



**Universidade Católica Dom Bosco**

Centro de Ciências Exatas e Tecnológicas

Curso de Engenharia de Computação

**Segmentação de Múltiplos Objetos Utilizando  
Mistura Gaussiana**

Fábio Mamoré Conde

Prof(a). Orientador(a): Priscila Silva Martins, M. Eng.

*Relatório Final submetido como um dos requisitos  
para a obtenção do grau de Engenheiro de Com-  
putação.*

UCDB - Campo Grande - MS - Novembro/2005

# Agradecimentos

Agradeço em especial aos meus pais Denilson e Maria de Fátima que apesar de estarem morando em outra cidade sempre estiveram comigo nos momentos alegres e difíceis deste curso, ao meu brother Daniel que apesar das nossas brigas me transmite uma energia que só ele possui.

Agradeço ao meu amigo/irmão/brother Antônio (vulgo Melão) pelas horas nos dedicando ao curso, jogando sinuca e aprontando um monte. Agradeço ao amigo de longa data Edy por ter me ajudado com a parte de programação em linguagem Java. Agradeço a minha orientadora Priscila Martins, que sempre me ajudou no desenvolvimento da pesquisa e nos empecilhos encontrados durante o percurso deste projeto.

E a todos aqueles que acompanharam e contribuíram para meu crescimento tanto pessoal quanto profissional.

# Resumo

Este trabalho aborda o desenvolvimento de um módulo para captura e segmentação de imagens através do uso da técnica de mistura gaussiana, o que permite realizar o reconhecimento de objetos em movimento. A motivação principal deste trabalho está na pesquisa de novos meios de interação homem-máquina, visto que tal tecnologia pode ser de grande importância para a inclusão de pessoas com necessidades especiais ao mundo tecnológico, minimizando o abismo digital existente. Ao término deste trabalho, objetiva-se a inclusão do módulo desenvolvido junto a plataforma SIGUS, que reúne técnicas na área de visão computacional e interação homem-máquina na construção de sistemas capazes de realizar o reconhecimento de gestos e detecção da direção do olhar.

# Abstract

This work deals with the development of a module for capturing and segmentation images through the use of the gaussian mixture technique, which allows for the recognition of moving objects. The primary motivation for this work is the research and development of new human-machine interaction ways, since such technology can be of great importance to the technological inclusion of people with special needs, therefore minimizing the existing digital abyss. The objective of this work is to include the developed module to the SIGUS platform, which gathers computational vision and human-machine interaction techniques in the construction of systems which are capable of recognizing gestures and detecting gaze direction.

# Lista de Figuras

2.1	Representação de uma imagem Matricial (a) e representação de uma imagem Vetorial (b) . . . . .	13
2.2	Exemplo de uma imagem binarizada . . . . .	13
2.3	Imagem em tons de cinza . . . . .	13
2.4	Imagem no modelo de cor RGB . . . . .	14
2.5	Etapas de Manipulação de Imagens . . . . .	15
2.6	Imagem original (a) e imagem após equalização de histograma (b). Fonte:[9] . . . . .	15
2.7	Imagem original (a) e imagem após o tratamento com o algoritmo de Canny (b). Fonte:[6] . . . . .	16
2.8	Imagem original (a) e imagem limiarizada (b). Fonte:[9] . . . . .	16
2.9	Exemplo de seleção da pele para criação do modelo. . . . .	17
2.10	Exemplo da técnica K-Médias . . . . .	19
2.11	Distribuição Gaussiana . . . . .	19
2.12	Mistura de 2 Gaussianas . . . . .	20
2.13	Histograma de uma Distribuição Gaussiana (à esquerda), e Histograma de uma Mistura de Gaussianas (à direita) . . . . .	21
2.14	Posição dos eletrodos na pesquisa de Barea. Fonte:[2] . . . . .	22
2.15	Pontos de referência para identificação da face humana por Bourel. Fonte:[3] . . . . .	23
2.16	Exemplo das imagens da pupila apresentado por Morimoto. Fonte:[20] . . . . .	23
2.17	A imagem (a) mostra os objetos rastreados, e a imagem (b) representa a imagem capturada pela câmera. Fonte:[25] . . . . .	24
2.18	A imagem (a) representa a imagem real capturada, e a imagem (b) representa a área onde ocorre a circulação de automóveis e pedestres. Fonte:[14] . . . . .	24
2.19	Imagem em 360° capturada na pesquisa de Lee. Fonte:[15] . . . . .	25
3.1	Java Media Framework . . . . .	27
3.2	ImageJ . . . . .	28

---

3.3	Janela da aplicação de segmentação . . . . .	30
3.4	Amostra para criação do modelo de cor . . . . .	30
3.5	Histograma da amostra do objeto segmentado . . . . .	30
3.6	Segmentação do objeto a partir do modelo de cor . . . . .	31
3.7	Amostras de pele para criação do modelo . . . . .	32
3.8	Histograma das amostras utilizadas para criação do modelo com base na pele humana . . . . .	32
3.9	Espaço 2D do modelo criado com RG . . . . .	32
3.10	Exemplos de segmentação: a - RG do modelo RGB, b - SV do modelo HSV, c - rg do modelo RGB normalizado . . . . .	33
3.11	Modelo para criação do histograma em relação ao tempo . . . . .	33
3.12	Histograma de um pixel qualquer da imagem . . . . .	33
3.13	Mistura de M Gaussianas . . . . .	34
3.14	Representação em superfície das gaussianas . . . . .	34
3.15	Segmentação verificando a vizinhança do pixel . . . . .	36
3.16	Exemplos da não adaptação do fundo da cena . . . . .	38
3.17	Exemplos de segmentação com limiares diferentes . . . . .	38
3.18	Diagrama simplificado do protótipo de segmentação imple- mentado . . . . .	39
3.19	Diagrama simplificado do critério de “match”: Modelo Do Fundo da Cena . . . . .	39
3.20	Diagrama simplificado: Modelo Baseado na Cor . . . . .	40
3.21	Área da taxa de aprendizagem . . . . .	42
3.22	Exemplo de segmentação com ajuste periódico dos parâmetros . . . . .	42
3.23	Exemplo de análise da taxa de acerto . . . . .	43
3.24	Exemplo de uma seqüência do pior caso para a segmentação da mão . . . . .	45

# Lista de Tabelas

3.1	Taxa de acerto da quantidade de gaussianas que modelam o fundo . . . . .	43
3.2	Taxa de acerto para segmentado da mão e da face . . . . .	43
3.3	Intervalo de Acerto . . . . .	44
3.4	Percentual dos tipos de erros . . . . .	44
3.5	Baixa taxa de acerto no pior caso . . . . .	44

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>9</b>
<b>2</b>	<b>Fundamentação Teórica</b>	<b>12</b>
2.1	Processamento Digital de Imagem . . . . .	14
2.1.1	Segmentação . . . . .	16
2.1.2	Reconhecimento de Padrões . . . . .	18
2.2	Mistura Gaussiana . . . . .	19
2.3	Revisão de Literatura . . . . .	21
<b>3</b>	<b>Desenvolvimento da Pesquisa</b>	<b>26</b>
3.1	Programas Utilizados . . . . .	26
3.1.1	Java Media Framework . . . . .	26
3.1.2	ImageJ . . . . .	27
3.2	Implementação, Experimentos e Resultados Preliminares . . .	29
3.2.1	Modelo Baseado na Cor do Objeto . . . . .	30
3.2.2	Modelo Do Fundo da Cena . . . . .	33
3.2.3	Associação do Modelo Do Fundo da Cena e do Modelo Baseado na Cor do Objeto . . . . .	38
3.2.4	Resultados Obtidos nos Experimentos . . . . .	42
<b>4</b>	<b>Considerações Finais</b>	<b>46</b>
	<b>Referências Bibliográficas</b>	<b>47</b>



# Capítulo 1

## Introdução

A área de visão computacional pesquisa soluções para problemas que empregam como entrada de dados uma imagem ou seqüência de imagens. São utilizadas técnicas de processamento de imagens para manipulação das imagens com o objetivo de destacar a informação necessária, além de técnicas para interpretar as informações extraídas no pré-processamento inicial.

O poder computacional das máquinas utilizadas atualmente, possibilita o desenvolvimento de uma ampla gama de soluções na área de visão computacional em que um dispositivo de captura de imagem (Webcam) possa ser empregado para obtenção dos dados de entrada [13]. Soluções em visão computacional buscam resolver problemas específicos, fazendo com que o sistema desenvolvido torne-se um sistema especializado, em que as condições de captura da imagem devem ser controladas para se obter o resultado esperado.

Na busca da solução de problemas específicos na área de visão computacional, os pesquisadores da plataforma SIGUS [22] - Plataforma de Apoio ao Desenvolvimento de Sistemas para Inclusão Digital de Pessoas com Necessidades Especiais - pesquisam técnicas para desenvolvimento de novos meios de interação homem-máquina destinados a pessoas com necessidades especiais. A plataforma SIGUS tem como forma de incentivo à pesquisa a disponibilização, gratuita na internet<sup>1</sup>, dos resultados obtidos. As implementações disponibilizadas são desenvolvidas em Java para oferecer portabilidade ao sistema e utilizam pacotes livres, como o ImageJ, para processamento digital de sinais e o JMF, para manipulação de filmadoras digitais.

O reconhecimento de gestos baseado na Língua Brasileira de Sinais (LIBRAS) é tema de estudo para os pesquisadores da plataforma SIGUS. O sinal de entrada é obtido através de uma Webcam e então é realizada a identificação do sinal gestual indicado pelo usuário. Um exemplo de aplicação

---

<sup>1</sup> Disponível no site da plataforma SIGUS <http://www.gpec.ucdb.br/sigus>

a ser desenvolvido é um editor de texto [22] que utiliza a identificação dos sinais como parâmetros de entrada, possibilitando o aumento a inclusão de pessoas com necessidades especiais.

Na detecção da direção do olhar, que também é tema de estudo para os pesquisadores da plataforma SIGUS, os olhos do usuário são rastreados para que se possa calcular a posição na tela onde o esteja olhando. Pessoas tetraplégicas poderiam se comunicar utilizando o movimento dos olhos para, por exemplo, deslocar o mouse na tela do monitor e acionar o comando de clique do mouse.

A segmentação das mãos e dos olhos é uma etapa importante para o desenvolvimento dessas aplicações. Quanto mais robusto a segmentação em relação ao ambiente de captura das imagens, melhor será a interpretação dos sinais obtidos pela Webcam.

