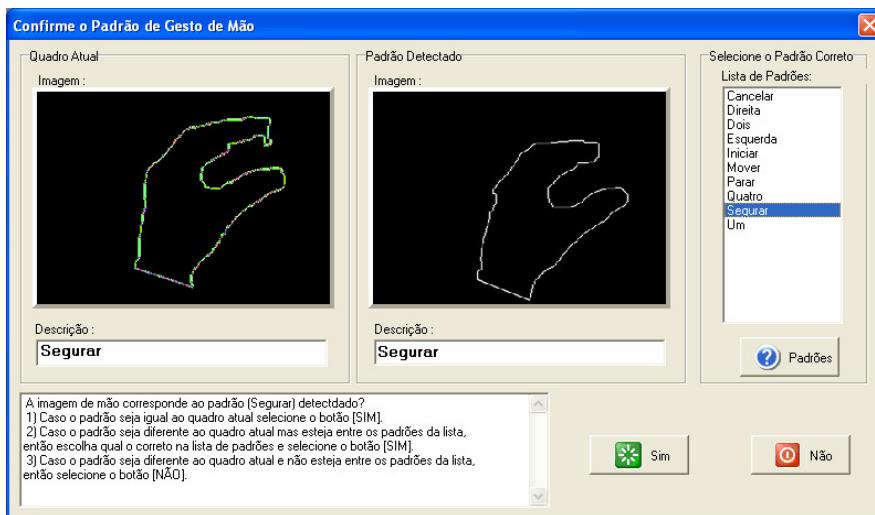


Figura A.14: Contorno encontrado na imagem à esquerda e à direita o contorno padrão Segurar detectado corretamente.

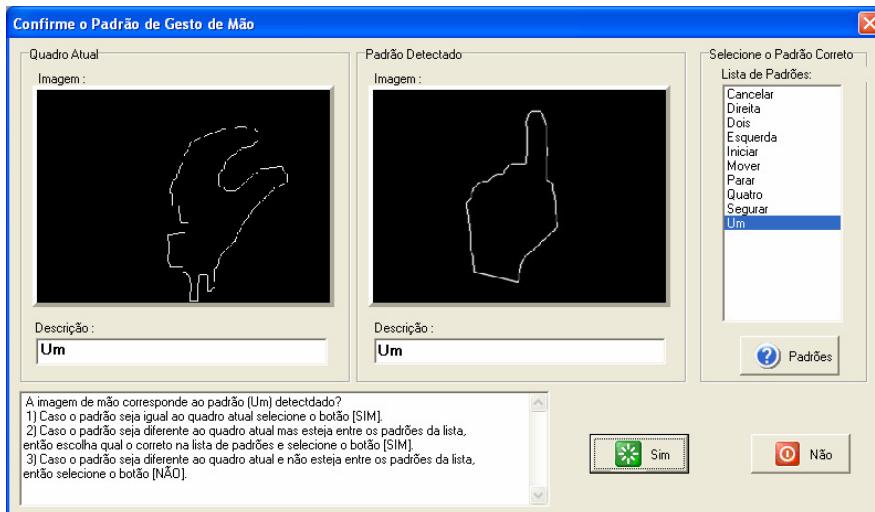


Figura A.15: Contorno encontrado na imagem à esquerda e à direita o contorno padrão Um detectado erroneamente.

APÊNDICE B – Tabelas de Classificações de Gestos

• Tabelas de Classificações de Classes de Gestos

São apresentadas nessa seção todas as tabelas de todos os 10 blocos de testes construídas a partir da exportação de dados do próprio sistema e representados nessas tabelas (tabelas B1 a B10). As tabelas resumem os resultados de classificação de testes de gestos de mãos em cada bloco de teste. Em sua diagonal estão as quantidades totais de acertos em número de quadros para cada vídeo de um gesto de classe diferente. Também existe a tabela resumo de todos os 10 blocos de testes (tabela B11).

	Teste1	A	B	C	D	E	F	G	H	I	J	Total
	Total	48	6	85	19	85	13	71	41	147	57	572
A	Cancelar	48	0	0	0	0	0	0	0	0	0	48
B	Direita	0	6	0	0	0	0	0	0	0	0	6
C	Dois	0	0	76	0	0		0	0	15	0	91
D	Esquerda	0	0	0	19	0	0	0	0	0	0	19
E	Iniciar	0	0	0	0	84	0	1	1	0	0	86
F	Mover	0	0	3	0	1	13	0	1	0	1	19
G	Parar	0	0	3	0	0	0	70	0	0	0	73
H	Quatro	0	0	2	0	0	0	0	39	0	0	41
I	Segurar	0	0	1	0	0	0	0	0	131	3	135
J	Um	0	0	0	0	0	0	0	0	1	53	54

Tabela B.1: Dados gerados do Bloco 1 de testes.