Um método para realizar a segmentação de objetos em movimento em uma cena, é realizado com a utilização de uma distribuição gaussiana. Uma função gaussiana é a representação do comportamento de um conjunto de dados, sendo que uma característica da gaussiana é a ocorrência de um valor médio com maior frequência e a medida que se afasta da média, a probabilidade de ocorrência diminui. O valor do pixel pertencente ao fundo da cena, num intervalo de tempo, possui o comportamento de uma distribuição gaussiana e com isso possibilita a criação de um modelo com a utilização de uma gaussiana, em que o valor de moda é igual ao valor da média.

Em ambientes dinâmicos, quando possui mais de um pixel pertencendo ao fundo da cena, uma gaussiana simples não representa a realidade. Para situações que ocorrem mais de um valor para moda, quando ocorre mais de um valor frequentemente, pode ser utilizada a mistura de gaussianas. A mistura gaussiana é a soma das “M” gaussianas pertencentes a mistura, onde cada moda do sistema corresponde ao valor de média de uma gaussiana pertencente ao modelo.

Este trabalho visa colaborar com a construção da plataforma SIGUS através da análise de técnicas de segmentação e a implementação de um módulo de segmentação de múltiplos objetos utilizando mistura gaussiana em linguagem Java, na qual a segmentação realizada nesta pesquisa possa ser utilizada como parâmetro para módulo de rastreamento. A implementação de protótipos, experimentos e a análise dos resultados obtidos foram realizados como parte final do desenvolvimento deste projeto de pesquisa.

Os protótipos desenvolvidos neste trabalho de graduação possuem três métodos diferentes em relação a criação dos modelos. O primeiro foi a criação de um classificador que discrimina os pixels da imagem de acordo com um modelo baseado na cor do objeto. Numa segunda abordagem, o modelo criado não é a representação de um objeto específico, mas o modelo representa

o fundo da cena. No terceiro método, as duas técnicas são utilizadas em conjunto para se definir a qual classe o pixel pertence: ao fundo da cena ou a frente da cena. A mistura gaussiana é utilizada para criar o *modelo* e com isso definir se o pixel pertence ao fundo da cena ou à frente da cena.

A atualização periódica dos parâmetros da mistura gaussiana, realizada nos protótipos desenvolvidos, proporciona uma adaptação ao ambiente quando este varia no tempo. De acordo com alguns autores [25, 15] um sistema robusto deve possuir algumas características importantes:

- Deve se adaptar ao ambiente de captura da imagem.
- Variações na iluminação ou mudança da área de monitoramento, por exemplo, não podem comprometer o funcionamento do sistema.
- Objetos em movimento que permanecem por um longo período de tempo sem deslocamento na imagem podem ser incorporados ao fundo da cena.

O restante deste projeto de graduação está organizado da seguinte maneira: No Capítulo 2, é apresentada a fundamentação teórica e a revisão de literatura, onde são apresentados alguns conceitos sobre processamento digital de imagens e trabalhos correlatos utilizados como referência no desenvolvimento deste projeto de graduação. No Capítulo 3, encontra-se a descrição do desenvolvimento da pesquisa, onde são demonstrados os programas livres utilizados, os experimentos realizados com os protótipos desenvolvidos e os resultados preliminares. No Capítulo 4, as considerações finais sobre o projeto de graduação e alguns possíveis trabalhos futuros na área de visão computacional.

## Capítulo 2

# Fundamentação Teórica

Um dos objetivos da visão computacional é, na interpretação de imagens, simular o sistema visual humano (SVH), porém esta é uma tarefa extremamente complexa [7]. Primeiramente, para entender a complexidade de simular o sistema visual humano, devemos entender conceitos básicos sobre o funcionamento da visão humana.

Resumidamente, no olho humano a formação da imagem ocorre quando a luz atravessa a córnea e chega ao cristalino que projeta a imagem na retina que então passa essa informação, através do nervo ótico, ao cérebro. A quantidade de luz que chega é controlada pela íris, através da ação dos músculos da pupila. O cristalino tem a função de uma lente com o objetivo de focar o objeto desejado, sendo que o ajuste do foco é realizado pela variação do tamanho do cristalino devido a ação dos músculos ciliares, que prendem o cristalino.

A retina possui milhares de sensores sensíveis à luz denominados cones e bastonetes que são excitados em frequências eletromagnéticas variando entre o vermelho e o violeta. A imagem projetada na retina é invertida em relação ao objeto real observado porém, quando enviado pelo nervo ótico, este posicionamento é corrigido no cérebro.

Matematicamente, uma imagem pode ser representada por vetores, neste caso são armazenadas apenas as coordenadas de pontos, linhas e polígonos que formam a imagem, essa representação é denominada Vetorial [7], Figura 2.1 (a) (b) representa este modelo. Uma imagem também pode ser representada por uma matriz bidimensional em que cada coordenada da matriz representa um pixel da imagem (o pixel é definido como uma coordenada no espaço com cor), essa representação é denominada Matricial [7], a Figura 2.1 (b) representa este modelo.

Diversos modelos de cores podem ser utilizados para representar uma imagem, em que cada modelo existente representa um método diferente de

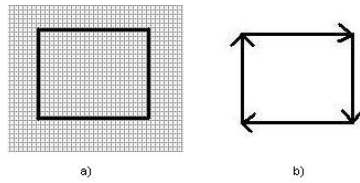


Figura 2.1: Representação de uma imagem Matricial (a) e representação de uma imagem Vetorial (b)

representação da cor, definindo mais ou menos cores que podem ser representadas. A maneira mais simples de se representar de uma imagem é através do modelo binário. São utilizadas duas cores para representar a imagem. A Figura 2.2 é representada pelo modelo binário, pois possui apenas duas cores fundamentais: Branco e Preto. Outra maneira de se representar um imagem, é feita com a representação em tons de cinza, variando entre 0 e 255. O intervalo corresponde as tonalidades existente entre o branco e o preto, Figura 2.3.

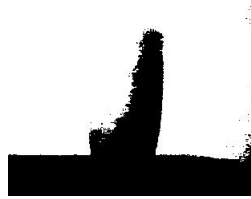


Figura 2.2: Exemplo de uma imagem binarizada



Figura 2.3: Imagem em tons de cinza

O modelo RGB define três cores como base: R que representa a cor vermelha (do inglês *red*), G que representa a cor verde (do inglês *green*) e B que representa a cor azul (do inglês *blue*). Cada cor primária do modelo RGB varia entre 0 e 255, sendo que as demais cores representadas pelo modelo RGB são formadas pela somatório das três cores primárias, a cor Rosa, por exemplo, pode ser representada por RGB(249,81,143) em que o valor de R é igual a 249, G é igual a 81 e B é igual a 143. A Figura 2.4

mostra uma imagem no modelo de cor RGB. O modelo RGB normalizado realiza uma transformação em relação ao modelo RGB. As cores neste modelo são denominadas cores "puras" ou cromáticas [11, 8]. Os parâmetros são normalizados para:  $r + g + b = 1$ , pelas equações:

$$r = \frac{R}{R + G + B}$$

$$g = \frac{G}{R + G + B}$$

$$b = \frac{B}{R + G + B}$$



Figura 2.4: Imagem no modelo de cor RGB

O modelo de cor HSV é mais intuitivo que os modelos RGB. O parâmetro H indica a cor desejada, o parâmetro S indica a saturação da cor e o parâmetro V indica a intensidade da cor. Uma saturação com valor 1 significa que a cor é pura, enquanto uma saturação com valor 0 significa que a cor é totalmente branca.

## 2.1 Processamento Digital de Imagem

A visão computacional utiliza técnicas de processamento de imagem com o objetivo de melhorar a extração dos atributos da imagem. O processo de manipulação de imagens resumido por Márcio e Marcelo [7] refere-se as etapas necessárias para análise de informações presentes em uma imagem. Algumas etapas do processamento de imagens são mostradas na Figura 2.5.

Inicialmente ocorre o tratamento fora da imagem que corresponde, por exemplo, na correção da iluminação do ambiente, na fixação da câmera utilizada para capturar a imagem, na utilização de luvas para o rastreamento das mãos ou ainda na colocação de um fundo uniforme para diminuir a complexidade do ambiente. Já a captura da imagem pode ser realizada através, por exemplo, de scanner de mesa, câmera digital ou webcam.



Figura 2.5: Etapas de Manipulação de Imagens

Na etapa de pré-processamento podem ser utilizadas técnicas para melhorar a qualidade da imagem. Uma técnica para melhorar a imagem denomina-se de equalização de histograma<sup>1</sup>. O histograma da imagem é reformulado em um histograma diferente, com o objetivo de melhorar o contraste da imagem [9]. A Figura 2.6 demonstra um exemplo de equalização de histograma.



Figura 2.6: Imagem original (a) e imagem após equalização de histograma (b). Fonte:[9]

A pesquisa em [6] utiliza, na etapa de pré-processamento, o algoritmo de Canny para suavizar os ruídos e localizar as bordas da imagem. A pesquisa tem o objetivo de realizar um mapeamento móvel para identificar ao longo de ruas ou rodovias, por exemplo, postes ou sinais de trânsito. A Figura 2.7 demonstra o processamento do algoritmo de Canny.

<sup>1</sup>Gráfico em barras que representa a distribuição da frequência dos dados

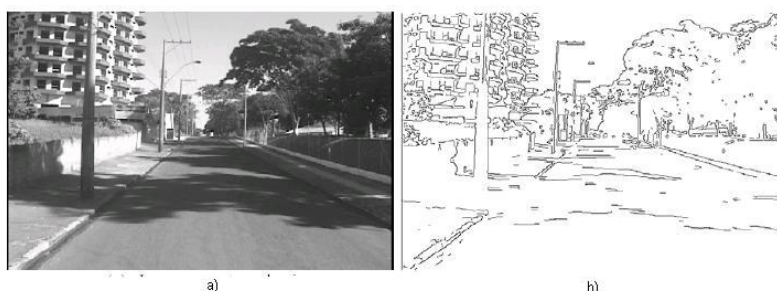


Figura 2.7: Imagem original (a) e imagem após o tratamento com o algoritmo de Canny (b). Fonte:[6]

As etapas de segmentação e reconhecimento de padrões, demonstradas na Figura 2.5, serão detalhadas nas seções abaixo. Estas etapas estão diretamente relacionadas a pesquisa deste projeto.

### 2.1.1 Segmentação

Uma imagem pode possuir diversas regiões semânticas que representam informações a serem extraídas da imagem. As informações podem ser representadas a partir da geometria, da forma, da textura ou cor da região [9]. Um dos objetivos da segmentação é diminuir o volume de informações que serão processadas na etapa seguinte.

Segmentar então consiste em dividir a imagem em diferentes regiões que serão posteriormente analisadas por algoritmos especializados em busca de informações [7]. A análise para identificar segmentos não é trivial. Os pixels das regiões devem possuir certa homogeneidade, sendo delimitadas por bordas ou outras região vizinhas.

Uma técnica de segmentação é a segmentação por limiarização, que consiste em converter uma imagem em níveis de cinza para uma imagem binária. Na Figura 2.8 deseja-se realizar a análise de documentos manuscritos [9], para isso é realizado a segmentação por limiarização para destacar os caracteres que serão analisados posteriormente.

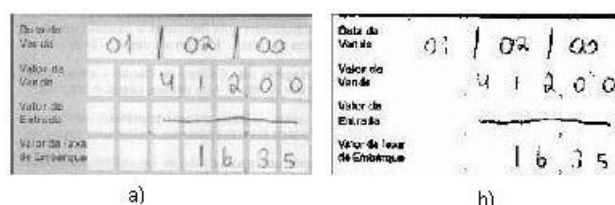


Figura 2.8: Imagem original (a) e imagem limiarizada (b). Fonte:[9]



Com a imagem, de uma seqüência de imagens, segmentada pode ser realizado o rastreamento de objetos na qual cada área segmentada pode corresponder a um objeto a ser rastreado. Rastreamento é o ato de seguir uma região específica em uma seqüência de quadros. Uma técnica simples de rastreamento é a subtração de dois quadros consecutivos. A diferença das imagens indicará os pixels que sofreram alteração e com isso indicam possível movimento de objetos na seqüência de imagens.

Outro exemplo de método de rastreamento, é a utilização da modelagem 2D do objeto. Após a identificação do objeto pelo processo de segmentação, o modelo 2D do objeto é criado através de transformações que preservam as linhas e o paralelismo dos objetos na imagem.

O rastreamento de objetos específicos, como a face humana, pode ser realizado com a criação de um modelo com base na cor do objeto. O classificador baseado na cor do objeto irá definir se o pixel analisado possui a propriedade da cor da pele. Com a identificação dos pixels pertencentes a cor da pele, os pixels são rastreados na seqüência de quadros. A Figura 2.9 mostra um exemplo de seleção de amostras para criação do modelo de cor da pele [11].



Figura 2.9: Exemplo de seleção da pele para criação do modelo.

A técnica de rastreamento denominada correlação é uma técnica de reconhecimento de padrões que mensura a similaridade entre um padrão amostrado e outro padrão de teste. Para se determinar a correlação entre duas imagens pode-se utilizar a soma das diferenças quadradas (SDQ) entre os pixels das duas imagens analisadas. Quanto menor o valor obtido entre a subtração do SDQ das duas imagens, menor é a diferença entre as duas imagens.

Outra técnica de rastreamento é a identificação de regiões homogêneas na imagem. Após a identificação da região desejada, o rastreamento utiliza o valor da medida de correlação cruzada para seguir o objeto na seqüência de quadros. Uma maneira de criar regiões é através da técnica de semear, em que são identificadas regiões homogêneas ao ponto selecionado aleatoriamente na imagem. Esta técnica possui a vantagem de criar regiões independentes uma da outra, podendo rastrear mais de uma objeto.

### 2.1.2 Reconhecimento de Padrões

O reconhecimento de padrões é a ciência que trata da classificação de objetos, em categorias, através da análise de seus atributos. O reconhecimento de padrões pode ser feito através de aprendizado de máquina com a identificação de grupos de padrões (do inglês *clusters*) que possuem características semelhantes entre si e que os diferenciam de outra classe [10].

Existem dois métodos de aprendizado utilizados no reconhecimento de padrões: supervisionado e não supervisionado. O primeiro possui uma etapa de treinamento em que os dados de entradas são divididos em classes com o auxílio de um supervisor que os discrimina, enquanto que, no outro, os dados são classificados de acordo com a similaridade entre os padrões e não com o auxílio de um supervisor [4]. Uma maneira de classificação é através de técnicas estatísticas, em que se observa a probabilidade de um determinado padrão (X) pertencer a uma classe C específica [27].

O reconhecimento de padrões é difícil de ser realizado, pois este implica em vários fatores que podem comprometer o desempenho computacional da aplicação[26]. Uma das muitas técnicas de reconhecimento de padrões é a utilização de Redes Neurais Artificiais.

Nunes e Prado [21] implementaram um simulador de visão computacional para reconhecimento de formas geográficas obtidas através de um scanner ou câmera CCD<sup>2</sup>. Neste trabalho foi desenvolvida uma rede neural probabilística como classificador, a rede classificou corretamente todas as imagens, sendo que nas imagens adquiridas pela câmera CCD a quantidade de dados de treinamento da rede foi maior.

A técnica de K-Médias também é utilizada para reconhecimento de padrões. O “K” corresponde a quantidade de classes existente no sistema e deve ser um valor fixo pré-definido. Tendo como exemplo a Figura 2.10, são definidas 3 classes e cada classe possui seu valor de centróide<sup>3</sup>, ponto mais central na classe, mostrado em cinza no exemplo. Para se definir a qual classe um padrão pertence é calculada a distância entre o padrão desconhecido e a centróide de cada classe. A centróide que possuir a menor distância, em relação ao padrão analisado, é a classe da qual o padrão pertence.

Outro método para realização do reconhecimento de padrões, é através da utilização de uma distribuição matemática que represente o comportamento de um conjunto de dados e com isso se crie um modelo para representar uma determinada classe. Neste trabalho de pesquisa, a modelagem da classe é realizada através da utilização da mistura de gaussianas que será explicada

---

<sup>2</sup>CCD (Charge Coupled Device) - Tipo de sensor de captura de imagem utilizado em câmeras.

<sup>3</sup>A centróide é o valor médio dos padrões na classe.

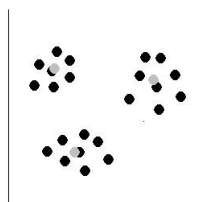


Figura 2.10: Exemplo da técnica K-Médias

na próxima Seção.

## 2.2 Mistura Gaussiana

A distribuição gaussiana é um método utilizado para representar o comportamento de um conjunto de dados através de um modelo estatístico, e é representada pelos seus parâmetros: média ( $\mu$ ) e desvio padrão ( $\sigma$ ). O desvio padrão é utilizado para o caso de atributo unidimensional e a matriz de covariância ( $\Sigma$ ) para atributo multidimensional. Com a distribuição gaussiana pode-se modelar a função de densidade de probabilidade (fdp) e com isso construir um sistema de classificação através da estimação da fdp e do teorema de Bayes [19, 5].

Uma característica da função gaussiana é a ocorrência de um valor médio com maior frequência, e a medida que se afasta desse valor médio, a probabilidade de ocorrência diminui. Neste caso o valor da moda é igual à média. A Figura 2.11 representa o gráfico e a Equação 2.1 representa a fórmula da distribuição gaussiana em uma dimensão.

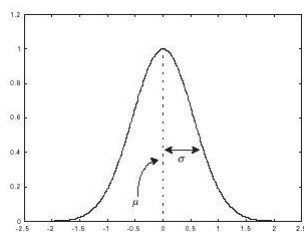


Figura 2.11: Distribuição Gaussiana

$$P(x) = \frac{1}{(2\pi\sigma_1^2)^{\frac{1}{2}}} \exp\left(\frac{-||x - \mu_1||^2}{2\sigma_1^2}\right) \quad (2.1)$$

Quando ocorre mais de um valor para moda, pode ser utilizada a Mistura de M Gaussianas. A variável “M” corresponde a quantidade de gaussianas (modas) da distribuição. A Figura 2.12 representa a mistura de duas gaussianas, com uma dimensão, pois ocorrem dois valores de moda no sistema, sendo representada pela Equação 2.2.

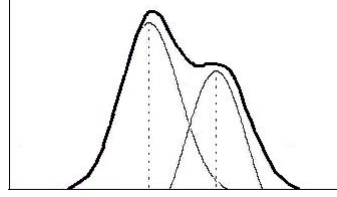


Figura 2.12: Mistura de 2 Gaussianas

$$P(x) = \frac{\omega_1}{(2\pi\sigma_1^2)^{\frac{1}{2}}} \exp\left(\frac{-\|x - \mu_1\|^2}{2\sigma_1^2}\right) + \frac{\omega_2}{(2\pi\sigma_2^2)^{\frac{1}{2}}} \exp\left(\frac{-\|x - \mu_2\|^2}{2\sigma_2^2}\right) \quad (2.2)$$

A fórmula geral da fdp é representada pela soma da multiplicação do peso ( $\omega$ ) pela gaussiana ( $\eta$ ) nas M gaussianas da mistura [25]. A soma dos M pesos da mistura é 1. A Equação 2.3 é utilizada para atributos unidimensionais e a Equação 2.4 é utilizada para atributos multidimensionais.

$$P(x) = \sum_{j=1}^M \omega_j * \eta(x; \mu_j, \sigma_j) \quad (2.3)$$

$$P(x) = \sum_{j=1}^M \omega_j * \eta(x; \mu_j, \Sigma_j) \quad (2.4)$$

É apresentado em [12] uma pesquisa que tem como objetivo realizar a segmentação do disco glandular da mama. Para ser feita a segmentação na imagem é utilizada a mistura gaussiana através do cálculo da fdp descrita na Equação 2.5. A imagem utilizada está representada em tons de cinza com intensidade que varia entre 0 e 255. Por estar representada apenas pela intensidade do tons de cinza, o atributo é unidimensional.

$$\eta(x; \mu_j, \sigma_j) = \frac{1}{(2\pi)^{\frac{d}{2}} |\sigma_j|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(x-\mu_j)^T \sigma_j^{-1} (x-\mu_j)} \quad (2.5)$$

Além do caso unidimensional existem situações que o atributo possui mais de uma dimensão, nesses casos é utilizada a Equação 2.6 para o cálculo da fdp multidimensional [23, 5]. No caso multidimensional a matriz de covariância possui dimensões  $d \times d$ , em que a variável  $d$  representa o número de dimensões do atributo. Por exemplo, o pixel do modelo de cor RGB possui três dimensões: vermelho, verde e azul.

$$\eta(x; \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)} \quad (2.6)$$

No caso específico deste trabalho de pesquisa, a mistura de gaussianas será utilizada para modelar o fundo da cena. A Figura 2.13 mostra o histograma da variação em relação ao tempo, de um pixel capturado pela webcam.