	Teste2	A	B	C	D	E	F	G	H	I	J	Total
		7	49	131	4	37	23	22	36	209	40	558
A	Cancelar	7	0	0	0	0	0	0	1	0	0	8
B	Direita	0	49	0	0	0	0	0	0	0	0	49
C	Dois	0	0	118	0	0		0	0	15	0	133
D	Esquerda	0	0	0	4	0	0	0	0	0	0	4
E	Iniciar	0	0	1	0	34	0	0	1	0	0	36
F	Mover	0	0	2	0	0	23	0	1	0	0	26
G	Parar	0	0	0	0	0	0	21	0	0	0	21
H	Quatro	0	0	1	0	1	0	1	33	0	0	36
I	Segurar	0	0	0	0	0	0	0	0	194	0	194
J	Um	0	0	9	0	2	0	0	0	0	40	51

Tabela B.2: Dados gerados do Bloco 2 de testes.

Apêndice B

	Teste3	A	B	C	D	E	F	G	H	I	J	Total
		12	68	101	1	73	18	63	92	147	44	619
A	Cancelar	12	0	0	0	0	0	0	0	0	0	12
B	Direita	0	68	0	0	0	0	0	0	0	0	68
C	Dois	0	0	98	0	0		0	0	0	0	98
D	Esquerda	0	0	0	1	0	0	0	0	0	0	1
E	Iniciar	0	0	0	0	73	0	1	0	0	0	74
F	Mover	0	0	0	0	0	18	1	0	0	0	19
G	Parar	0	0	0	0	0	0	61	0	0	0	61
H	Quatro	0	0	0	0	0	0	0	91	0	0	91
I	Segurar	0	0	0	0	0	0	0	0	146	0	146
J	Um	0	0	3	0	0	0	0	1	1	44	49

Tabela B.3: Dados gerados do Bloco 3 de testes.

	Teste4	A	B	C	D	E	F	G	H	I	J	Total
		49	27	43	3	46	16	38	45	70	47	384
A	Cancelar	49	0	0	0	0	0	0	0	1	0	50
B	Direita	0	27	0	0	0	0	0	0	0	0	27
C	Dois	0	0	34	0	0		0	6	5	0	45
D	Esquerda	0	0	0	3	0	0	0	0	0	0	3
E	Iniciar	0	0	0	0	46	0	0	0	0	1	47
F	Mover	0	0	1	0	0	16	0	0	0	2	19
G	Parar	0	0	0	0	0	0	38	0	0	0	38
H	Quatro	0	0	8	0	0	0	0	39	0	0	47
I	Segurar	0	0	0	0	0	0	0	0	63	3	66
J	Um	0	0	0	0	0	0	0	0	1	41	42

Tabela B.4: Dados gerados do Bloco 4 de testes.

	Teste5	A	B	C	D	E	F	G	H	I	J	Total
		23	76	43	4	165	16	39	32	108	48	554
A	Cancelar	23	0	0	0	0	0	0	0	0	0	23
B	Direita	0	76	0	0	0	0	0	0	0	0	76
C	Dois	0	0	21	0	0		0	0	0	0	21
D	Esquerda	0	0	0	4	0	0	0	0	0	0	4
E	Iniciar	0	0	1	0	164	0	0	0	0	0	165
F	Mover	0	0	1	0	0	16	1	1	0	0	19
G	Parar	0	0	0	0	0	0	38	0	0	0	38
H	Quatro	0	0	19	0	0	0	0	31	0	0	50
I	Segurar	0	0	0	0	0	0	0	0	108	0	108
J	Um	0	0	1	0	1	0	0	0	0	48	50

Tabela B.5: Dados gerados do Bloco 5 de testes.