O histograma à esquerda da Figura 2.13 ocorre em ambientes sem variações no fundo da cena, já o histograma à direita da Figura 2.13 ocorre em ambientes dinâmicos, quando o fundo possui mais de uma valor para moda.

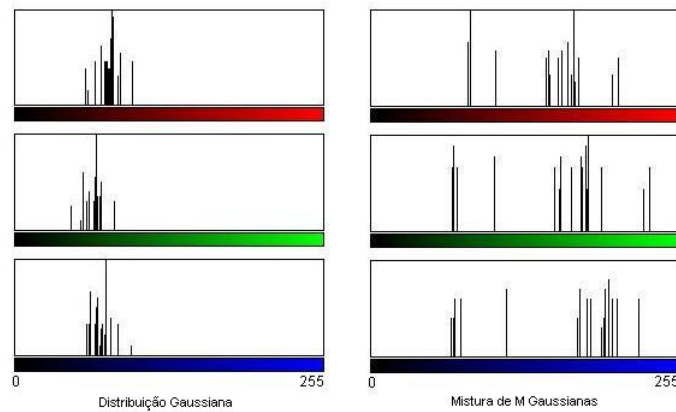


Figura 2.13: Histograma de uma Distribuição Gaussiana (à esquerda), e Histograma de uma Mistura de Gaussianas (à direita)

Com a análise dos histogramas da Figura 2.13, podemos concluir que o pixel possui uma distribuição gaussiana, em relação a variação no tempo, e com a utilização da mistura de gaussianas é possível modelar um fundo com cena dinâmica, que possui mais de um valor para moda.

## 2.3 Revisão de Literatura

O reconhecimento de faces, que pertencem ao escopo da área de visão computacional, é usado desde o controle de acessos à prédios até sistemas de

reconhecimento acoplados em robôs. Em [17] são criados grupos de padrões que determinam as variações da cor da pele humana para serem utilizados para detecção de faces e características faciais. Os modelos criados com base na cor da pele humana são feitos através de uma distribuição gaussiana, sendo que a classificação da pele independe da raça.

Criação de modelos 3-D para rastreamento de objetos onde são dos objetos capturados pela câmera. O sistema desenvolvido por Lin [16] é utilizado em três aplicações: a primeira para controle das ações do mouse, a segunda é uma modelagem 3-D da mão simulando o movimento da mão real capturada pela câmera e a terceira é o controle de uma mesa de áudio simulando a ação de um DJ.

A interação homem-máquina como os artigos publicados por [1, 3, 20] descritos abaixo, visam através do rastreamento da face humana uma nova interface de comunicação com o computador.

Barea [2] apresenta um sistema baseado em eletro-oculograma em que o usuário do sistema controla uma cadeira de rodas através do movimento da cabeça e dos olhos. São colocados ao redor dos olhos cinco eletrodos que captam a variação em micro volts da movimentação dos olhos para se estimar a direção do olhar. A Figura 2.14 mostra a localização dos eletrodos no rosto do usuário do sistema.

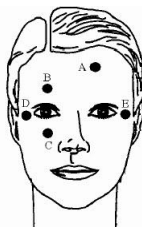


Figura 2.14: Posição dos eletrodos na pesquisa de Barea. Fonte:[2]

Bagci [1] propõe um método de detecção e rastreamento baseado na característica da cor e geometria do rosto humano. Neste método, a decisão de que o olho está fechado ou não, é realizada utilizando modelo de Markov, o método funciona com várias poses da cabeça, porém enquanto a narina seja visível pela câmera.

Bourel [3] apresenta uma solução de rastreamento da face baseada na técnica KLT (Lucas- Kanade-Tomasi). A versão desenvolvida é especializada para o rastreamento da face humana possuindo uma recuperação automática dos pontos perdidos na transição de quadros, e tem como ponto de referência da aplicação a narina. No total são calculados doze pontos, dois pontos pertencentes à narina, seis pontos pertencentes à sobrancelha e quatro pontos

pertencentes à boca. Os pontos de referência citados acima são mostrados na Figura 2.15



Figura 2.15: Pontos de referência para identificação da face humana por Bourel. Fonte:[3]

Morimoto [20] apresenta uma solução que utiliza dispositivos de iluminação infravermelha para gerar duas imagens da pupila, uma clara e outra escura. A pupila resultante utilizada na estimação da direção do olhar é calculada com a comparação entre a diferença das duas imagens obtidas e um valor limiar ( do inglês *threshold* ) base. A estimativa do tamanho e posição da face é feita com os dois melhores candidatos à pupila. Na Figura 3.3 um exemplo das imagens da pupila.



Figura 2.16: Exemplo das imagens da pupila apresentado por Morimoto. Fonte:[20]

Além do rastreamento de objetos visando a interação homem-máquina, o rastreamento pode ser utilizado com o objetivo de monitorar e controlar um ambiente. Os artigos publicados por [24, 25, 14, 15] descritos abaixo utilizam a técnica de mistura gaussiana para realizar o rastreamento de objetos.

Para discriminar um objeto no primeiro plano da cena, Simon e Andrew [24] utilizam um modelo estatístico da distribuição das intensidades do pixel do fundo da cena. O foco da pesquisa é realizar a análise em um ambiente onde a imagem capturada é efetuada em uma base fixa. Quando a câmera é fixa em um veículo em movimento não apresenta resultados satisfatórios, pois o fundo sofre muitas variações dificultando o rastreamento de objetos no primeiro plano.

Stauffer [25] demonstra um modelo de segmentação que modela cada pixel da imagem como uma Mistura Gaussiana para se estimar o fundo da

imagem ( do inglês *background* ). São utilizados dois parâmetros de análise, o primeiro a constante de aprendizagem e o segundo a proporção de dados que devem ser considerados para formação do fundo da imagem. Exemplo, de um quadro, após o rastreamento desenvolvido por Stauffer é mostrado na Figura 2.17. A imagem (b) da Figura 2.17 mostra a área que está sendo monitorada pela aplicação e a imagem (a) da Figura 2.17 mostra o resultado do rastreamento, em que são segmentados os objetos que são rastreados.

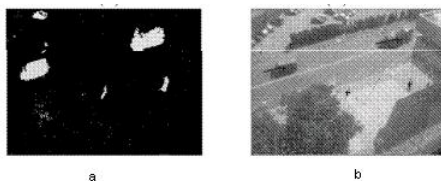


Figura 2.17: A imagem (a) mostra os objetos rastreados, e a imagem (b) representa a imagem capturada pela câmera. Fonte:[25]

Grimson [14] efetua o rastreamento com o objetivo de monitoramento e vigilância em tempo real de um ambiente externo em que o modelo base para subtração possui a característica de se adaptar à cena observada de acordo com a função de mistura gaussiana que é um dos métodos mais robustos de rastreamento. Exemplo, de um quadro, após o rastreamento desenvolvido por Grimson é mostrado na Figura 2.18. A imagem (a) da Figura 2.18 mostra a área que está sendo monitorada pela aplicação e a imagem (b) da Figura 2.18 mostra a área que ocorre circulação de automóveis e pedestres.

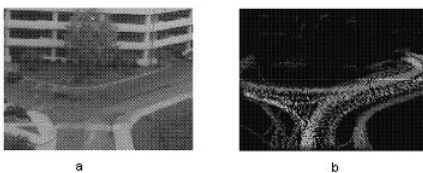


Figura 2.18: A imagem (a) representa a imagem real capturada, e a imagem (b) representa a área onde ocorre a circulação de automóveis e pedestres. Fonte:[14]

Lee [15] desenvolve um sistema de rastreamento com o objetivo de identificar pessoas em um ambiente interno que utiliza uma câmera com visão panorâmica capturando 360° do ambiente. No problema de segmentação do fundo da cena o pixel é definido como sendo a representação do fundo da cena ou frente da cena (do inglês *foreground*).



Na perspectiva Bayesiana, utilizada por Lee, a definição é baseada na suposição do valor do pixel no tempo futuro ( $t+1$ ) ser o fundo da cena  $P(B|x(r, c, t))$ , em que  $x(r, c, t)$  representa o pixel  $x$  na posição  $r, c$  no tempo  $t$  e  $B$  representa a classe background. Imagem da Figura 2.19 mostra a área de monitoramento do sistema.

O problema de segmentação é dividido em dois sub problemas:

- estimar a distribuição para cada pixel como uma mistura gaussiana,
- e verificar a probabilidade do pixel pertencer ao fundo da cena.



Figura 2.19: Imagem em 360° capturada na pesquisa de Lee. Fonte:[15]

# Capítulo 3

## Desenvolvimento da Pesquisa

Este capítulo trata das etapas de implementação e experimentos realizados com a utilização de programas livres como o Java Media Framework (JMF) e o ImageJ. Inicialmente, neste capítulo serão descritos os programas utilizados para obtenção dos resultados. A descrição dos experimentos realizados será descrita na Seção 3.2 deste capítulo.

### 3.1 Programas Utilizados

Foram utilizados dois programas livres para o desenvolvimento deste projeto de graduação:

- JMF: aquisição da imagem capturada pela webcam;
- ImageJ: teste de técnicas de processamento de imagens.

#### 3.1.1 Java Media Framework

A Sun disponibiliza gratuitamente pela internet<sup>1</sup> um conjunto de soluções, denominado *The Java Media Family*, para manipulação de diversos tipos diferentes de mídias. O Java Media Framework (JMF), Figura 3.1, desenvolvido pela Sun faz parte desse conjunto de soluções, assim como:

- Java 2D - que provê um modelo abstrato para trabalhar com imagens do tipo vetorial 2D
- Java 3D - que provê um modelo abstrato e interativo para manipulação de objetos 3D.

---

<sup>1</sup><http://java.sun.com/>

O JMF possui uma arquitetura simples e unificada, um protocolo de troca de mensagens e tratamento de mídia. Para a linguagem Java não há a barreira da portabilidade, sendo desenvolvida como uma linguagem de propósito geral, orientada a objetos e com características modernas de programação.

Com o JMF é possível a manipulação de áudio, vídeo e outras mídias em aplicações Java, pois proporciona um alto nível de abstração em relação à captura, conferência e reprodução de mídia baseada em tempo. Com ela é possível, capturar áudio e vídeo e codificá-lo em diversos formatos como também a transmissão das mídias pelo padrão RTP (Real time Transport Protocol), para o desenvolvimento de aplicações que utilizem vídeo sob demanda.



Figura 3.1: Java Media Framework

O JMF foi desenvolvido para suportar as principais arquiteturas e codecs de áudio e vídeo. Diversos tipos de mídia são suportados como AIFF, AU, AVI, GSM, MIDI, MPEG, QuickTime, RMF, e WAV. As aplicações desenvolvidas através do JMF possuem alto grau de interoperabilidade sendo independentes de plataforma.

Na pesquisa deste trabalho de graduação, o JMF está sendo utilizado como um forma rápida e simples de acesso ao quadro capturado em tempo real pela webcam. Com o quadro facilmente capturado, foi realizado a implementação de protótipos de segmentação de múltiplos objetos através da técnica de mistura gaussiana apresentada na Seção 2.2 do Capítulo 2. A captura da imagens pela webcam é realizada, no protótipo implementado, pela classe Frame Grabber disponibilizada pelo site: <http://www.uk-dave.com/>.

### 3.1.2 ImageJ

O ImageJ, Figura 3.2, é um robusto manipulador de imagens, com código aberto, disponibilizado gratuitamente pelo site <http://rsb.info.nih.gov/ij/>. Além de possuir uma barra padrão para processamento de imagens do programa, possibilita a fácil instalação de novos módulos (do inglês *plugins*) desenvolvidos em linguagem Java, para processamento de imagens.

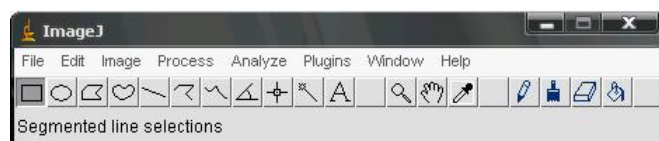


Figura 3.2: ImageJ

O acesso aos dados da imagem é realizado por uma camada intermediária implementada na classe `ImageAccess`, que abstrai o acesso aos dados através de chamadas a métodos, criando um acesso de alto nível aos pixels da imagem.

O ImageJ é multi-plataforma, sendo executado no Linux, no Mac OS 9, no Mac OS X e no Windows. Vários tipos de dados são aceitos para serem executados no ImageJ: imagens em escala de cinza de 8-bits, cores indexadas de 8-bit, 32-bit RGB color etc.

### Adição de Plugins ao ImageJ

A adição de novos plugins ao ImageJ é uma tarefa trivial. Após serem implementados na linguagem de Java e compilados, basta serem armazenados na pasta `plugins` no diretório do ImageJ. Com a inicialização do ImageJ, será criado um submenu com o mesmo nome do módulo desenvolvido.

No desenvolvimento deste trabalho de pesquisa foi implementado um módulo para o ImageJ com o objetivo de criar um modelo baseado na cor do pixel. São calculados os valores média e variância da intensidade da cor de cada pixel da imagem desejada. Os valores obtidos no módulo implementado foram utilizados para criação de um modelo de cor para ser utilizado como parâmetro na segmentação de objetos.

Foram implementados um módulo para criação do gráfico em duas dimensões, referente a seleção dos parâmetros da mistura de gaussianas, e um módulo para estimar a taxa de acerto do protótipo. Ainda foi utilizado, neste trabalho de pesquisa, a classe `Color Histogram` para a criação dos histogramas e a classe `Color Inspector 3D` para criação do Gráfico 3D mostrado no Capítulo 4. É disponibilizado pelo site <http://rsb.info.nih.gov/ij/plugins/> códigos de plugins desenvolvidos.

Para o cálculo da taxa de acerto dos protótipos, foram implementados dois módulos para o ImageJ. O primeiro calcula a taxa de acerto com diferentes números de gaussianas modelando o fundo da cena. O segundo calcula a taxa de acerto de cada quadro na pilha de imagens.

## 3.2 Implementação, Experimentos e Resultados Preliminares

Para o desenvolvimento das implementações e dos experimentos deste projeto foi utilizada uma máquina com processador AMD Duron de 1.20 GHz e 376 MB de memória RAM. Os dados de entrada dos protótipos são capturados diretamente de uma webcam D-link<sup>2</sup>

Duas análises foram realizadas no desenvolvimento dos protótipos: ajuste dos parâmetros da mistura de gaussianas e estimativa de cada pixel do quadro pertencer ao fundo da cena. Com o decorrer da pesquisa, observou-se a possibilidade de três abordagens para criação dos modelos nos protótipos implementados neste trabalho de pesquisa:

1. criação de um modelo pré-definido com base na cor do objeto, para a criação do modelo foi desenvolvido um módulo para o ImageJ que calcula a média e variância dos pixel, Seção 3.2.1;
2. criação de um modelo base que se adapta a variações no ambiente de captura como, por exemplo, a variação da iluminação do ambiente ou inclusão ao fundo da cena de objetos que estavam em movimento e param de se deslocar, Seção 3.2.2;
3. associação dos itens anteriores para criação de um modelo que se adapte a variações no ambiente, mas maior probabilidade de serem definidos com frente da cena os pixel com a propriedade de pertencer ao modelo de pele, Seção 3.2.3.

Para o desenvolvimento dos modelos, a cor do pixel foi o parâmetro utilizado. Os parâmetros utilizados foram:

- RG do modelo de RGB. A webcam captura como padrão neste modelo de cor e com isso não é adicionado mais uma etapa para o processamento da imagem;
- Os parâmetros rg do modelo de cor RGB normalizado. O modelo RGB normalizado representa a cor "pura" e possui menos influência da iluminação do ambiente.
- e o modelo HSV pela propriedade de representar a tonalidade indicando a cor, a saturação indicando o grau de pureza da cor e o brilho indicando as variações de iluminação do ambiente.

---

<sup>2</sup>Características técnicas da webcam: padrão USB versão 1.1, sensor de imagem CMOS, resolução 100K Pixel (352x288), extensão de cor 24-bits.

Alguns experimentos utilizando mistura gaussiana para modelar o fundo da cena foram iniciados nesta pesquisa. O usuário no momento da execução da aplicação irá ver uma janela que possui duas cenas, a cena (a) da janela, Figura 3.3, mostra a cena capturada pela webcam sem nenhum processamento e a cena (b) da janela, Figura 3.3, mostra o resultado da segmentação.

Na imagem processada os pixels definidos como pertencentes ao fundo da cena recebem a cor preta e os pixels que não pertencem ao fundo da cena recebem o valor capturado pela webcam, Figura 3.3.



Figura 3.3: Janela da aplicação de segmentação

### 3.2.1 Modelo Baseado na Cor do Objeto

A abordagem inicial da pesquisa foi a criação de um modelo baseado na cor do objeto com os parâmetros da gaussiana (média e variância). Para este experimento foi implementado um módulo no ImageJ que calcula os parâmetros do objeto.

A Figura 3.4 representa a amostra do objeto a ser segmentado e a Figura 3.5 mostra o histograma do objeto, demonstrando que os componentes do modelo RGB, para o objeto, possuem a forma de uma gaussiana.



Figura 3.4: Amostra para criação do modelo de cor

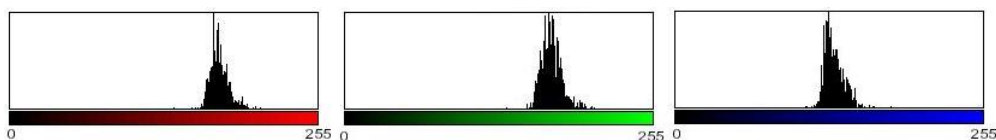


Figura 3.5: Histograma da amostra do objeto segmentado

A definição da classe que um padrão pertence é feita através da verificação da similaridade entre o modelo criado e o padrão desconhecido. A similaridade entre os dados é medida com a análise da média da gaussiana

do modelo e o padrão analisado. Para o modelo que possui um atributo com mais de dimensão, todas as dimensões do atributo serão analisadas.

O primeiro modelo criado no desenvolvimento da pesquisa, é representado no modelo de cor RGB pelos 3 componentes, sendo cada componente uma dimensão da gaussiana. A Figura 3.6 demonstra os resultados iniciais obtidos, em que a imagem (a) mostra o quadro capturado pela webcam e a imagem (b) mostra o resultado obtido da segmentação do objeto. Variações na iluminação do ambiente prejudicaram o desempenho da segmentação do objeto. O Algoritmo 1 demonstra a abordagem adotada para a definição do pixel na cena capturada.

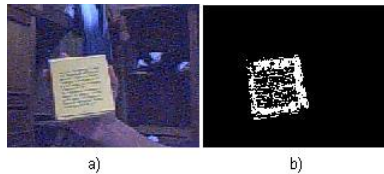


Figura 3.6: Segmentação do objeto a partir do modelo de cor

---

**Algorithm 1** Algoritmo simplificado para o Modelo Com Base na Cor do Objeto

---

```
for  $i \leq$  quantidade de pixels da imagem do  
  if  $\text{match}(\text{pixel}[i]) = \text{verdadeiro}$  then  
     $\text{pixel} \leftarrow$  frente  
  else  
     $\text{pixel} \leftarrow$  fundo  
  end if  
end for
```

---

Como a plataforma SIGUS visa a interação homem-computador, foi implementado também um modelo de mistura gaussiana com base na cor da pele humana, em que foram calculados os valores média e variância para a cor da pele humana.

Para a criação do modelo de pele, foram extraídas três amostras da pele, Figura 3.7, em que cada amostra modela uma gaussiana específica da mistura de 3 gaussianas. Os testes foram realizados nos três modelos de cores citados na seção 3.2. A Figura 3.8 mostra o histograma da amostra de pele no modelo de cor RGB.

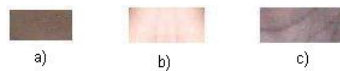


Figura 3.7: Amostras de pele para criação do modelo

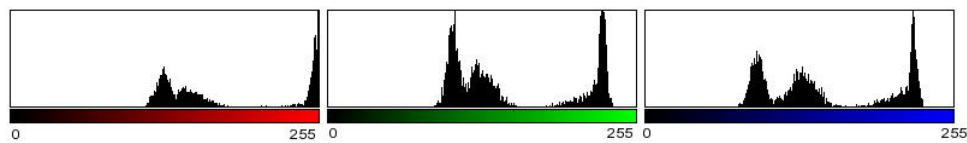


Figura 3.8: Histograma das amostras utilizadas para criação do modelo com base na pele humana

A Figura 3.9 mostra a região no espaço bidimensional (utilizando os parâmetros RG do modelo de cor RGB) dos pixels da amostra. Os gráficos (a), (b), (c) da Figura 3.9 representam a área de cada amostra, e o Gráfico (d) representa a área total correspondente ao modelo de cor criado.