	Teste6	A	B	C	D	E	F	G	H	I	J	Total
		7	2	40	3	150	11	107	62	59	66	507
A	Cancelar	7	0	0	0	1	0	0	0	0	0	8
B	Direita	0	2	0	0	0	0	0	0	0	0	2
C	Dois	0	0	35	0	0	0	1	0	0	0	36
D	Esquerda	0	0	0	3	0	0	0	0	0	0	3
E	Iniciar	0	0	1	0	149	0	0	0	0	0	150
F	Mover	0	0	4	0	0	11	0	0	0	0	15
G	Parar	0	0	0	0	0	0	106	0	0	0	106
H	Quatro	0	0	0	0	0	0	0	62	0	0	62
I	Segurar	0	0	0	0	0	0	0	0	59	2	61
J	Um	0	0	0	0	0	0	0	0	0	64	64

Tabela B.6: Dados gerados do Bloco 6 de testes.

	Teste7	A	B	C	D	E	F	G	H	I	J	Total
		9	2	32	3	96	5	64	117	6	62	396
A	Cancelar	8	0	0	0	0	0	0	0	2	0	10
B	Direita	0	2	0	0	0	0	0	0	0	0	2
C	Dois	0	0	10	0	0		0	1	0	0	11
D	Esquerda	1	0	0	3	0	0	0	0	0	0	4
E	Iniciar	0	0	0	0	95	0	0	1	0	0	96
F	Mover	0	0	0	0	0	5	0	1	0	0	6
G	Parar	0	0	6	0	1	0	64	1	0	0	72
H	Quatro	0	0	16	0	0	0	0	113	0	0	129
I	Segurar	0	0	0	0	0	0	0	0	4	0	4
J	Um	0	0	0	0	0	0	0	0	0	62	62

Tabela B.7: Dados gerados do Bloco 7 de testes

	Teste8	A	B	C	D	E	F	G	H	I	J	Total
		33	4	8	22	241	37	120	52	100	157	774
A	Cancelar	33	0	0	0	0	0	0	0	0	0	33
B	Direita	0	4	0	0	0	0	0	0	0	0	4
C	Dois	0	0	4	0	0		0	0	0	0	4
D	Esquerda	0	0	0	22	0	0	0	0	0	0	22
E	Iniciar	0	0	0	0	241	0	0	0	0	0	241
F	Mover	0	0	0	0	0	37	1	4	0	0	42
G	Parar	0	0	0	0	0	0	112	0	0	0	112
H	Quatro	0	0	2	0	0	0	7	48	0	0	57
I	Segurar	0	0	0	0	0	0	0	0	100	0	100
J	Um	0	0	2	0	0	0	0	0	0	157	159

Tabela B.8: Dados gerados do Bloco 8 de testes.

Apêndice B

	Teste9	A	B	C	D	E	F	G	H	I	J	Total
		9	2	36	5	35	18	33	16	143	59	356
A	Cancelar	9	0	0	0	1	0	0	0	0	1	11
B	Direita	0	2	0	0	0	0	0	0	0	0	2
C	Dois	0	0	33	0	0		0	0	0	0	33
D	Esquerda	0	0	0	5	0	0	0	0	1	0	6
E	Iniciar	0	0	0	0	33	0	0	0	0	0	33
F	Mover	0	0	0	0	1	18	0	0	0	0	19
G	Parar	0	0	0	0	0	0	32	0	0	0	32
H	Quatro	0	0	0	0	0	0	0	16	0	0	16
I	Segurar	0	0	0	0	0	0	0	0	142	0	142
J	Um	0	0	3	0	0	0	1	0	0	58	62

Tabela B.9: Dados gerados do Bloco 9 de testes.

	Teste10	A	B	C	D	E	F	G	H	I	J	Total
		12	10	12	6	81	9	43	37	152	62	424
A	Cancelar	12	0	0	0	0	0	0	0	1	0	13
B	Direita	0	10	0	0	0	0	0	0	0	0	10
C	Dois	0	0	8	0	0		0	1	0	0	9
D	Esquerda	0	0	0	6	0	0	0	0	0	0	6
E	Iniciar	0	0	0	0	81	0	0	0	0	0	81
F	Mover	0	0	0	0	0	9	2	0	0	1	12
G	Parar	0	0	0	0	0	0	41	1	0	0	42
H	Quatro	0	0	2	0	0	0	0	33	0	0	35
I	Segurar	0	0	0	0	0	0	0	0	148	0	148
J	Um	0	0	2	0	0	0	0	2	3	61	68

Tabela B.10: Dados gerados do Bloco 10 de testes.