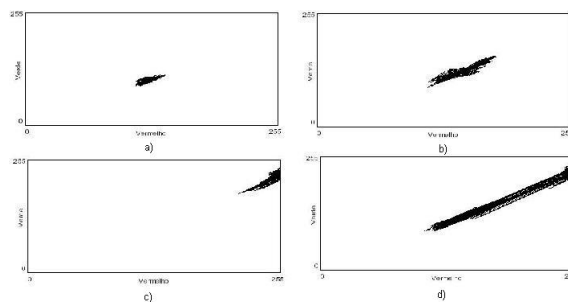


Figura 3.9: Espaço 2D do modelo criado com RG

Utilizando somente o modelo de cor da pele, não é possível definir se os pixels são parte de um objeto em movimento. Adotando esta abordagem para a segmentação, independentemente do estado do pixel (parte de um objeto em movimento ou parado), este sempre será definido como frente da cena.

A Figura 3.10 demonstra resultados da segmentação com a utilização do modelo de cor baseado na cor da pele. A imagem (a) mostra o resultado de um quadro utilizando os componentes RG do modelo RGB, a imagem (b) mostra o resultado de um quadro utilizando os componentes SV do modelo de com HSV, e a imagem(c) mostra o resultado de um quadro utilizando os componentes rg do modelo de com RGB normalizado.





Figura 3.10: Exemplos de segmentação: a - RG do modelo RGB, b - SV do modelo HSV, c - rg do modelo RGB normalizado

### 3.2.2 Modelo Do Fundo da Cena

Primeiramente verifica-se, com a análise do histograma da imagem, se cada pixel da imagem pode ser considerado como uma distribuição gaussiana. Os histogramas para modelagem do fundo da cena correspondem aos valores que um pixel recebe em um intervalo de tempo, Figura 3.11. O histograma na Figura 3.12 representa um pixel qualquer capturado pela webcam no modelo de cor RGB.

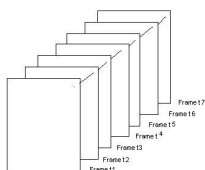


Figura 3.11: Modelo para criação do histograma em relação ao tempo

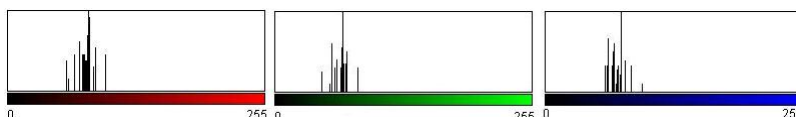


Figura 3.12: Histograma de um pixel qualquer da imagem

Verifica-se através da análise do histograma que a intensidade da cor do pixel possui uma distribuição gaussiana. Porém, em ambientes dinâmicos ( quando possui mais de um valor para moda ), uma gaussiana simples não corresponde a realidade do ambiente, Figura 3.13. Em ambientes dinâmicos podem ser encontrados, por exemplo, dois valores de moda. Para situações em que uma gaussiana simples não pode ser utilizada, pode-se utilizar a mistura de gaussianas.

A mistura de gaussianas, Equação 2.3, pode ser utilizada na segmentação de objetos para modelagem do fundo da cena, em que cada pixel da imagem

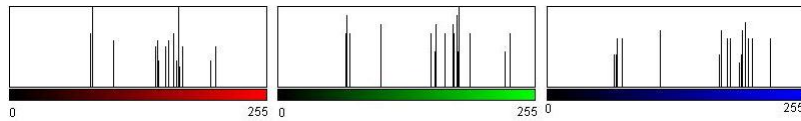


Figura 3.13: Mistura de M Gaussianas

é considerado um processo estatístico que define se o pixel avaliado pertence ao modelo base [24, 14].

Como visto na Seção 2.2 do Capítulo 2, cada gaussiana da mistura de gaussianas é representada pelos seus parâmetros: peso ( $\omega$ ), média ( $\mu$ ) e a variância ( $\sigma^2$ ) para atributos unidimensionais, ou a matriz de covariância ( $\Sigma$ ) para atributos multidimensionais.

Quando o modelo de mistura de gaussianas representa o fundo da cena, três definições iniciais são importantes para a segmentação:

1. Cada gaussiana da mistura de gaussianas modela um superfície da imagem: ou modela o fundo da cena ou a frente da cena [23]. Na utilização de, por exemplo, uma mistura de 3 gaussianas. Duas podem modelar o fundo da cena e uma pode modelar a frente da cena, a representação em superfícies é mostrada na Figura 3.14.

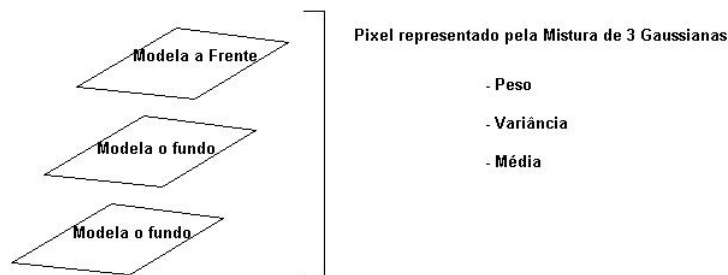


Figura 3.14: Representação em superfície das gaussianas

2. O fundo da cena tem a característica de ocorrer frequentemente (tem peso alto) e variar pouco (tem variância baixa), pois o valor do pixel permanece constante ou tem poucas variações em um intervalo de tempo. Com isso podemos determinar quais gaussianas modelam o fundo da cena, para isso ordenamos de forma decrescente as gaussianas dividindo o peso pela variância através da fórmula  $\omega / \sigma$  [14].
3. Com a ordenação das gaussianas realizada a próxima etapa é a adaptação do fundo da cena. Para que a atualização dos parâmetros

possa ser realizada em tempo real, na prática, é realiza uma atualização K-média dos parâmetros [18].

Os parâmetros são atualizados em relação ao valor da variável no instante anterior. Por isso nas Equações 3.2, 3.3 e 3.1 a atualização dos parâmetros é feita em relação ao tempo t-1. As equações são propostas em [18].

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha S_k \quad (3.1)$$

$$\mu_{i,t} = (1 - \alpha)\mu_{i,t-1} + \alpha x_t \quad (3.2)$$

$$\sigma_{i,t}^2 = (1 - \alpha)\sigma_{i,t-1}^2 + \alpha(x_t - \mu_{i,t})^T(x_t - \mu_{i,t}) \quad (3.3)$$

- A constante  $\alpha$  corresponde a taxa de aprendizagem do sistema.
- O valor  $S_k$  presente na Equação 3.1 para atualização do peso pode receber dois valores: 0 ou 1.

A atualização dos parâmetros é realizada com a verificação da similaridade entre o pixel analisado e a média de cada gaussiana do modelo, sendo que a análise é feita de forma independente em cada superfície do modelo. Quando o pixel desconhecido está dentro do “intervalo” da gaussiana há a ocorrência do “match” na gaussiana. Três situações definem a atualização dos parâmetros da mistura de gaussianas:

1. Caso ocorra o match, a variância (  $\sigma^2$  ) é atualizada de acordo com a Equação 3.3 e a média (  $\mu$  ) é atualizada de acordo com a Equação 3.2. O valor  $S_k$  da Equação 3.1, para atualização do peso, recebe valor 1.
2. Caso não ocorra o match, a variância e a média permanecem com o mesmo valor e o valor de  $S_k$  da Equação 3.1, para atualização do peso, recebe 0.
3. Caso não ocorra nenhum match na mistura de gaussianas o valor da média é substituído pelo padrão atual, a variância recebe um valor alto e o peso recebe um valor baixo[18].

Além do “match”, visto anteriormente, um segundo limiar é utilizado. O segundo limiar é responsável em definir se o pixel analisado pertence ao fundo da cena. Como as gaussianas estão ordenadas de forma decrescente em relação a equação  $\omega/\sigma$ , então as primeiras gaussianas, que possuem peso alto e baixa variância, modelam o fundo da cena. A modelagem de um novo fundo ocorre em duas situações:

1. quando objetos que estavam em movimento param de se movimentar,
2. ou quando objetos que estavam parados começam a se deslocar na cena.

Com o modelo do fundo atualizado, a definição do fundo é realizada através da comparação da soma do peso das  $k$  gaussianas que modelam o fundo com um limiar denominado “limiar fundo”. Caso o valor da soma seja superior ao limiar fundo, o pixel é definido como fundo da cena, ou caso o valor seja inferior ao limiar fundo, o pixel é considerado como frente da cena. O Algoritmo 2 mostra uma abordagem para realização da modelagem e definição do fundo da cena.

Inicialmente, na implementação dos protótipos, a definição do fundo da cena era feita somente com a comparação das gaussianas que modelam o fundo com o “limiar fundo”. Porém, com esta abordagem, o resultado final do quadro analisado possuía muito ruído, pois alguns pixel eram definidos isoladamente como frente da cena.

Para minimizar este problema é feita uma análise dois oito vizinhos mais próximos, em que um pixel é definido também em relação a sua vizinhança. A janela (a) e a janela (b) da Figura 3.15, mostram os resultados da segmentação com os mesmos limiares: “limiar match” igual à 3 e “limiar fundo” igual à 0.7.

A diferença entre os resultados, é que na janela (a) da Figura 3.15, a definição do fundo da cena é feita sem a verificação da vizinhança do pixel, e a janela (b) da Figura 3.15, a definição do fundo da cena é feita com a análise da vizinhança do pixel.



Figura 3.15: Segmentação verificando a vizinhança do pixel

Como visto na Introdução, um sistema robusto deve se adaptar ao ambiente de captura da imagem. Um exemplo de quando não ocorre a adaptação

**Algorithm 2** Algoritmo para o Modelo de Fundo da Cena

---

```
for all  $i \leq$  quantidade de pixels da imagem do  
    ordenaGaussianas()  
  
    for all  $j \leq$  quantidade de gaussianas da mistura de gaussianas do  
        if  $\text{match}(\text{pixel}[i]) = \text{verdadeiro}$  then  
            atualizaVariância()  
            atualizaMedia()  
             $S_k \leftarrow 1$   
        else  
             $S_k \leftarrow 0$   
        end if  
         $\text{peso} \leftarrow \text{atualizaPeso}(S_k)$   
  
    end for  
  
    if nenhuma gaussiana  $\text{match} = \text{verdadeiro}$  then  
        ajustaPeso()  
         $\text{mediagaussiana} \leftarrow \text{pixel}[i]$   
    end if  
  
     $\text{mfundo} \leftarrow$  número de gaussianas que representam o fundo  
     $\text{limiarfundo} \leftarrow$  limiar para definir o fundo  
  
    normalizaPeso()  
  
     $\text{somapeso} \leftarrow 0$   
    for all  $l \leq \text{mfundo}$  do  
         $\text{somapeso} \leftarrow \text{somapeso} + \text{gaussiana}[l]$   
    end for  
  
    if  $\text{somapeso} \leq \text{limiarfundo}$  then  
         $\text{pixel} \leftarrow \text{frente}$   
    else  
         $\text{pixel} \leftarrow \text{fundo}$   
    end if  
  
end for
```

do fundo da cena é mostrado na Figura 3.16. O objeto circulado de vermelho na janela (a) da Figura 3.16, é definido como fundo da cena, pois não está em movimento. A partir do momento que se desloca, janela (b) da Figura 3.16, não ocorre a adaptação do fundo da cena.