	Teste Geral	A	B	C	D	E	F	G	H	I	J	Total
		209	246	531	70	1009	166	600	530	1141	642	5144
A	Cancelar	208	0	0	0	2	0	0	1	4	1	216
B	Direita	0	246	0	0	0	0	0	0	0	0	246
C	Dois	0	0	437	0	0	0	1	8	35	0	481
D	Esquerda	1	0	0	70	0	0	0	0	1	0	72
E	Iniciar	0	0	3	0	1000	0	2	3	0	1	1009
F	Mover	0	0	11	0	2	166	5	8	0	4	196
G	Parar	0	0	9	0	1	0	583	2	0	0	595
H	Quatro	0	0	50	0	1	0	8	505	0	0	564
I	Segurar	0	0	1	0	0	0	0	0	1095	8	1104
J	Um	0	0	20	0	3	0	1	3	6	628	661

Tabela B.11: Dados de valores totalizados gerados a partir de todos os 10 blocos de testes.

- Tabelas de Totais de erros e acertos**

São apresentadas nessa seção todas a tabelas de todos os 10 blocos de testes construídas a partir da exportação de dados do próprio sistema e representados nessas tabelas (tabelas B12 a B22). As tabelas resumem os resultados de classificação de testes de gestos de mãos em cada bloco de teste em relação ao percentual de acertos e erros e um resumo de todos os 10 blocos de testes é mostrado na tabela B23.

	Gestos Teste 1	Total de Quadros/60s	Total de Acertos	Total de Erros	Acertos (%)	Erros (%)
A	Cancelar	339	48	0	100	0
B	Direita	392	6	0	100	0
C	Dois	341	76	15	83,52	16,48
D	Esquerda	395	19	0	100	0
E	Inicial	396	84	2	97,67	2,33
F	Mover	340	13	6	68,42	31,58
G	Parar	387	70	3	95,89	4,11
H	Quatro	345	39	2	95,12	4,88
I	Segurar	395	131	4	97,04	2,96
J	Um	378	53	1	98,15	1,85

Tabela B.12: Valores de acertos e erros realizados nos teste do bloco 1.

	Gestos Teste 2	Total de Quadros/60s	Total de Acertos	Total de Erros	Acertos (%)	Erros (%)
A	Cancelar	368	7	1	87,5	12,5
B	Direita	367	49	0	100	0
C	Dois	368	118	15	88,72	11,28
D	Esquerda	368	4	0	100	0
E	Inicial	368	34	2	94,44	5,56
F	Mover	367	23	3	88,46	11,54
G	Parar	367	21	0	100	0
H	Quatro	368	33	3	91,67	8,33
I	Segurar	368	194	0	100	0
J	Um	368	40	11	78,43	21,57

Tabela B.13: Valores de acertos e erros realizados nos teste do bloco 2.

	Gestos Teste 3	Total de Quadros/60s	Total de Acertos	Total de Erros	Acertos (%)	Erros (%)
A	Cancelar	368	12	0	100	0
B	Direita	368	68	0	100	0
C	Dois	367	98	0	100	0
D	Esquerda	368	1	0	100	0
E	Inicial	368	73	1	98,65	1,35
F	Mover	352	18	1	94,74	5,26
G	Parar	366	61	0	100	0
H	Quatro	360	91	0	100	0
I	Segurar	368	146	0	100	0
J	Um	368	44	5	89,8	10,2

Tabela B.14: Valores de acertos e erros realizados nos teste do bloco 3.

	Gestos Teste 4	Total de Quadros/60s	Total de Acertos	Total de Erros	Acertos (%)	Erros (%)
A	Cancelar	393	49	1	98	2
B	Direita	350	27	0	100	0
C	Dois	355	34	11	75,56	24,44
D	Esquerda	356	3	0	100	0
E	Inicial	355	46	1	97,87	2,13
F	Mover	397	16	3	84,22	15,78
G	Parar	355	38	0	100	0
H	Quatro	355	39	8	82,98	17,02
I	Segurar	356	63	3	95,46	4,54
J	Um	344	41	1	97,62	2,38

Tabela B.15: Valores de acertos e erros realizados nos teste do bloco 4.

	Gestos Teste 5	Total de Quadros/60s	Total de Acertos	Total de Erros	Acertos (%)	Erros (%)
A	Cancelar	372	23	0	100	0
B	Direita	389	76	0	100	0
C	Dois	340	21	0	100	0
D	Esquerda	356	4	0	100	0
E	Inicial	355	164	1	99,4	0,6
F	Mover	394	16	3	84,22	15,78
G	Parar	356	38	0	100	0
H	Quatro	338	31	19	62	38
I	Segurar	394	108	0	100	0
J	Um	396	48	2	96	4

Tabela B.16: Valores de acertos e erros realizados nos teste do bloco 5.