Figura 3.16: Exemplos da não adaptação do fundo da cena

Esse problema é resolvido com a verificação das ocorrências do match. Como descrito no Algoritmo 2, se nenhuma gaussiana da mistura de gaussianas “match”, a média da gaussiana recebe o valor do pixel atual. Com essa verificação, o modelo base possuirá o valor do pixel pertencente ao fundo da cena, e com isso se manterá atualizado em relação ao tempo.

Foram feitos alguns teste com valores diferentes para o “limiar match” e para a análise da vizinhança do pixel. A Figura 3.17 demonstra a segmentação com valores diferentes do “limiar match”. A janela (a) da Figura 3.17 realiza a segmentação com o “limiar match” igual à 11 e sem a análise os vizinhos mais próximos, e a janela (b) da Figura 3.17 com “limiar match” igual à 8 e com a análise da vizinhança.

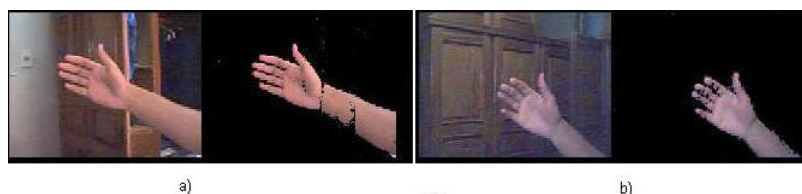


Figura 3.17: Exemplos de segmentação com limiares diferentes

### 3.2.3 Associação do Modelo Do Fundo da Cena e do Modelo Baseado na Cor do Objeto

Nesta etapa da pesquisa são utilizados dois critérios para ajuste dos parâmetros das gaussianas e com isso definir os pixels pertencentes ao fundo da cena. Como mostra o diagrama da Figura 3.18 os parâmetros da mistura de gaussianas são ajustados separadamente tendo com base:

1. O Modelo Do Fundo da Cena, Seção 3.2.2
2. O Modelo Baseado na Cor do Objeto, Seção 3.2.1

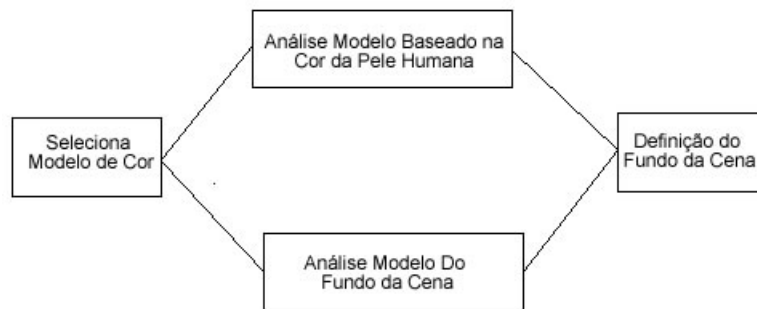


Figura 3.18: Diagrama simplificado do protótipo de segmentação implementado

O método que modela o fundo da cena é utilizado juntamente com a modelagem da cor do objeto. Como a pesquisa visa colaborar com a plataforma SIGUS, os testes foram voltados a interação homem-máquina, em que a probabilidade de um pixel pertencer à um objeto em movimento é maior quando o modelo baseado na cor da pele “match”.

Diferentemente da Seção 3.2.2, o modelo do fundo é influenciado, na tomada de decisão, pelo método que modela o objeto com base na cor. Conforme a Figura 3.19, inicialmente os parâmetros das gaussianas são ajustados em relação ao critério de “match” do modelo de fundo.

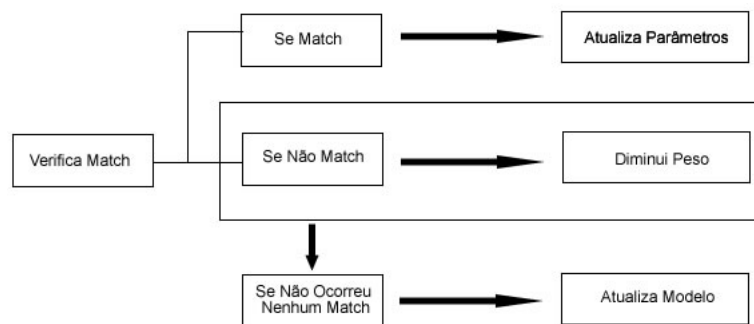


Figura 3.19: Diagrama simplificado do critério de “match”: Modelo Do Fundo da Cena

Após a modelagem do fundo da cena, os pesos das gaussianas são ajustados de acordo com o “match” do modelo de cor, aumentando ou diminuindo a probabilidade de um pixel ser definido como fundo da cena, Figura 3.20.

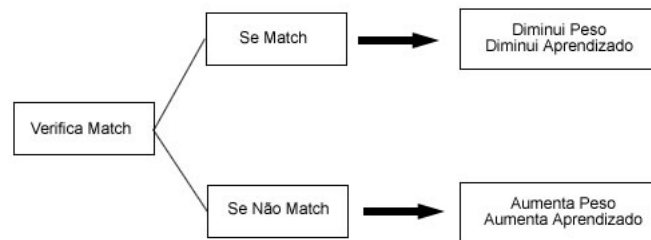


Figura 3.20: Diagrama simplificado: Modelo Baseado na Cor

Como visto na Seção 3.2.2, a definição do fundo é realizada através da comparação da soma do peso das  $k$  gaussianas que modelam o fundo com um limiar denominado “limiar fundo”. Quando maior a soma do peso, maior é a probabilidade do pixel ser classificado como fundo da cena e, por isso, na ocorrência do “match” no modelo de cor, o peso é decrementado. O Algoritmo 3 mostra uma versão de como o modelo de cor da pele influência na tomada de decisão.

A taxa de aprendizagem, mostrada na Figura 3.20, define a “velocidade” com que o pixel será incorporado ao fundo da cena. Portanto, nesta etapa da pesquisa, a taxa de aprendizagem não é mais global no sistema, mas cada pixel possui seu valor que altera em relação ao tempo, criando áreas com maior ou menor taxas de aprendizagem.

A Figura 3.21 é um exemplo simples de visualização de áreas com taxa de aprendizagem diferente. Nesta situação o protótipo deve definir como frente da cena somente os pixels pertencentes a mão do usuário do sistema. A imagem (a) mostra o quadro capturado pela webcam e a imagem (b) mostra a situação ideal em relação a taxa de aprendizagem.

A mão do usuário, na imagem (b) da Figura 3.21, deve possuir taxa de aprendizagem baixa, pois esta região deve ser incorporada “lentamente” ao fundo da cena<sup>3</sup>. A área com a cor branca, na imagem (b) da Figura 3.21, deve possuir taxa de aprendizagem alta, pois o pixel deve ser “rapidamente” incorporado ao fundo da cena<sup>4</sup>.

<sup>3</sup>Lembrando: a mão não estando em movimento, dever ser incorporada ao fundo da cena.

<sup>4</sup>Objetos que não possuem a propriedade da cor da pele, devem ser incorporados “rapidamente” ao fundo da cena.



---

**Algorithm 3** Algoritmo simplificado para abordagem que utiliza o Modelo de Cor e o Modelo Do Fundo da Cena Dinâmico

---

```
for all  $i \leq$  quantidade de pixels da imagem do  
  
  modelaFundo()  
  
  if  $\text{PeleMatch}(\text{pixel}[i]) = \text{verdadeiro}$  then  
    ajustaPeso(verdadeiro)  
    ajustaAprendizado(verdadeiro)  
  else  
    ajustaPeso(falso)  
    ajustaAprendizado(falso)  
  end if  
  
   $m_{\text{fundo}} \leftarrow$  número de gaussianas que representam o fundo  
   $\text{limiar}_{\text{fundo}} \leftarrow$  limiar para definir o fundo  
  
  normalizaPeso()  
  
   $\text{somapeso} \leftarrow 0$   
  for all  $l \leq m_{\text{fundo}}$  do  
     $\text{somapeso} \leftarrow \text{somapeso} + \text{gaussiana}[l]$   
  end for  
  
  if  $\text{somapeso} \leq \text{limiar}_{\text{fundo}}$  then  
     $\text{pixel} \leftarrow$  frente  
  else  
     $\text{pixel} \leftarrow$  fundo  
  end if  
  
end for
```

---



Figura 3.21: Área da taxa de aprendizagem

A Figura 3.22 demonstra os resultados obtidos quando os parâmetros têm ajustes periódicos e quando a cor da pele humana é um critério para atualização dos parâmetros. Foram testados com os parâmetros RG do modelo de cor RGB, Figura 3.22 (a), e o SV do modelo de cor HSV, Figura 3.22 (b).

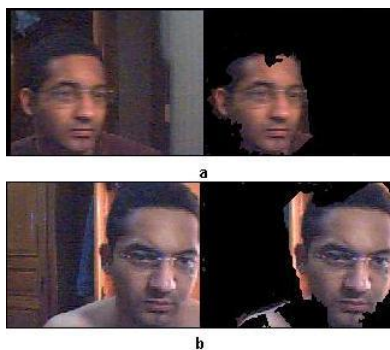


Figura 3.22: Exemplo de segmentação com ajuste periódico dos parâmetros

### 3.2.4 Resultados Obtidos nos Experimentos

Nesta etapa da pesquisa são mostrados os resultados obtidos nos testes. Para realizar o cálculo da taxa de acerto do protótipo, primeiramente os quadros capturados pela webcam foram adicionados em um pilha de imagens, para então serem analisados através de um módulo implementado para o ImageJ.