	Gestos Teste 6	Total de Quadros/60s	Total de Acertos	Total de Erros	Acertos (%)	Erros (%)
A	Cancelar	396	7	1	87,5	12,5
B	Direita	397	2	0	100	0
C	Dois	395	35	1	97,22	2,78
D	Esquerda	397	3	0	100	0
E	Inicial	396	149	1	99,33	0,67
F	Mover	395	11	4	73,33	26,67
G	Parar	396	106	0	100	0
H	Quatro	396	62	0	100	0
I	Segurar	397	59	2	96,72	3,28
J	Um	396	64	0	100	0

Tabela B.17: Valores de acertos e erros realizados nos teste do bloco 6.

	Gestos Teste 7	Total de Quadros/60s	Total de Acertos	Total de Erros	Acertos (%)	Erros (%)
A	Cancelar	394	8	2	80	20
B	Direita	356	2	0	100	0
C	Dois	355	10	1	90,91	9,09
D	Esquerda	356	3	1	75	25
E	Inicial	355	95	1	98,96	1,04
F	Mover	339	5	1	83,34	16,66
G	Parar	356	64	8	88,89	11,11
H	Quatro	356	113	16	87,6	12,4
I	Segurar	355	4	0	100	0
J	Um	355	62	0	100	0

Tabela B.18: Valores de acertos e erros realizados nos teste do bloco 7.

	Gestos Teste 8	Total de Quadros/60s	Total de Acertos	Total de Erros	Acertos (%)	Erros (%)
A	Cancelar	396	33	0	100	0
B	Direita	396	4	0	100	0
C	Dois	397	4	0	100	0
D	Esquerda	396	22	0	100	0
E	Inicial	397	241	0	100	0
F	Mover	396	37	5	88,1	11,9
G	Parar	396	112	0	100	0
H	Quatro	396	48	9	84,21	15,79
I	Segurar	397	100	0	100	0
J	Um	396	157	2	98,74	1,26

Tabela B.19: Valores de acertos e erros realizados nos teste do bloco 8.

	Gestos Teste 9	Total de Quadros/60s	Total de Acertos	Total de Erros	Acertos (%)	Erros (%)
A	Cancelar	368	9	2	81,82	18,18
B	Direita	356	2	0	100	0
C	Dois	368	33	0	100	0
D	Esquerda	353	5	1	83,33	16,67
E	Inicial	355	33	0	100	0
F	Mover	354	18	1	94,74	5,26
G	Parar	363	32	0	100	0
H	Quatro	355	16	0	100	0
I	Segurar	354	142	0	100	0
J	Um	356	58	4	93,55	6,45

Tabela B.20: Valores de Acertos e erros realizados nos teste do bloco 9.

	Gestos Teste 10	Total de Quadros/60s	Total de Acertos	Total de Erros	Acertos (%)	Erros (%)
A	Cancelar	354	12	1	92,31	7,69
B	Direita	356	10	0	100	0
C	Dois	356	8	1	88,89	11,11
D	Esquerda	356	6	0	100	0
E	Inicial	356	81	0	100	0
F	Mover	386	9	3	75	25
G	Parar	355	41	1	97,62	2,38
H	Quatro	355	33	2	94,29	5,71
I	Segurar	356	148	0	100	0
J	Um	384	61	7	89,71	10,29

Tabela B.21: Valores de acertos e erros realizados nos teste do bloco 10.

	Gestos	Total de Quadros/60s	Total de Acertos	Total de Erros	ACERTOS (%)	Erros (%)
	Geral					
A	Cancelar	3748	208	8	92,713	7,287
B	Direita	3727	246	0	100	0
C	Dois	3642	437	44	92,482	7,518
D	Esquerda	3701	70	2	95,833	4,167
E	Inicial	3701	1000	9	98,632	1,368
F	Mover	3720	166	30	83,457	16,543
G	Parar	3697	583	12	98,24	1,76
H	Quatro	3624	505	59	89,787	9,642
I	Segurar	3740	1095	9	98,922	1,078
J	Um	3741	628	33	94,2	5,8

Tabela B.22: Valores de acertos e erros realizados em todos teste dos 10 blocos.