Inicialmente foi medido a taxa de acerto em relação a quantidade de gaussianas que modelam o fundo da cena. Foram criadas 4 pilhas de imagens, cada pilha corresponde ao resultado obtido com uma quantidade específica de gaussianas modelando o fundo da cena.

Foi medida também a taxa de acerto referente a quantidade total de pixels corretamente classificados em cada quadro da pilha de imagens. Foram utilizadas 4 gaussianas para modelar a superfície fundo de um sistema formado por uma mistura de 5 gaussianas, é mostrada na Tabela 3.1 os resultados obtidos nos experimentos.

N Gaussianas	Mínimo	Média	Máximo
1	0.13%	2.64%	9.59%
2	27.64%	44.72%	60.30%
3	37.67%	73.42%	96.05%
4	56.57%	81.31%	98.35%

Tabela 3.1: Taxa de acerto da quantidade de gaussianas que modelam o fundo

Uma segunda análise sobre a taxa de acerto, foi realizada através da comparação entre o resultado obtido e um resultado ideal para o quadro. A análise foi realizada com imagens em preto e branco. A imagem (a) na Figura 3.23, mostra o resultado da segmentação realizado, e a imagem (b) na Figura 3.23 indica o resultado ideal para a segmentação.

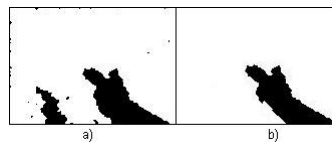


Figura 3.23: Exemplo de análise da taxa de acerto

Foram medidos separadamente a segmentação da mão e da face. Foi calculado o percentual de acerto em cada situação. Os percentuais mínimo, médio e máximo de acerto são mostrados na Tabela 3.2.

	Mínimo	Média	Máximo
Mão	70.22%	82.46%	88.81%
Face	72.63%	84.24%	90.76%

Tabela 3.2: Taxa de acerto para segmentado da mão e da face

Os dados da Tabela 3.2, foram discriminados para se determinar qual a faixa de acerto que ocorre o maior número de acerto. Esta análise é mostrada na Figura 3.3

No espaço de erros existente, são possíveis dois tipos de erro:

1. quando classificado como fundo da cena, sendo que deveria ser classificado como frente da cena
2. quando classificado como frente da cena, sendo que deveria ser classificado como fundo da cena.

A Tabela 3.4 mostra o percentual de erro médio para cada tipo de erro.

	Acima de 90%	Entre 80% e 90%	Entre 70% e 80%
Mão	0.0%	69.57%	30.43%
Face	5.56%	83.33%	11.11%

Tabela 3.3: Intervalo de Acerto

	Média do Erro 1	Média do Erro 2
Mão	78.63%	21.37%
Face	89.45%	10.55%

Tabela 3.4: Percentual dos tipos de erros

Quando ocorre o pior caso na segmentação da mão, a taxa de acerto é muito baixa, como mostrada na Tabela 3.5. O pior caso ocorre quando a mão se desloca “rapidamente” na seqüência de quadros, como visto na Figura 3.24. A imagem (a) da Figura 3.24, mostra o resultado da segmentação e a imagem (b) da Figura 3.24, mostra o resultado ideal para o quadro.

	Mínimo	Média	Máximo
Pior Caso	65.75%	73.02%	78.14%

Tabela 3.5: Baixa taxa de acerto no pior caso

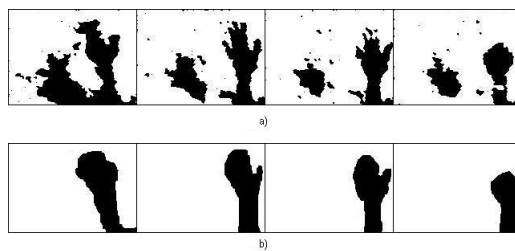


Figura 3.24: Exemplo de uma seqüência do pior caso para a segmentação da mão

# Capítulo 4

## Considerações Finais

Este projeto de graduação apresentou um estudo sobre técnicas de segmentação e rastreamento, voltadas a área de visão computacional. A literatura estudada neste trabalho pesquisa técnicas para desenvolvimento de novos meios de interação homem-máquina e técnicas de segmentação utilizando mistura de gaussianas. Com o estudo inicial concluído, foi desenvolvido um módulo de segmentação utilizando mistura de gaussianas, que será incorporado à plataforma SIGUS, a qual visa o estudo e implementação de técnicas de processamento de imagem voltadas a área de visão computacional.

O protótipo desenvolvido inicialmente utiliza a cor do objeto como parâmetro para a segmentação, independentemente do objeto estar em movimento ou parado na cena. A segunda abordagem visa a modelagem do fundo da cena, através da técnica de mistura de gaussianas, proporcionando adaptabilidade ao ambiente de captura. Como este projeto de graduação tem como objetivo incorporar o módulo desenvolvido junto a plataforma SIGUS, o protótipo principal utiliza a informação da cor da pele como parâmetro na adaptação do modelo de fundo.

Com a disponibilidade do acesso ao código fonte pela internet, há a possibilidade de correção de possíveis falhas existentes ou a otimização do módulo desenvolvido, proporcionando um avanço mais rápido às implementações de segmentação destinadas à plataforma SIGUS.

Como proposta para trabalhos futuros, cita-se a criação de um modelo com base na cor da pele mais preciso. A implementação de um modelo de pele mais preciso proporcionaria uma adaptação mais “lenta” aos pixels classificados como pele e com isso melhorando a taxa de acerto da segmentação. Outra proposta para trabalhos futuros, cita-se que a segmentação realizada neste projeto de graduação poderá ser utilizada como parâmetro para o rastreamento de múltiplos objetos ou no caso da plataforma SIGUS, como parâmetro para o rastreamento das mãos.

# Referências Bibliográficas

- [1] A. Bagci, R. Ansari, e A. Khokar. Eye tracking using markov models. *ICPR 2004*, 3:818–821, 2004.
- [2] R. BAREA, L. BOQUETE, M. MAZO, e E. LÓPEZ. System for assisted mobility using eye movements based on electrooculography. 2002.
- [3] F. Bourel, C. Chibelushi, e A. Low. Robust facial feature tracking. *BNVC*, 2000.
- [4] J. Campbell e F. Murtagh. Image processing and pattern recognition. 1998.
- [5] Charles Casimiro Cavalcante. Predição neural e estimação de função de densidade de probabilidade aplicadas à equalização cega. 2001.
- [6] João Fernando Custódio da Silva, Ricardo Luís Barbosa, Rodrigo Bezerra de Araújo Gallis, e Leonardo Marini Pereira. Avaliação da qualidade da detecção de bordas em uma seqüência de imagens de ruas e rodovias. *Revista Brasileira de Cartografia*, 2004.
- [7] Márcio Portes de Albuquerque e Marcelo Portes de Albuquerque. Processamento de imagens: Métodos e análises. *Centro Brasileiro de Pesquisas Físicas MCT*, 2000.
- [8] Teófilo Emídio de Campos. 3d hand and object tracking for intention recognition. 2003.
- [9] Alceu de Souza Britto Jr., Cinthia Obladen de Almendra Freitas, Edson José R. Justino, Díbio Leandro Borges, Jacques Facon, Flávio Bortolozzi, e Robert Sabourin. Técnicas em processamento e análise de documentos manuscritos. *RITA*, 2, 2001.
- [10] R. Duda, P. Hart, e D. Stork. *Pattern Classification*. Wiley, New York, Chichester, Weinheim, Brisbane, Singapore, Toronto, 2001.

- 
- [11] Rogério Schmidt Feris, Teófilo Emídio de Campos, e Roberto Marcondes Cesar Junior. Detection and tracking of facial features in video sequences. *Lecture Notes in Artificial Intelligence*, 1793:197–206, 2000.
- [12] R.J. Ferrari<sup>1</sup>, R.M. Rangayyan, J.E.L. Desautels, e A.F. Frère. Segmentação de mamogramas: Identificação da borda da mama, músculo peitoral e disco glandular. *ANAIS DO CBEB*, 2000.
- [13] Alvaro Roberto Silvestre Fialho. Estudo de técnicas de rastreamento das mãos para o desenvolvimento de interfaces homem-máquina. 2004.
- [14] W. Grimson, C. Stauffer, R. Romano, e L. Lee. Using adaptive tracking to classify or monitor activities in a site. *CVPR*, página 22, 1998.
- [15] Dar-Shyang Lee. A bayesian framework for background segmentation based on adaptive gaussian mixtures.
- [16] E. Lin, A. Cassidy, D. Hook, A. Baliga, e T. Chen. Hand tracking using spatial gesture modeling and visual feedback for a virtual dj system. *IEEE International Conference on Multimodal Interfaces*, página 197, 2002.
- [17] Eduardo Costa Lopes. Detecção de faces e características faciais.
- [18] Alan McIvor. Background subtraction techniques. *IEEE SMC 2004 International Conference on Systems*, 2004.
- [19] Arlindo N. Montagnoli. Controle por voz utilizando um classificador de máxima verossimilhança aproximado em microcontroladores de 8-bits. *Anais do 3º congresso temático de dinâmica e controle da SBMAC*, 2004.
- [20] C. Morimoto, D. Koons, A. Amir, e M. Flicker. Real-time detection of eyes and faces. *Workshop on Perceptual User Interfaces*, páginas 117–120, 1998.
- [21] L. Nunes e P. Prado. Reconhecimento de objetos contidos em imagens através de redes neurais. *Revista de Ciências Exatas - UNITAU*, 5/8, 1999/2002.
- [22] Hemerson Pistori, João José Neto, Amaury Antônio de Castro Junior, Mauro Conti Pereira, e Tania Regina Vilela dos Santos. Sigus - plataforma de apoio ao desenvolvimento de sistemas para inclusão digital de pessoas com necessidades especiais. 2004.



- 
- [23] P. Wayner Power e Johann A. Scoones. Understanding background mixture models for foreground segmentation. *Proceedings Image and Vision Computing New Zealand 2002*, 2002.
- [24] Simon Rowe e Andrew Blake. Statistical background modelling for tracking with a virtual camera. *British Machine Vision Conference*, páginas 423–432, 1995.
- [25] C. Stauffer e W. Grimson. Adaptive background mixture models for real-time tracking. *CVPR*, páginas 246–252, 1999.
- [26] Michel Alain Quintana Truyenque. Realidade virtual aumentada e cooperativa. Disponível em, <http://www.tecgraf.puc-rio.br/~alain/RAC/>.
- [27] D. Webb. *Statistical Pattern Recognition*. Wiley, Malvern, 2002.